

DENEYAP

Teknoloji Atölyeleri

ROBOTİK VE KODLAMA ORTAOKUL Sözcük Blokları

Yazarlar

Doç. Dr. Memet ÜÇGÜL
Prof. Dr. İbrahim ÇETİN
Prof. Dr. Erman YÜKSELTÜRK
Prof. Dr. Ercan TOP

Güncelleme

Doç. Dr. Memet ÜÇGÜL
Haydar Emin ERDOĞAN
Melike ÖZDEMİR

İçindekiler

Robotik Kodlama Dersi Öğretim Kılavuzu	1
1. Hafta: Robot ve Algoritma	6
GÖZLE VE UYGULA.....	7
Gözle: Temel Kavramlar Üzerine Tartışma	7
Uygula: Algoritma Öğreniyorum	8
1.3. Uygula: Çarpım Eşleri Algoritması	8
TASARLA	8
“Hayalimdeki Robotu Çiziyorum” Etkinliği	8
ÜRET	9
“İlk Robotumu Yapıyorum” Etkinliği	9
DEĞERLENDİR	10
İlave Etkinlik	11
2. Hafta - Robotlar ile Hareket	13
LEGO Education SPIKE Prime Yazılımı.....	14
GÖZLE VE UYGULA.....	17
Gözle: Programlama Alanının Kullanılması	17
Gözle: Robotun Hareket Ettirilmesi	19
Gözle: Robotun Temel Hareketleri.....	20
Gözle ve Uygula: İleri ve Geri Hareket Eden Robot	22
Uygula: Kare Çizen Robot	23
Uygula: A noktasından B noktasına Hareket.....	25
TASARLA	27
ÜRET	28
DEĞERLENDİR	28
İlave Etkinlik	29
3. Hafta – Işık, Ses ve Müzik	31
GÖZLE VE UYGULA.....	32
IŞIK.....	32
Gözle: Hub Ekranı	34
Uygula: Büyüyen Kalp Efektı	34
Gözle: Sürekli Tekrar Eden Büyüyen Kalp Efektı.....	35
Uygula: Başla ve Geri Sayım	36
SES.....	36
Gözle: Ses Blokları.....	38
Uygula: Kare Çize Robot Aktivitesi	39
MÜZİK.....	39
Göster: Müzik Blokları.....	40
Uygula: Beethoven - 9. Senfoni (Neşeye Övgü).....	41
TASARLA	43

ÜRET	44
DEĞERLENDİR	46
İlave Etkinlik	46
4. Hafta - Kuvvet Sensörü	47
Kuvvet Sensörü	48
GÖZLE VE UYGULA	51
Gözle: Kuvvet Sensörü	51
Uygula: Kuvvet Sensörü ile Motorların Çalıştırılması	52
Gözle: Kuvvet Sensörü ile İlk Program	52
Uygula: Kuvvet Sensörü ile Yüzde ve Newton	53
Gözle ve Uygula: Kuvvet Sensörü Oyunu	54
Gözle ve Uygula: Engele Çarpınca Duran Robot	55
Gözle ve Uygula: Çarpışma Testi	57
TASARLA	58
ÜRET	59
DEĞERLENDİR	61
İlave Etkinlik	61
5. Hafta - Mesafe Sensörü	63
Mesafe Sensörü	64
GÖZLE VE UYGULA	65
Gözle: Mesafe Sensörü	65
Gözle: Belirli Bir Mesafeye Kadar İlerleme	66
Uygula: İstenilen Mesafe Kadar Geri Gitme	67
Uygula: Engele Yaklaştıkça Yavaşlayan Robot	68
Gözle: Robota Yaklaşan Nesne Olursa Ses Çıkar	68
Uygula: Park Sensörü	69
TASARLA VE ÜRET	70
DEĞERLENDİR	71
İlave Etkinlik	72
6. Hafta - Hareket Sensörü	73
Hareket Sensörü	74
GÖZLE VE UYGULA	79
Gözle ve Uygula: Robotun 90 Derece Döndürülmesi	79
Uygula: Kare Şeklinde Hareket	82
Gözle ve Uygula: Sürekli Yukarıyı Gösteren Ok	83
Gözle ve Uygula: Yunuslama, Yuvarlanma Işıkları	84
TASARLA	85
ÜRET	87
DEĞERLENDİR	89
İLAVE ETKİNLİK	90
7. Hafta- Renk Sensörü	92

Robotik ve Kodlama Ortaokul

Renk Sensörü	93
GÖZLE VE UYGULA.....	95
Gözle ve Uygula: Karşılaşılan Rengin Numarasını Hub Ekranında Gösterme	95
Gözle: Yansıyan Işık Şiddeti.....	97
Uygula: Çalışma Alanının İçinden Çıkmayan Robot	97
Uygula: Yansıyan Işığın Değerini Hub Ekranında Gösterme.....	99
TASARLA VE ÜRET	100
ÜRET.....	101
DEĞERLENDİR	101
İlave Etkinlik	102
8. Hafta – Sensörlerin Birlikte Kullanımı.....	103
Sensörlerin Birlikte Kullanımı.....	104
GÖZLE VE UYGULA.....	105
Gözle ve Uygula: Sensörlerin Birlikte Kullanımı.....	105
Gözle ve Uygula: Mantık Operatörleri Değil (Not).....	109
Gözle ve Uygula: Mantık Operatörleri Veya (Or)	111
Gözle ve Uygula: Mantık Operatörleri Ve (And).....	112
TASARLA	114
ÜRET.....	119
DEĞERLENDİR	121
9. Hafta – Oransal Programlar.....	122
Oransal Programlar	123
GÖZLE VE UYGULA.....	123
Gözle ve Uygula: Öndeki Aracı Takip Eden Robot	123
Gözle ve Uygula: Yönünü Kaybetmeyen Robot.....	124
Gözle ve Uygula: Tepki Gücü	125
Gözle ve Uygula: Oransal Çizgi Takibi.....	127
TASARLA	128
ÜRET.....	131
DEĞERLENDİR	132
10. Hafta- Bloklarım.....	134
Bloklarım	135
GÖZLE VE UYGULA.....	135
Gözle: Bloklarım Paleti	135
Uygula: Sapma Açısı ile Sağa Dönen Robot.....	136
Gözle: Bloklarla Değer Göndermek	137
Uygula: Bloğa Gönderilen Değerler ile İlerleme ve Dönmenin Belirlenmesi	138
Uygula: Hızlanmaya Devam Eden Robot	138
TASARLA VE ÜRET	139
ÜRET.....	140
DEĞERLENDİR	141

11. Hafta - Proje Hazırlıyoruz	143
GELİŞTİRME – TEST ETME	144
12. Hafta - Proje Hazırlıyoruz	145
SUNUM	146

ROBOTİK KODLAMA DERSİ ÖĞRETİM KILAVUZU

Dünyada 21.yüzyıl ile başlayan yenilikler yaşamın her alanını etkilemektedir. Hayatımızdaki hızlı değişim ve gelişim, her gün beraberinde yeni bilgiler, yeni teknolojiler ve yeni yaklaşım tarzları ortaya çıkarmaktadır. Bu gelişmelere bağlı olarak küresel eğitimde ve işgücü piyasasında ihtiyaç duyulan öğrenen ve çalışan özelliklerinde değişimler gözlenmektedir. Bu nedenlerden, eleştirel düşünme, problem çözme, analiz ve sentez yapma, işbirlikli öğrenme ortamlarında çalışabilme, doğru ve güncel bilgiye kolay bir şekilde ulaşabilme ve yenilikçi olabilme gibi üst düzey becerilerin kazanılması beklenmektedir. Alan yazında birçok çalışma da göstermiştir ki iyi planlanan programlama eğitiminin, 21.yüzyıl becerileri (P21, 2020) olarak da tarif edilen çeşitli üst düzey becerilerin kazanımında olumlu katkılar sağlamaktadır (Gürer, Çetin & Top, 2019; Yükseltürk & Altıok, 2017).

Programlama ile öğrencilerin bilgi okuryazarlığı, algoritmik düşünme ve problem çözme becerilerinin kazanılmasında etkili olsa da bunlarla sınırlı değildir. Programlama eğitimi ile öğrencilerin araştırma yapması sağlanarak özgün bir projenin baştan sona gelişmesine yardımcı olunabilir. Ayrıca, yapılacak grup çalışmaları ile sorumluluk becerilerinin, hazırlanacak projeler ile de işbirlikli çalışma becerilerinin kazanılmasına da katkı sağlanabilir (Gürer, Çetin & Top, 2019; Yükseltürk & Altıok, 2017).

Günümüzde bireylerden bilgi ve iletişim teknolojileri okur-yazarı olmanın yanında hem eğitim hem de iş hayatında farklı dijital araçları kullanarak karşılaştıkları problemlere çözümler üretmesi beklenmektedir. Araştırmacılar hem temel bilgisayar bilimleri kavramlarını hem de programlama becerilerini bilgi işlemsel düşünme ile ilişkilendirmektedir. Bilgi işlemsel düşünme ise temel olarak bir problem çözme sürecidir ve beş bileşeni içerir (Wing, 2008, 2011):

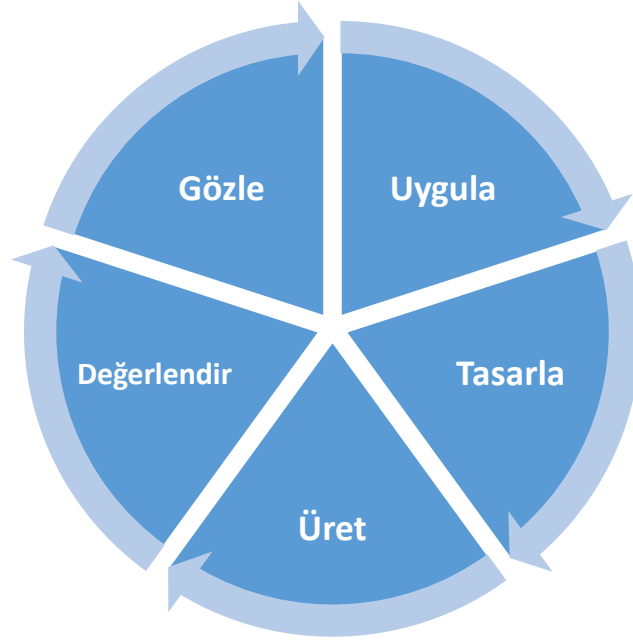
- (i) soyutlama, probleme ilişkin toplanan ve çözümlenen bilgiler ve çözümlerin sınıflandırılması yoluyla kuralların kavramsallaştırılması
- (ii) algoritma, problemler için yeni çözümlerin inşa edilmesi
- (iii) otomasyon ya da otomatikleştirme, geliştirilen kavramsallaştırmaların farklı sorunlarda da verimli ve etkili bir biçimde işlemesi
- (iv) problemleri ayrıştırma, problemlerin daha küçük anlamlı alt problemlere parçalanması
- (v) genelleştirme, problemlere ilişkin üretilen çözüm desenlerinin benzer problemlerde de uygulanabilir olması

Bu derste, öğrencilerin öncelikle bilgi işlemsel düşünme becerisinin geliştirilmesi için robotik programlama eğitiminin verilmesi planlanmaktadır. Yapılandırmacı (constructivist) eğitim yaklaşımını temel alınarak tasarlanan derste, öğrenenin rolü aktif bir şekilde bilgiyi inşa etmek, eğitmen rolü ise öğrencilerini pedagojik olarak destekleyerek keşfetme veya buluş yapabilme ortamlarını hazırlanmak olarak belirlenmiştir. Ders boyunca hazırlanan etkinliklerde programlanabilir robotik materyaller öğrenenlerin çeşitli işlevlere sahip yaratıcı robot tasarımları sağlanacaktır. Ayrıca, öğrenciler tasarım becerilerini kullanarak inşa ettikleri robotları programlayabilecek, böylelikle programlamanın temellerini de öğrenecekler ve öğrendiklerini somut projelere dönüştürebilecektir. Özetle bu dersin amaçları şu şekilde sıralanabilir:

- Bir problemin çözümü için gerekli olan algoritma ve programlama ile ilgili temel kavramları kullanmak

- Robotların belirli amaçlar için programlamalarını sağlayarak, bilgi işlemsel düşünme çözme becerilerini geliştirmek
- Etkinlikler esnasında gerçek hayattan problemlerin sunulduğu öğrenci merkezli bir öğrenme ortamı oluşturarak, öğrencileri çok yönlü ve disiplinler arası düşünmeye yönlendirmek
- Öğrencileri grup çalışmalarına yönlendirerek, onların sosyal becerilerini, iletişim ve işbirliği becerilerini ve liderlik ve sorumluluk alabilme becerilerini geliştirmek

Bu temel amaçlar doğrultusunda ders verilirken daha önceki Robotik ve Kodlama kitabında detaylarının anlatıldığı GUTÜD (Üçgül, Çetin, Yükseltürk ve Top, 2021) “gözle, uygula, tasarla, üret ve değerlendir” öğrenme döngüsü kullanılacaktır. Bu aşamalar özetle şöyle devam edecektir.



Resim 1.1 Öğrenme Döngüsü

Gözle: Öğrenme döngüsü bu bölümle başlar ve öğretmenin daha aktif olduğu bölümdür. Öğretmen öğrencilerin geçmiş bilgilerini aktive eder ve onların dikkat ve motivasyonlarını sağlamaya çalışır. Ayrıca, öğretmen ilgili konuları uygulamalı olarak (göstererek) anlatır. Öğrencilere sorular sorabilir ve öğrencilerin sorularını yanıtlayabilir.

Uygula: Bu bölümde öğretmen öğrenenlerden gözle esnasındaki gösterilen uygulamaların aynısını ya da bir benzerini ister veya onlarla birlikte yapar. Anlatılan konulara göre gözle ve uygula bölümleri birkaç kez tekrar edilebilir.

Tasarla: Bu bölümde öğrenciler daha aktif rol üstlenir. Öğretmen rehber görevini üstlenir ve öğrencilere takıldıkları noktalarda destek olacaktır. Tasarla aşaması öğretmen tarafından öğrenenlere bir problem verilerek başlar. Öğrenenlerden öncelikle bu problemin çözümünü tasarlamaları istenir. Tasarlama aşamasında, öğrenenler temel itibarıyla bilinenler ile istenenler arasındaki bağı kurarak bir plan üreteceklerdir. Bu amaçla, öğrenenler bilgi işlemsel düşünme becerisinin temel bileşenlerini kullanması sağlanacaktır.

Üret: Bu bölümde öğrenciler yine aktif rol üstlenmeye devam eder ve öğretmen rehber pozisyonundadır. Üret aşamasında, öğrenenlerden bir önceki adımda tasarladığı planı kullanarak

probleme algoritmik bir çözüm üretmesi istenir. Öğrenenler bilgisayar ve robot başında çalışarak gerekli donanımsal ve yazılımsal çözümleri geliştirirler.

Değerlendir: Bu bölümde temel hedef, öğrenenin, öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşünmesini sağlamaktır. Böylelikle, öğrenen problem çözme süreci, dersin konusu ve kendisi ile ilgili gözlemler yaparak yeni öğrenmeler, kendini değerlendirme ve planlama açısından fırsatlar elde edecektir.

EŞLİ PROGRAMLAMA ve GRUPLAR ARASI İLETİŞİM

Bu ders sürecinde etkinlikler yapılırken eşli programlama ve gruplar arası iletişime önem verilmektedir. Öğrenenler kendilerine verilen görevlerden uygun olanlarını eşli programlama yaparak tamamlayabilecektir. Eşli programlamada iki öğrenen, bir robot veya bilgisayar karşısında yan yana oturarak tasarım, algoritma, kod yazma veya hata ayıklama için işbirlikli çalışır. Eşli programlama eşli araba yarışlarına benzetilebilir. Eşli araba yarışlarında sürücü arabayı kullanırken kılavuz sürücüye yön tayini konusunda yardımcı olur. Eşli programlamada bilgisayarı veya robotu kullanan kişiye sürücü denir. Sürücünün görevi robotun istenenleri gerçekleştirmesi için tasarımı ve kodlamayı yapmaktır. Sürücünün yanındaki kişiye kılavuz denilir. Kılavuzun görevi çıkan problemler veya ana problem için çözüm üretmek, bu süreçte ortaya çıkan hataları belirlemek ve sürücünün nasıl çalıştığını değerlendirmektir. Eşli programlamada, araba yarışından farklı olarak, sürücü ve kılavuz düzenli olarak yer değiştirir. Öğrenenlerin her iki görevden de öğreneceği şeyler vardır. Bu yüzden rol değiştirme çok önemlidir. Öğretmen öğrenenlerin periyodik olarak görev değiştirmesini sağlamalıdır. Ayrıca, etkinlikler boyunca gruplar arası bilgi alışverişine izin verilmesi önerilmektedir. Gruplar arasında paylaşımcı bir ortam oluşturulmalıdır. Fakat bu paylaşım, komple bir çözümün paylaşımı şeklinde olmamalıdır. Öğrenenler; çözüm yolları, stratejiler ve eksik bilgiler gibi konular için paylaşım yapabilirler. Ayrıca, bu eğitim programı içerisinde zaman zaman gruplar arası yarışlar düzenlenmiştir. Fakat bu yarışlar, sınıfın paylaşımcı ortamını zedeleyecek içerikte uygulanmamalıdır.

ROBOTİK KİTLER VE PROGRAMLAMA ORTAMI

LEGO Education SPIKE Prime Seti

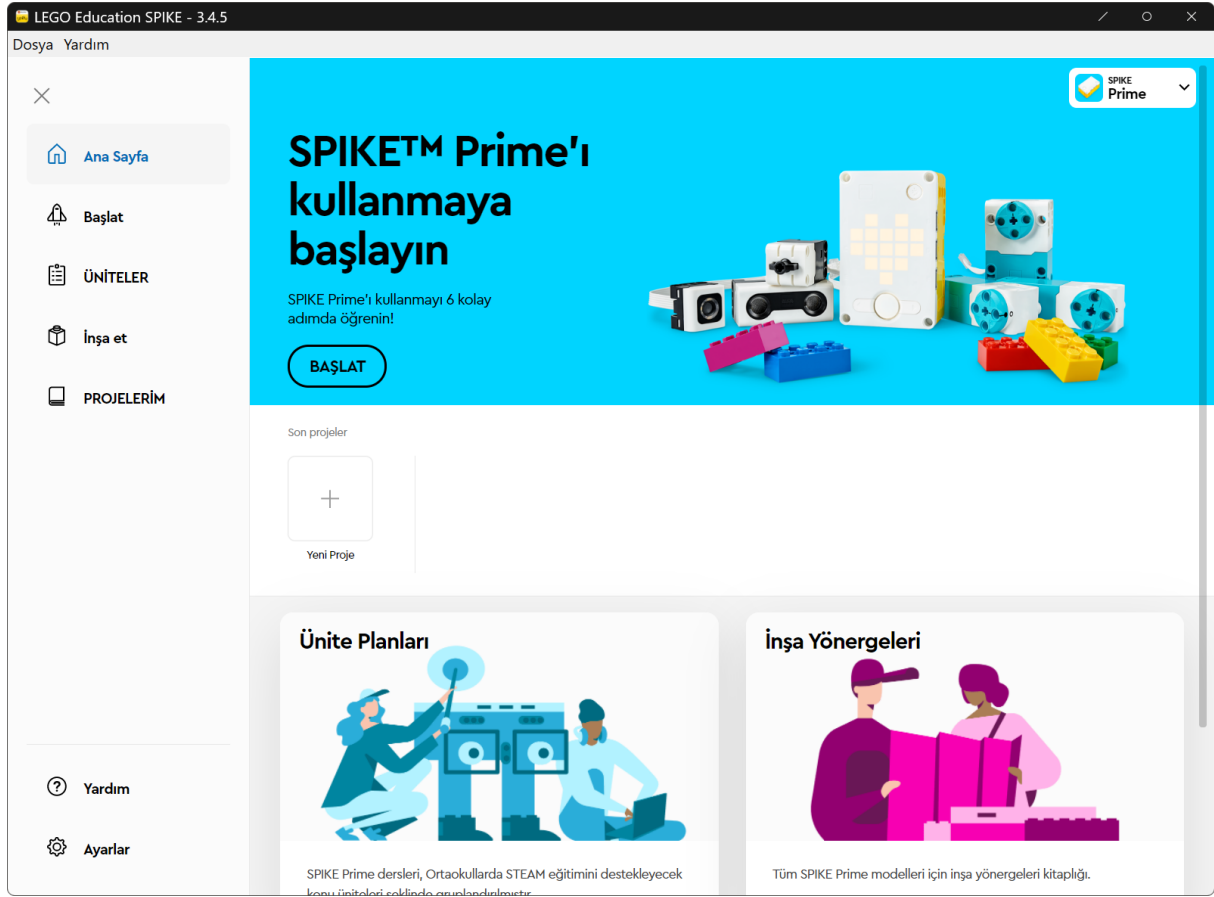
Bu eğitimde LEGO firmasının 2020 yılında piyasaya sürdüğü SPIKE Prime kullanılacaktır. 528 parçadan oluşan bu setin içinde programlanabilir Hub, Mesafe Sensörü, Kuvvet Sensörü, Renk Sensörü, Büyük Motor, iki Orta Motor dâhil birçok teknik parça mevcuttur. SPIKE Prime ve eklenti seti ile inşa edebilen farklı robot tasarımları mümkündür. Aşağıdaki Resim 1.2'de SPIKE Prime robot seti görülmektedir.



Resim 1.2 LEGO® Education SPIKE™ Prime Seti

SPIKE Prime Yazılımı

SPIKE Prime yazılımı robotları programlamak için kullanılan ücretsiz olarak indirilen ve üç farklı programlama ortamını destekleyen bir araçtır. SPIKE Prime yazılımının programlama arayüzünün giriş bölümü aşağıdaki Resim 1.3'te görüldüğü gibidir. Ünite Planlarıyla farklı konularda (İcat takımı, Bir İş Kurma gibi) ders içerikleri sunulmaktadır. İnşa Yönergeleriyle de farklı robotların tasarımlarına örnekler verilmiştir. "Yeni Proje" düğmesi tıklanarak yeni bir proje oluşturulabilir. Bu aşamada kullanıcılara 3 farklı programlama ortamı seçeneği sunulmaktadır. Simgeli Bloklar ve Sözcük Blokları seçenekleri blok tabanlı programlama ortamı ile desteklenmektedir. Sözcük Blokları ile ise yaygın olarak desteklenen sürük ve bırak ortamlarında (Örn. Scratch) programlama yapılabilmektedir. Bu iki ortamda da öğrenciler blok halinde var olan veya kendilerinin oluşturdukları komutları sürük ve bırak yöntemiyle taşıyarak programları oluştururlar. Komutlar bloklar halinde bulunduğu için, öğrencilerin söz dizimini öğrenmek için fazladan zaman ayırmaları gerekmez ve program yazarken hata yapma ihtimalleri ortadan kaldırılır. Bu sayede öğrenciler zihinsel kaynaklarını problem çözmeye yönlendirebilirler. Bu eğitim boyunca "Sözcük Blokları" programlama ortamı tercih edilecektir. Ayrıca, metin tabanlı programlama yaklaşımlarından Python programlama dili de SPIKE Prime tarafından üçüncü seçenek olarak desteklenmektedir. Metin tabanlı programlama yaklaşımı profesyonel programcılar tarafından yıllardır kullanılmaktadır. Bu yaklaşıma alıştıktan sonra karmaşık / ileri seviye programların oluşturulması daha kolay olabilmektedir. Diğer yandan bu ortamda öğrenciler yazacakları her bir komutu en ince detayına kadar tam olarak bilmek zorundadırlar.



Resim 1.3 SPIKE Prime Yazılımı

Eğitimde Kullanılacak Çalışma Alanları

Öğrenciler ders boyunca bazı etkinliklerde çalışma alanları/kâğıtları kullanacaklardır. Bu çalışma alanlarıyla ilgili bilgiler haftalık içeriklerde detaylı bir şekilde verilmektedir.

Kaynakça

- Gürer, M. D., Çetin, I., & Top, E. (2019). Factors affecting students' attitudes toward computer programming. *Informatics in Education*, 18, 281–296.
- P21 (2020). Framework for 21st Century Learning - Battelle for Kids, <http://www.battelleforkids.org/networks/p21> adresinden 22.05.2022 tarihinde erişim sağlanmıştır.
- Üçgül, M., Çetin, İ., Yükseltürk, E. & Top, E. (2021). *Robotik ve Kodlama Ortaokul*. TÜBİTAK Yayınları, Ankara.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical transactions of the royal society of London A: mathematical, physical and engineering sciences*, 366(1881), 3717-3725.
- Wing, J. (2011). Research notebook: Computational thinking—What and why. *The link magazine*, 6, 20-23.
- Yükseltürk, E., & Altıok, S. (2017). An investigation of the effects of programming with Scratch on the preservice IT teachers' self-efficacy perceptions and attitudes towards computer programming. *British Journal of Educational Technology*, 48(3), 789-801.

1. Hafta: Robot ve Algoritma

Haftanın Amacı:

Bu haftanın temel amacı, öğrencileri 'robot' kavramıyla tanıştırmaktır. Ayrıca, örnek robotlar göstererek öğrencilerin ilgisini çekmek ve derse olan motivasyonlarını artırmaktır. Öğrenciler, bu hafta boyunca ders süresince kullanılacak robot setini tanıyacak ve bu seti kullanarak robot tasarımları yapmaya başlayacaklardır. Bunun yanı sıra, algoritma kavramını tanımlayacak ve basit algoritmalar oluşturabileceklerdir.

Haftanın Kazanımları:

- Öğrenciler robot kavramını temel düzeyde açıklar.
- Öğrenciler algoritma kavramını açıklar.
- Öğrenciler basit algoritma örnekleri oluşturabilir.
- Öğrenciler robot setini tanır.
- Öğrenciler robot setiyle yönergeye uygun robot oluşturur.

Kullanılacak Malzemeler:

Robot seti, bilgisayar, kâğıt, kalem

GÖZLE VE UYGULA*Gözle: Temel Kavramlar Üzerine Tartışma*

Rehber öğretmen, öğrencilerin "robot", "algoritma" ve "programlama" gibi kavramlar hakkındaki önbilgilerini öğrenmek ve bu kavramların bütün öğrenciler tarafından tam olarak anlaşılabilmesi için aşağıdaki sorulardan bazılarını sorarak derse başlayabilir.

- Robot nedir?
- Robotların özellikleri nelerdir? Robotlar nerelerde kullanılır?
- Uzaktan kumandalı bir oyuncak ile robot arasında ne gibi farklılıklar vardır?
- Robotlar hangi parçalardan oluşur?
- Algoritma ve program nedir?

Rehber öğretmen; robot, algoritma ve program kavramları için aşağıdaki tanımları dikkate alarak öğrencileri yönlendirebilir veya gerektiği yerde (öğrenciler kavramı yanlış tanımladıklarında veya doğru olandan çok farklı bir tanım yaptıklarında) bu tanımları doğrudan kendisi yapabilir.

Rehber Öğretmen İçeriği – Robot, Algoritma ve Program

Robot: Türk Dil Kurumunun hazırladığı Türkçe Sözlük'te "Belirli bir işi yerine getirmek için manyetizma ile kendisine çeşitli işler yaptırılabilen otomatik araç." şeklinde tanımlanır. Oxford İngilizce Sözlük'te ise "Özellikle bir bilgisayar tarafından programlanan, karmaşık bir dizi işlemi otomatik olarak gerçekleştirebilen makine." şeklinde tanımlanır. Her iki tanım da vurgulanan nokta, robotların bir görevi otomatik olarak yerine getirmesidir. Eğer bir makinenin hareketleri bir insan tarafından kontrol ediliyorsa bu makineye robot denilemez. Ama makine çevresini algılayabiliyor ve buna göre hareket ediyorsa, örneğin bir engeli algılayıp yön değiştirebiliyorsa, bu makine robot olarak adlandırılır. Robotlar çevrelerini sensörleri aracılığıyla algılar. Robotlar kullanılan sensöre bağlı olarak; nesnenin uzaklığı, nesnenin rengi, ışık miktarı, ses şiddeti, nem oranı gibi birçok çevresel veriyi algılayıp işlemcileri ile yorumlayabilir ve programları dâhilinde bu verilere tepkide bulunabilirler. Bu durumu göz önüne alduğumuzda robotu, "çevresini algılayabilen ve algıladığını yorumlayarak bağımsız tepkiler verebilen makine" şeklinde tanımlayabiliriz.

Algoritma: Algoritma, belirli bir problemi çözmek veya belirli bir amaca ulaşmak için çözüm yolunun adımlarını tasarlamaktır. Algoritmalar, yalnızca bilgisayar bilimlerinde değil, hayatın birçok alanında da kullanılır. Örneğin, günlük rutin işlerimizde, yemek yapımında veya basit matematik işlemlerinin çözümünde algoritmalarından yararlanırız. Çay demleme sürecini düşünerek, bu süreçteki adımları şu şekilde sıralayabiliriz: çaydanlığa suyu koy, ocağı aç, suyun kaynamasını bekle, çayı koy, kaynayan suyu üzerine ekle, 15 dakika bekle ve çayı servis et.

Program: Temel olarak bir algoritmanın bilgisayar veya robot için uygulanmasıdır. Programlar bilgisayara veya robota yapması gereken şeyleri (algoritmanın adımlarını) bir programlama dili vasıtası ile aktarılır. Kişi ile bilgisayar arasındaki iletişimi sağlarlar ve geçmişten günümüze farklı programlama dilleri kullanılmaktadır.

Uygula: Algoritma Öğreniyorum

Rehber öğretmen, sayı tahmini oynayacaklarını söyler ve 1 ile 100 arasında rastgele bir sayı tutar. Öğrencilerden bu sayıyı sırayla tahmin etmesini ister. Öğrenciler sayıdan küçük tahminde bulunursa “Yukarı”, büyük tahminde bulunursa “Aşağı” ifadesini kullanır. Bu işlemi doğru tahmin edilinceye kadar devam ettirir. Doğru tahmini yaptığında “Tebrikler” deyip oyunu bitirir. Ayrıca her seferinde sayı isterken kaçınıcı deneme olduğunu da tahtaya yazar. Bu oyunu oynanırken hangi algoritmayı takip ettiklerini tahtaya beraberce yazılır.

1. 1 ile 100 arasında rastgele sayı tut,
2. Tahmin et,
3. Eğer tahmin, tutulan sayıdan küçükse “yukarı”, büyükse “aşağı” mesajını ver,
4. Eğer tahmin, doğru ise “tebrikler” mesajını ver,
5. Kaçınıcı denemede bulunduğunu hesapla.

Başka bir algoritma etkinliğine geçilir. Sınıftaki öğrencilere üzerinde 1 ile 24 arasındaki sayıların yazılı olduğu kartlar rastgele dağıtılır. Bu sayılar veya kartlar artırılabilir. Daha sonra bir öğrenci seçilir. Sadece bu öğrenci duyacak şekilde aşağıdaki algoritma verilir:

- 1) Her bir öğrenciye git,
 - a) Sayısını sor,
 - b) Eğer öğrencinin kartındaki sayı 3’ün katı ise öğrenciyi gruptan ayır ve tahtanın önüne getir.

Seçilen öğrenci diğer öğrencilere sadece kartlarındaki sayıyı sormalıdır ve ipucu oluşturacak cümleler kurmamalıdır. Bütün öğrencilerin kartlarındaki sayılar sorulmasının ardından rehber öğretmen, diğer öğrencilerden, tahtaya çıkarılan öğrencilerin sayılarına bakarak hangi işlemin yapılmış olabileceğini söylemelerini ister. Öğrenciler yapılan işlemi bildikten sonra onlardan bu işlemin algoritmasını yazmaları istenir.

1.3. Uygula: Çarpım Eşleri Algoritması

Öğrencilerden, çarpımları 20 olan çarpım eşlerinin (1-20, 2-10, 4-5) bulunması için bir algoritma yazmaları istenir. Öğrenciler buldukları algoritmaları sınıfça tartışabilirler.

Son olarak algoritmanın bir görevi gerçekleştirmesi için net bir şekilde tanımlanmış ve sıralanmış adımlardan oluşması gerektiği vurgulanır. Eğer görevleri açıkça tanımlanmadıysa veya adımların sıralaması uygun değilse robotun veya bilgisayarın istenilen görevi yerine getiremeyeceği belirtilir.

TASARLA

“Hayalimdeki Robotu Çiziyorum” Etkinliği

Öğrencilerden belirli bir amaç için tasarlanmış bir robot hayal etmeleri ve grup olarak hayal edilen robotun resmini çizmeleri istenir. Daha sonra gruplar çizimleri üzerinden robotlarını arkadaşlarına tanıtır ve robotun ne amaçla kullanılacağını anlatırlar. Bu etkinlik için şu malzemelere ihtiyaç duyulacaktır: Her bir grup için beyaz A3 kâğıt, Boya kalemleri, Makas ve Yapıştırıcı.

Bu aşamada öğrenciler gruplara ayrılır. Her gruptan günlük hayatta karşılaşılan bazı problemleri düşünmeleri, bu problemleri çözebilecek bir robot hayal etmeleri istenir. Daha sonra öğrencilerden hayal ettikleri robotları kâğıtlara çizmeleri ve kesmeleri istenir. Öğrencilere 20 dakika süre verilir. Süre bittikten sonra her grup tek tek tahtaya çıkarak kendi robotunu tanıtır. Hayal edilen robotlar panoya asılır ve eğitim süresi boyunca panoda bırakılır.

ÜRET

“İlk Robotumu Yapıyorum” Etkinliği

Öğrencilere eğitim süresince neler başarabileceklerini göstermek için robot setleri kullanılarak yapılmış robotların videoları gösterilir. Öğrencilere robot setleri ile çalışırken nelere dikkat etmeleri gerektiği ile ilgili bilgilendirme yapılır.

Örnek videolar:

- Robot köpek <https://www.youtube.com/watch?v=mnIUwapCF4>
- Robot kol <https://www.youtube.com/watch?v=61fNK07TcFE>
- Robot akrep <https://www.youtube.com/watch?v=EytUEHudNoc>
- Şifreli kasa <https://www.youtube.com/watch?v=sNZ8-A-Y0L8>
- Küçük top düzeneği <https://www.youtube.com/watch?v=moDpbomZA94>
- Voleybol oyunu <https://www.youtube.com/watch?v=dxJIndGL238>

Rehber öğretmen öğrencilere LEGO Education SPIKE Prime setini tanıtır. Robotu yaparken dikkat edilmesi gereken noktaları belirtir. Parçaların sağlam olduğunu fakat düşme, çarpma gibi durumlar sonucu kırılıp yıpranabilecekleri ifade eder. Parçaların kaybolmaması için robot seti ile gelen veya var olan tasnif kutusunun kullanılmasını önerir.

Rehber Öğretmen İçeriği –Robot Seti İçeriğinin Tanıtılması

Hub: SPIKE Prime sisteminin beyni, programlanabilir Hub’dır. Bu gelişmiş kullanımı kolay tuğla biçimli aygıtın altı giriş/çıkış portu, 5x5 ışık matrisi, Bluetooth bağlantısı, hoparlörü, altı eksenli hareket sensörü ve şarj edilebilir pili bulunur.

Motorlar: Robot seti içerisinde hareketi sağlayan bir büyük ve iki orta motor bulunmaktadır. Büyük boy açılı motor, entegre dönme sensörü ve gerçek düz çizgi kontrolü için mutlak hizalama özelliğine sahiptir ve yüksek güçlü, yüksek torklu uygulamalar için idealdir. Orta boy açılı motor ise, düşük profilli tasarım, mutlak hizalama ve bir derecelik doğruluk özelliğine sahip entegre dönme sensörlüdür.

Sensörler: SPIKE Prime robot setinde farklı sensörler bulunmaktadır. 1-200 cm menzilli, +/- 1 cm doğruluk ve programlanabilir LED gözlere sahip **mesafe sensörü**, 8 rengi ayırt edebilen ve karanlıktan parlak gün ışığına kadar yansıyan ışığı ve ortam ışığını ölçen **renk sensörü** ile doğru ve tekrarlanabilir sonuçlar için 10 Newton’a (~1 kg) kadar basınçları ölçen, basıldığında, bırakıldığında veya çarpıldığında durumlarında kullanılabilen **kuvvet sensörü** bulunmaktadır.

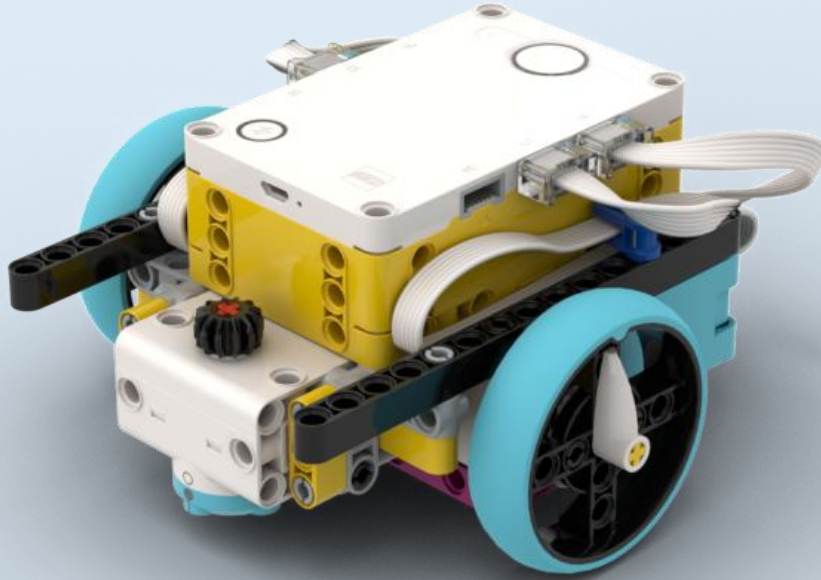
Lego Teknik Parçaları: Robotun farklı şekillerde birleştirilebilmesine olanak sağlayan dişliler, tekerlekler, akslar, kablolar ve diğer lego teknik parçalarını içermektedir.

Her gruba bir robot seti dağıtılır. Beş dakika kadar öğrencilerden robot setlerini incelemeleri, robot setlerindeki parçaları ve ne amaçla kullanılacaklarını keşfetmeleri beklenir. Ardından öğrenciler bu konu hakkındaki düşüncelerini paylaşır. Gerekli noktalarda rehber öğretmen temel parçaların nasıl kullanılacağıyla ilgili daha detaylı bilgiler verir.

Öğrencilerden “Sürüş Modeli 1” yönergesine uyararak ilk robotlarını oluşturmaları istenir. Öğrenciler grup çalışması sırasında görev paylaşımı yapmaları konusunda cesaretlendirilir. Daha sonra öğrencilere robot setleriyle sürüş modeli robotunu yapmaları için yeteri kadar süre verilir.

Not

Sürüş Modeli 1 yönergesine, SPIKE Prime yazılımındaki 'İnşa et' seçeneğinden erişilebilir. Bu robot modeli toplamda 34 adımda inşa edilebilir.



Rehber öğretmen, robotlar oluşturulduktan sonra, daha önce hazırladığı birkaç programı robot üzerinde çalıştırarak gösterir. Bu aşamada amaç, öğrencilere kod yazmayı öğretmek veya yüklenen kodun robot üzerinde nasıl çalıştırılacağını göstermek değildir. Amaç, robota basit bir görev yaptırarak öğrencilerin robota karşı olan ilgilerini artırmaktır.

DEĞERLENDİR

Öğrenciler sınıfın durumuna göre halka oluşturulur veya masalarda oturmaları sağlanır. Rehber öğretmen; robot, robot tasarımı, algoritma ve program konularından birini seçer ve rastgele seçtiği bir öğrenciden verilen kavramın onun için ne ifade ettiğini anlatmasını ister. Bütün konuların anlatılmasının ardından, öğrencilerden, robot tasarlama ve algoritma geliştirme aşamalarında önemli veya ilginç buldukları noktaları arkadaşlarıyla paylaşmaları istenir.

Daha sonra aşağıdaki Resim 1.4 gösterilerek ilgili sorular hakkında öğrencilerin görüşleri alınır.



Resim 1.4 Endüstriyel Robotlar

- Bu robotun görevleri nelerdir ve bu robotu kullanarak hangi sorunları çözebiliriz?
- Sizce mutfak robotu, derste işlediğimiz teknik anlamıyla, gerçek bir robot mudur? Sebeplerini açıklayınız.

İLAVE ETKİNLİK

Algoritma oyunları

Bu oyunda, rehber öğretmen sınıftan ayakkabısı bağcıklı olan iki öğrenci seçer. Seçilen öğrenciler, diğer öğrencilerin görebileceği bir yere sırt sırta oturtulur. Öğrencilerden birer ayakkabılarının bağlarını çözmeleri istenir. Bir öğrenciye, ayakkabısını bağlarken arkadaşına detaylı olarak anlatma görevi verilirken, diğer öğrenciye de arkadaşının anlattığı şekilde kendi ayakkabısını bağlama görevi verilir. Dinleyen öğrenciye, yalnızca arkadaşının anlattığı şekilde adımları yapması ve kendi ilave işlemler yapmaması gerektiği söylenir. Bu oyunda, anlatan öğrencinin kendi ayakkabı bağcığını bağlarken yaptığı adımları arkadaşına aktarırken yeterince açık ve bilgilendirici olup olmadığı, diğer öğrenciler tarafından gözlemlenir.

Bu oyunda, rehber öğretmen kurt-kuzu-ot oyununu oynayacaklarını söyler ve tahtaya kurt, kuzu ve otu çizer. Oyunda bir de nehir vardır ve bunların karşıya geçirilmesi istenir. Ancak karşıya geçebilecek tek araç bir kayıktır ve hepsinin aynı anda karşıya geçmesi imkânsızdır. Kayığa her defasında yalnızca biri alınabilir; ya kuzu, ya kurt ya da ot. Ancak bir sorun daha vardır: Kurtla kuzuyu yalnız bırakırsanız kurt kuzuyu yer, kuzuyla otu yalnız bırakırsanız bu sefer kuzu otları yer. Problemi çözmek isteyen öğrenciler tahtaya davet edilir ve çözüm algoritmasını diğer öğrencilerle paylaşmaları istenir. Örnek bir algoritma şöyle olabilir:

- 1 adım: ilk koyunu karşıya geçirir,
- 2 adım: sonra otu karşıya geçirir,
- 3 adım: sonra koyunu diğer tarafa geçirir,
- 4 adım: sonra kurdu karşıya geçirir,
- 5 adım: sonra koyunu karşıya geçirir.

Bu oyunların oynatılmasındaki amaç, öğrencilerin bir görevi gerçekleştirmek için gerekli adımları net bir şekilde tanımlamaları ve sıralamaları gerektiğini görmelerini sağlamaktır. Öğrenciler görevleri açıkça tanımlamazlarsa veya adımların sıralamasını uygun yapmazlarsa istenilen görevin yerine getirilemediğini oyun sonunda görürler.

2. Hafta - Robotlar ile Hareket

Haftanın Amacı:

Bu haftanın amacı, robot seti ve bu setin programlanacağı SPIKE Prime yazılımının grafiksel arayüzünü tanıtmaktır. Ayrıca öğrencilere, robotlarının belirli bir mesafeyi alabilmesi için gerekli programlama adımlarını öğretmek ve motorların tur, derece ve santimetre cinsinden çalıştırılarak robotun hareket etmesini sağlamaktır.

Haftanın Kazanımları:

- Öğrenciler robot setini programlamak için SPIKE Prime yazılımının grafik arayüzünü kullanır.
- Öğrenciler orta ve büyük motoru tanır.
- Öğrenciler açı, tur sayısı, uzunluk ve çevre kavramlarını robotun hareketini sağlarken kullanır.
- Öğrenciler robotun hareket etmesi için gerekli programlama adımlarını oluşturur.

Kullanılacak Malzemeler:

Robot seti ve bilgisayar.

Ekler:

Konu anlatımında ekran görüntüleri verilen örnek programlar, ayrıca dijital formatta rehber öğretmenlere sunulmuştur. Örnek program dosyalarının numaralandırılmasında, konu anlatımındaki resim numaraları temel alınmıştır. Bu hafta aşağıda sıralanmış programlar ekte rehber öğretmenlere sunulmuştur.

Program_2.5_Temel_Motor_Bloklari.llsp3
Program_2.6_Ileri_ve_Geri_Hareket_Eden_Robot_1.llsp3
Program_2.6_Ileri_ve_Geri_Hareket_Eden_Robot_2.llsp3
Program_2.7_Robotun_Hareketlerine_Ornek.llsp3
Program_2.8_Bir_Motorun_Kullanilmasi.llsp3
Program_2.9_Kare_Seklinde_Hareket_Eden_Robot.llsp3
Program_2.12_Örnek_Program.llsp3
Program_2.14_Lego_Parcasini_Alip_Geri_Getir.llsp3
Program_2.16_Örnek_Program.llsp3

LEGO Education SPIKE Prime Yazılımı

Rehber Öğretmen İçeriği – SPIKE Prime Yazılımı

İlk hafta hazırlanan robot setleri öğrencilere dağıtılır. Hazır bir robot üzerinde robot setinin parçaları gösterilerek öğrencilere parçalar hakkında temel bilgiler verilir. Böylece ilk hafta konuları tekrar edilmiş olur. Daha sonra robot setlerinin programlanması için gerekli yazılımın (LEGO Education SPIKE Prime Yazılımı) bilgisayarlara kurulur.

Not

LEGO Education SPIKE Prime Yazılımına

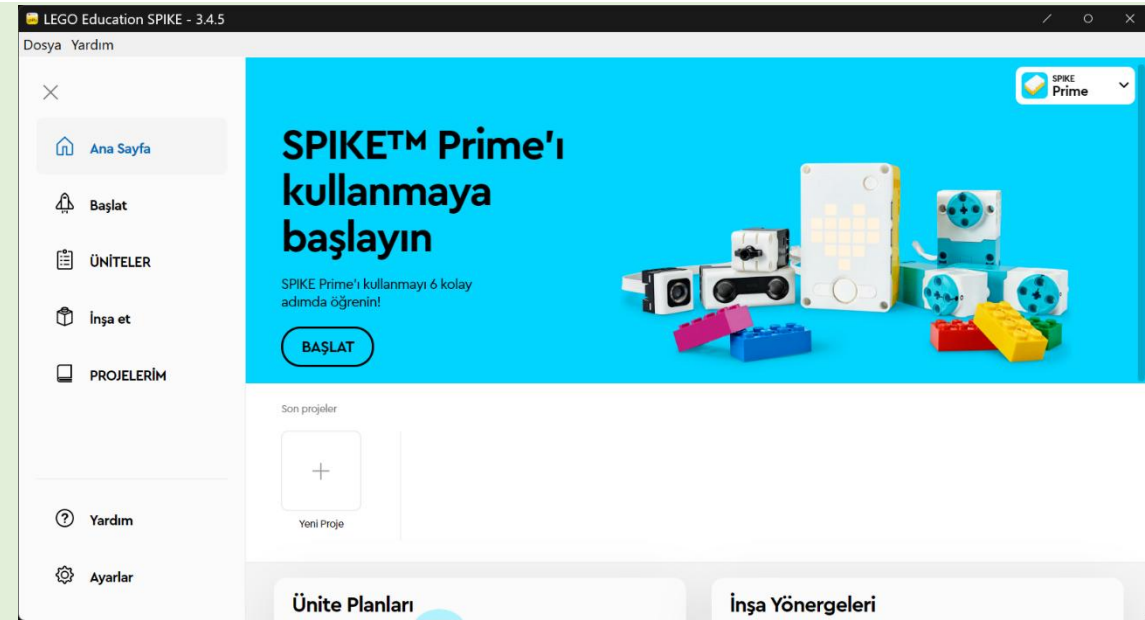
<https://education.lego.com/en-us/downloads/spike-app/software>

adresinden uygun işletim sistemi seçilerek ulaşılabilir.

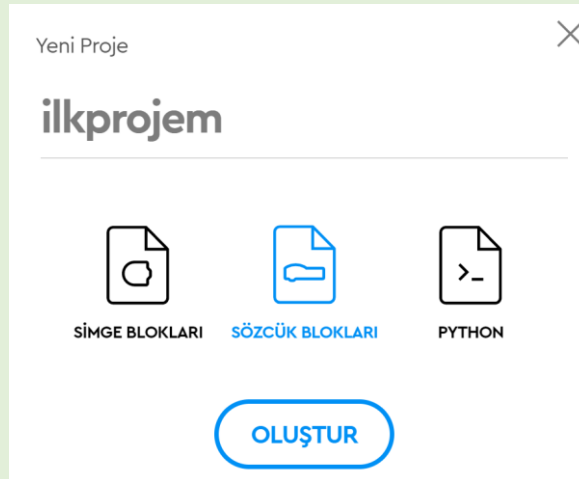
Robot setinin programlanacağı yazılımın kurulumu gerçekleştirildikten sonra indirilen uygulama çalıştırılıp aşağıdaki ekran görüntüsüne ulaşılır. Bu aşamada var olan sete göre seçim yapılır (SPIKE Prime seçilir).



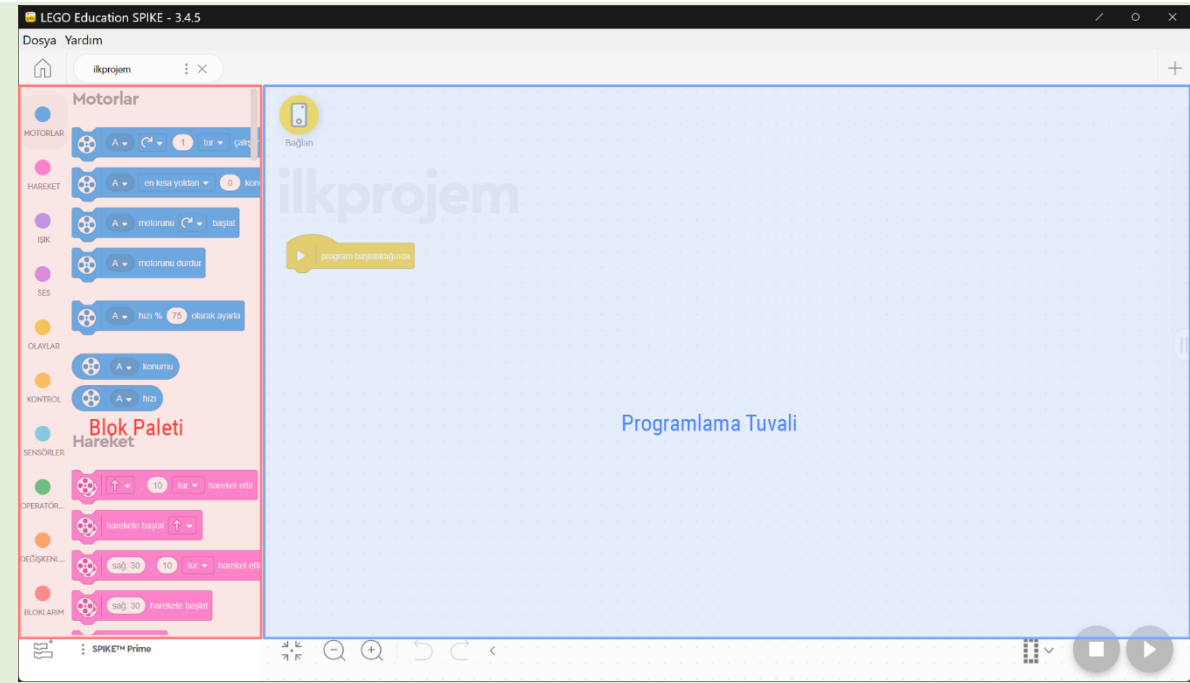
Gelen arayüzde neler olduğu gösterilir. Başlat, Üniteler ve İnşa Et bağlantıları üzerinden Öğretici etkinlikler, Ünite planları ve İnşa yönergeleri hızlıca gösterilir. Ayrıca her bir projemizin saklandığı söylenir ve PROJELERİM bağlantısı ile var olan projelere ulaşılacağı anlatılır. İlerleyen haftalarda bu başlıklardan faydalanacağı belirtilir. Daha sonra yardım menüsünden programlama ekranındaki blokların özelliklerinin anlatıldığı gösterilir. Ayrıca, ayarlar menüsünden de arayüzün özelliklerinin değiştirileceği vurgulanır. Örneğin dil seçeneğiyle Türkçe seçilir.



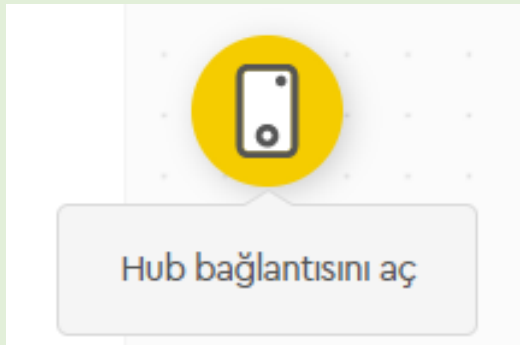
Arayüzden “Yeni Proje” seçilerek ilk uygulamaya başlanır. Gelen ekrandan oluşturulacak projenin adı verilir. Örneğin projenin adı “ilkprojem” olarak giriş yapılır ve Sözcük Blokları seçilerek programlama arayüzüne geçilir.



Yeni proje grafiksel arayüzü (Blok Tabanlı Programlama) açılır ve arayüzde neler olduğu gösterilir. Grafik arayüzü ile Hub arasındaki bağlantının USB veya Bluetooth ile gerçekleştirilebileceği vurgulanır. Önce kablo ile sonra da Bluetooth ile bağlantıların nasıl yapılacağı anlatılır. Daha sonra yazılımının grafiksel arayüzünün bölümleri tanıtılır.



Programlama arayüzü temel olarak iki kısımdan oluşur. Robotun programlanabilmesi için program bloklarının Blok Paletinden Programlama Tuvalinin (Alanının) içerisine sürüklenip bırakılması gerekir. Programlama alanında, varsayılan olarak yukarıdaki resimde görüldüğü gibi “program başlatıldığında” bloğu bulunur. Programlama alanında yaklaştırılan bloklar birbirine yapışır. Böylece sürükle-bırak yaklaşımı ile istenilen programın oluşturulması sağlanır. Programlama blokları, yukarıdaki resimde görüldüğü üzere, Blok Paletinde farklı kategoriler altında gruplanmıştır. Bu kategoriler Motorlar, Hareket, Işık, Ses, Olaylar, Kontrol, Sensörler, Operatörler, Değişkenler ve Bloklarım'dır.



Arayüzdeki sol üst bölümde bulunan “Hub Bağlantısını Aç” özelliği ile Hub’a bağlı olan motorlar ve sensörler görülebilir, ayrıca daha önce yazdığımız programlar yönetilebilir. Programlama tuvalinin sağ altındaki oynat düğmesi (sarı düğme) ile hazırlanan programları Hub’a göndererek çalıştırabilir. “Depolama Konumu Seçin” özelliği ile programların hangi alana yüklenebileceğine karar verilebilir. Hub, sıfırdan başlayarak 19’a kadar toplamda 20 adet programı hafızasında tutabilir.

GÖZLE VE UYGULA

Gözle: Programlama Alanının Kullanılması

Robot, istenilen program bloğunun programlama alanına sürüklenip bırakılması ve blokların parametrelerinin değiştirilmesi ile programlanır. Program, blokların programlama tuvalinde yer alan “program başlatıldığında” bloğunun altına birbiri ardına eklenmesiyle oluşturulur. Program çalıştırıldığında, robot, “program başlatıldığında” bloğundan başlayarak program bloklarında belirlenen görevleri sırasıyla gerçekleştirir. Aşağıdaki resimde blokların örnek bir yerleştirilmesi görülmektedir.



Resim 2.1 Blokların Yerleştirilmesi

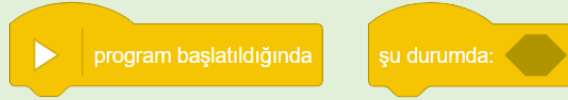
Robotun birden fazla işlemi aynı anda gerçekleştirmesi isteniyorsa, programlama alanında birden fazla şapka bloğu kullanılarak, iki veya daha fazla programın eş zamanlı çalışması sağlanabilir. Paralel işlemlere bir örnek Resim 2.2’de sunulmuştur. E motoru çalışırken ederken Hub ışığının rengi de değiştirilmektedir. Paralel işlemlerde farklı yığıtlarda, birbiri ile çelişen görevler verilmemelidir.



Resim 2.2 Paralel İşlemler

Rehber Öğretmen İçeriği – Blok Türleri**Şapka Blokları**

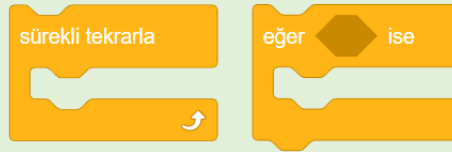
Şapka blokları, her diziyi başlatan bloklardır. Yuvarlak bir tepe ve altında bir çıkıntı şeklindedirler. Bu nedenle sadece altına blok eklenebilir. Örnek şapka blokları:

**Yığıt Blokları**

Yığıt blokları, ana komutları gerçekleştiren bloklardır. Üstlerinde bir çentik ve altlarında bir tümsek vardır. Bunlar motorları hareket ettiren ve ışıkları yakan bloklardır. Örnek bir yığıt bloğu:

**C Blokları**

Bu bloklar C şekilli bloklardır ve tamamı kontrol kategorisinde bulunur. C'lerin içindeki blokları bir döngüye sokar veya bir koşulun "doğru" olup olmadığını kontrol eder. Örnek C blokları:

**Haberci Bloklar**

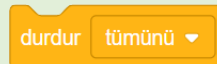
Bu bloklar, bir rakam veya dizi olabilecek değerleri içerir. Bir sensör değerini de içerebilir veya bir değişken değerini de taşıyabilir. Örnek haberci blokları aşağıda verilmiştir.

**Bloole Blokları**

Bu bloklar koşul belirtirler, ya doğru ya da yanlışlar. Altıgen şekle sahiptirler. Örnek bloole blokları aşağıda verilmiştir.

**Durdurma Blokları**

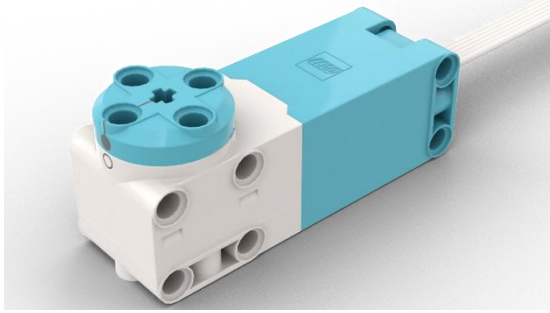
Bu bloklar komut dizilerini sonlandırır. Üstleri girintili ve altları düzdür. Bu sebeple altlarına herhangi bir blok yerleştirilemez. Örnek bir durdurma bloğu:

**Programlama Yığıtı**

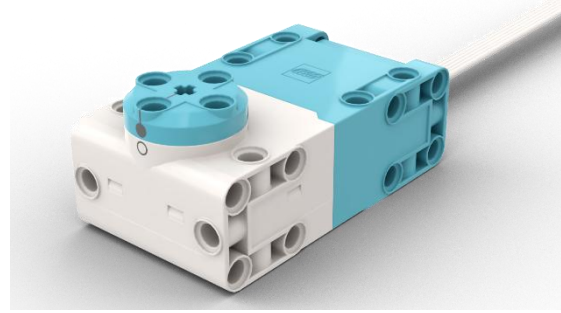
Bir araya getirilmiş bir blok dizisidir.

Gözele: Robotun Hareket Ettirilmesi

SPIKE Prime robot setinde robotu hareket ettiren iki çeşit motor bulunur. Bunlar, Resim 2.3'te görüldüğü gibi orta (medium) ve büyük (large) motor olarak adlandırılır. Set içerisinde bir büyük motor ve iki orta motor mevcuttur. Büyük motor, entegre dönme sensörü ve gerçek düz çizgi kontrolü için mutlak hizalama özelliğine sahiptir; bu nedenle yüksek güçlü ve yüksek torklu uygulamalar için idealdir. Orta motorlar ise hızlı tepki veren robot tasarımlarında tercih edilir. Ayrıca, mutlak hizalama ve 1 derecelik doğruluk özelliğine sahip entegre dönme sensörüne sahiptir.



Orta Motor



Büyük Motor

Resim 2.3 Orta ve Büyük Motor

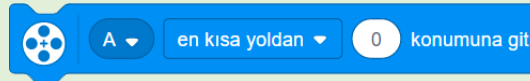
Rehber Öğretmen İçeriği – Motorlar ve Hareket Blokları

Robotu hareket ettirecek program blokları Motorlar ve Hareket blokları sekmesinde yer alır. Motor blokları ya motorların çalışmasını sağlar ya da motorlardan bilgi alır. Hareket blokları ise, iki motoru senkronize çalışmasını sağlar. Bu hafta sık kullanacağımız bazı bloklarla ilgili bilgiler aşağıda verilmiştir, bloklarla ilgili daha fazla bilgiye yardım menüsünden ulaşabilirsiniz.

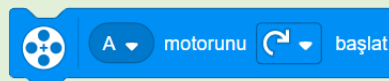
Motoru Belirli Süre Çalıştır bloğu, seçilen bir veya birden fazla motoru saat yönünde ya da saat yönünün tersine, belirli bir tur, saniye veya derece çalıştırır.



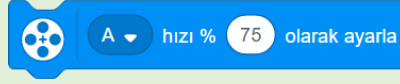
Motor Konumuna Git bloğu, bir veya daha fazla motoru belirli bir konuma göre ayarlar. Motor, saat yönünde ya da saat yönünün tersine çalışacak şekilde veya belirtilen konuma en kısa yoldan gidecek şekilde ayarlanabilir. Konum aralığı 0-359 derece arasındadır.



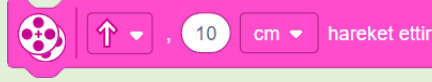
Motoru Başlat bloğu ile bir veya daha fazla motoru sürekli olarak saat yönünde ya da saat yönünün tersine çalıştırır.



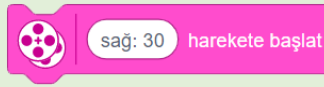
Motor Hızını Ayarla bloğu ile bir veya daha fazla motorun hızını -100 ile 100 arasında ayarlanabilir. Eksi değerler, motoru ters yönde çalıştırır. Varsayılan hız %75'tir.



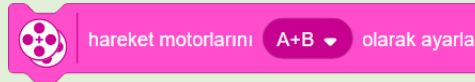
Belirli Süreyle Hareket Ettir bloğu, sürüş modelini belirli bir santimetre, inç, saniye, derece veya tur cinsinden saat yönünde veya saat yönünün tersine döndürür.



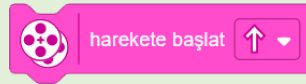
Direksiyonla Harekete Başla bloğu, sürüş modelini yönlendirme olanağıyla ileri hareket ettirmeye başlar.



Hareket Motorlarını Ayarla bloğu ile iki sürüş motorunun bağlanacağı portları tanımlar. Portlar tanımlanmazsa robot hareket etmez.



Hareket Başla bloğu, sürüş modelini ileri veya geri hareket ettirmeye başlar.



Belirli Bir Süre Direksiyonla Hareket Ettir bloğu, sürüş modelini belirli bir tur, derece, saniye ileri hareket ettirir. Düz bir çizgide gitmesi için "0", kendi çevresinde dönmesi için "100" ve "-100" arasındaki değerler kullanılır.



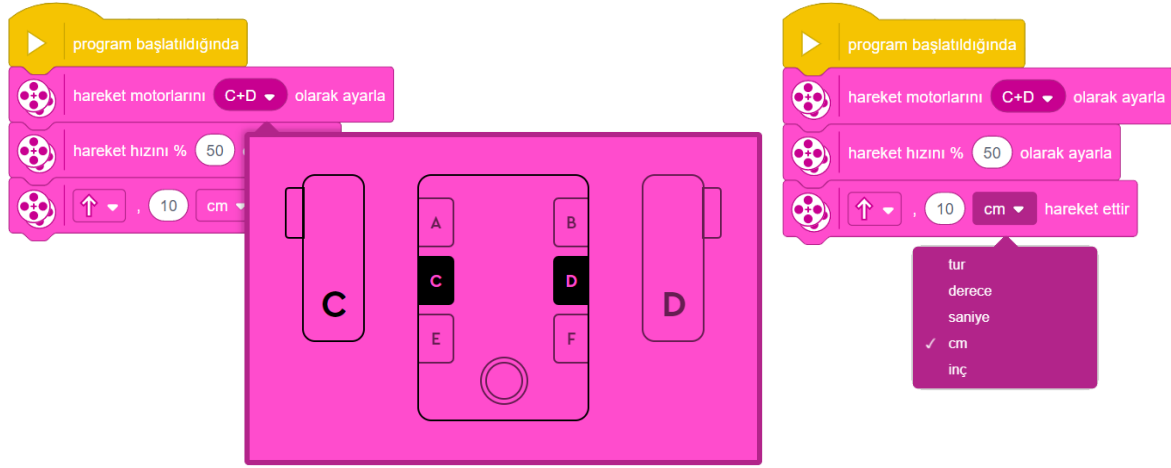
Not

Öğrencilerin, önceki hafta montajını tamamladıkları 'Sürüş Modeli 1' robotunu, yeterli düzeyde şarj edilmiş olarak sınıfta bulundurmaları gerekmektedir. Bu, robotun etkin bir şekilde kullanılabilmesi için önemlidir.

Gözle: Robotun Temel Hareketleri

Robotun iki motorunun aynı anda hareket ettirilmesi, motorların pozitif veya negatif yönde dönmesini ve dolayısıyla robotun ileri veya geri gitmesini sağlar. İlgili bloğun özelliği sayesinde (örneğin, Resim 2.4'te sol bölümde C+D tıklanığında) hangi portlara bağlı motorların kontrol

edileceği görülebilir. Eğer motorlar farklı portlara bağlanmışsa, bu bölümden portlar değiştirilebilir.



Resim 2.4 Temel Hareket Bloğu ve Ayarlanması

Hareket bloklarından motorlar seçildikten sonra, hızları ayarlanabilir. Resim 2.4'ün sağındaki bölümde olduğu gibi, motorlar %50 hızla hareket edecek şekilde ayarlanabilir ve ileri hareket etmesi için cm, inç, tur, derece ve saniye gibi özellikler belirlenebilir.



Resim 2.5 Temel Motor Blokları ve Ayarlanması

Resim 2.5'teki kod bloğu ile robot 10 cm ileri gider. Ardından, yalnızca C motoru bir tur çalıştığı için D tekerleği merkezinde robot kendi etrafında 180 derece döner. Motor seçimi, yönü ve tur, derece ile saniye gibi özelliklerin değiştirilmesi, robotun dönüş yönü ve miktarını belirler.

Not

Özellikle motorların dönerken seçilen yönleri aynı olsa bile, motorların montajına göre robotun hareketine farklı etkileri olabileceği vurgulanmalıdır. Örneğin, aşağıdaki program çalıştırılarak robotun hareketi gözlemlenmeli ve bu durum öğrencilerle tartışılarak oluşabilecek yanlış algıların önlenmesi sağlanmalıdır.

**Gözle ve Uygula: İleri ve Geri Hareket Eden Robot**

Direksiyon hareketi bloğu anlatıldıktan sonra robotun bir miktar düz gidip kendi etrafında dönmeden geri geldiği bir program çalıştırılıp gösterilir.



Resim 2.6 İleri ve Geri Hareket Eden Robot (iki alternatif yöntemle)

Öğrencilerden benzer bir uygulamayı bu seferde kendi etrafında dönerek tur, saniye ve derece seçeneklerini kullanarak yapmaları istenir.



Resim 2.7 Robotun Hareketlerine Başka Bir Örnek

Ayrıca öğrencilerden sadece bir motoru kullanarak dönüş yapıp geri gelen programı da yazmaları beklenir.



Resim 2.8 Bir Motorun Kullanılması

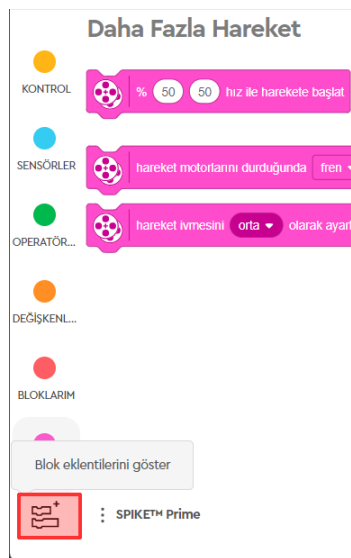
Uygula: Kare Çizen Robot

Motor bloklarını ve hareket bloklarını kullanarak robotun dönüşünde meydana gelen değişiklikleri gözlemleyen öğrencilerden, kare şeklinde bir rotada hareket eden robotun programını oluşturmaları istenir. Bu etkinlik sırasında, öğrencilerin robotun bulunduğu yerde veya tek tekerlekli hareketle dönebilmesi için gerekli düzenlemeleri keşfetmeleri sağlanır.



Resim 2.9 Kare Şeklinde Rotada Hareket Eden Robot

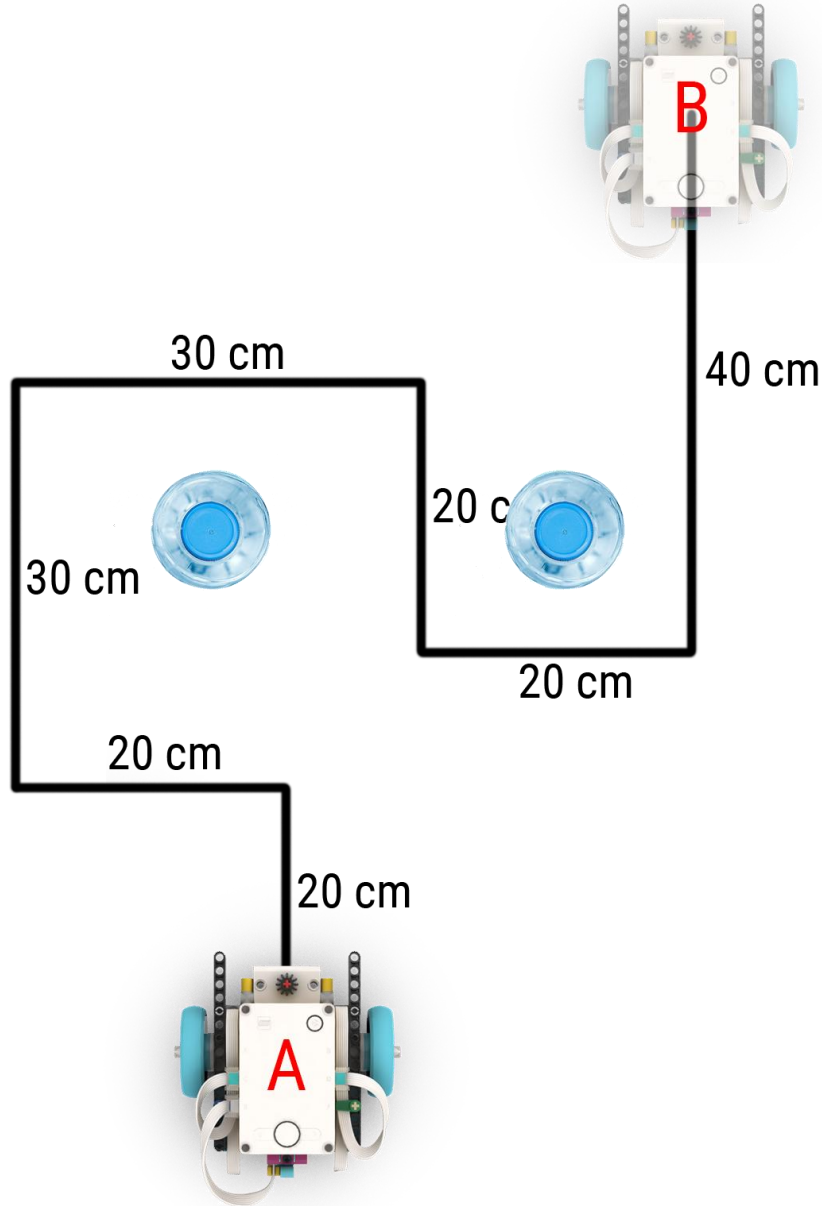
Kare şeklindeki bir rotada hareket eden robotun programlanması, ilerleyen haftalarda döngü konusu anlatıldıktan sonra daha az kod bloğu kullanılarak gerçekleştirilebilecektir. Ayrıca, bu aşamada 'Daha Fazla Motor' ve 'Daha Fazla Hareket' eklentilerinin de kullanılabileceği belirtilmelidir. İlgili eklentileri seçmek için sol alt köşede bulunan 'Blok eklentilerini göster' düğmesine tıklanması gerekmektedir. Bu eklentiler, temel bloklara ek olarak bazı ekstra özellikler sunarak programlama sürecini zenginleştirecektir.



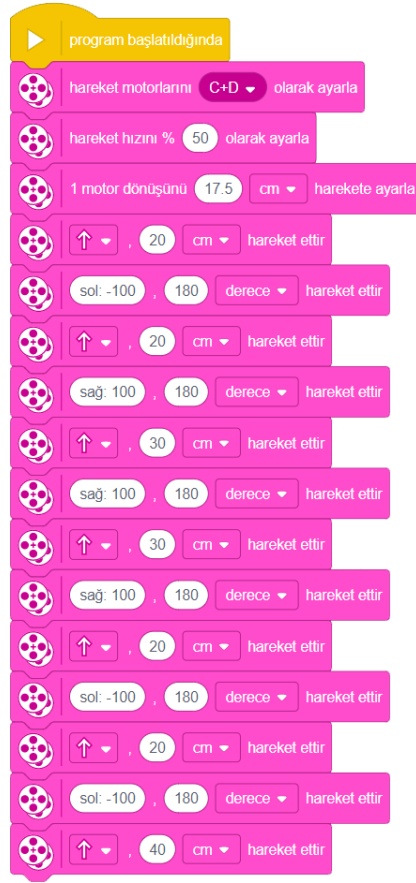
Resim 2.10 Daha Fazla Motor ve Daha Fazla Hareket Eklentileri

Uygula: A noktasından B noktasına Hareket

Bu etkinlikte, robotun Resim 2.11'de görüldüğü gibi A noktasından B noktasına gitmesi için gerekli programın oluşturulması istenir. Robotun rotası üzerinde boş pet şişeler bulunduğu görülmektedir. Robotun şişelerin çevresinden dolaşılması gerekmektedir. Ayrıca, robotun rotası üzerinde gideceği uzaklıklar belirtilmiştir.



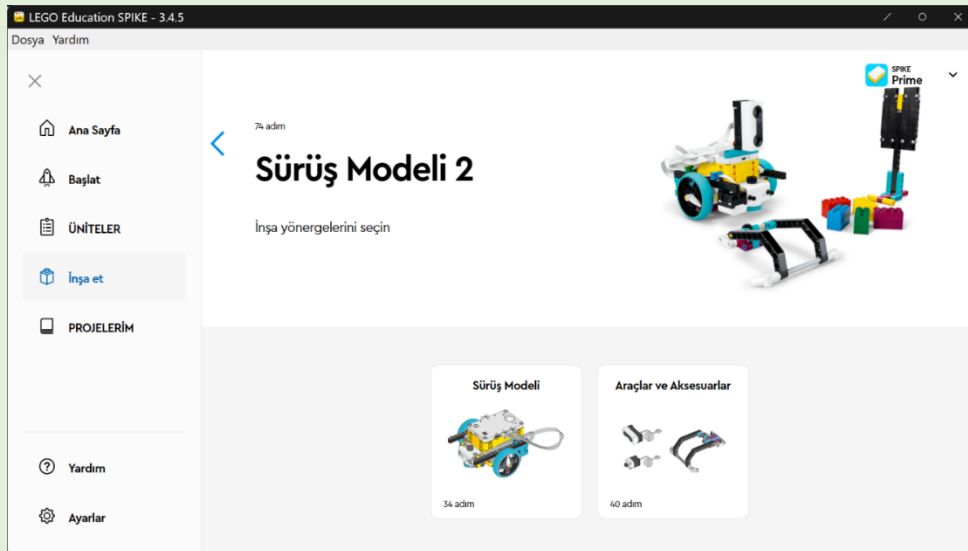
Resim 2.11 Robotun Rotası



Resim 2.12 Örnek Program

Rehber Öğretmen İçeriği – Farklı İnşa Yönergeleri

SPIKE Prime yazılımının 'İnşa Et' veya 'İnşa Yönergeleri' bölümlerinden farklı robot modellerinin nasıl tasarlanacağı incelenebilir. Örneğin, bu aşamada Sürüş Modeli 2 adımlarını takip ederek öğrenciler, istedikleri aksesuarları robotlarına ekleyebilirler. Sensörler ilerleyen haftalarda kullanılacağından, bu hafta yalnızca aksesuar eklemeleri yeterli olacaktır.



Öğrenciler, belirtilen adımları izleyerek büyük motor ile hareket eden kolu monte edebilirler.

TASARLA

Gün içerisinde robotları hareket ettirmeyi öğrenen öğrencilerden, bu bilgileri kullanarak robotlarının yeni bir görevi tamamlamaları istenir. Örneğin, robotlarının belirli bir mesafe düz ilerleyip durmasını, sonra robotlarına takılı olan kolu indirip LEGO parçaları ile yapılmış bir kutuyu (Resim 2.13) tutarak kendi etrafında 180 derece dönmesini ve başlangıç noktasına getirmesini sağlayan bir program tasarlamaları istenir. Tasarlama aşamasında öğrencilerden aşağıda örnek olarak verilen “Tanımlama” ve “Fikir Üretme” süreçlerini sırayla gerçekleştirmeleri sağlanır.

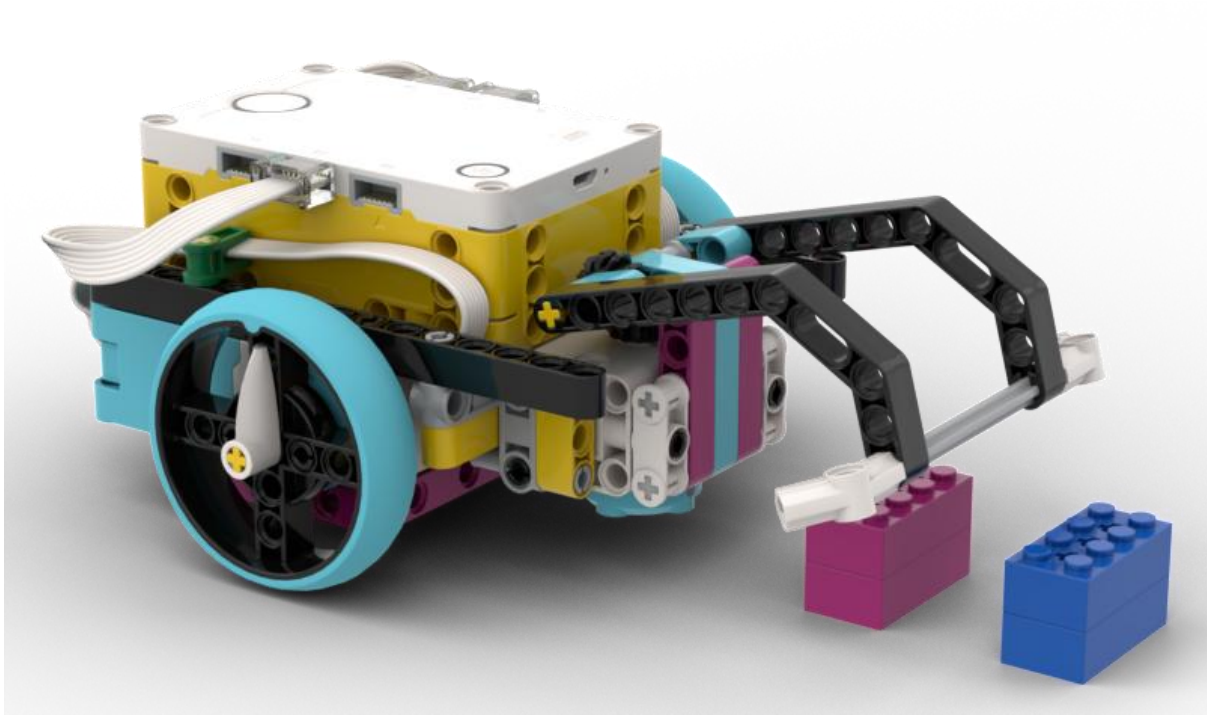
Tanımlama: Bu aşamada öğrencilerden kâğıt ve kalem kullanarak programın algoritmasını veya akış diyagramını hazırlamaları istenir. Öncelikle robotun istenilen hareketleri yapabilmesi için neler gerektiğini belirlemeleri ve maddeler hâlinde yazmaları beklenir. Zorlanan öğrencilere rehber öğretmen aşağıdaki adımları yapabileceğini söyleyerek yardımcı olabilir.

Örneğin robot;

- 20 cm ileri gidecek ve duracak,
- Kolunu kutunun üzerine indirecek,
- Kendi etrafında kutuyla birlikte dönecek,
- Başlangıç noktasına geri dönecek.

Fikir üretme: Bu aşamada öğrencilerin tanımlamada belirlenen işlemlerin nasıl yapılabileceği ile ilgili fikir yürütmesi beklenir. Örnek olarak öğrenciler aşağıdaki maddelere benzer fikirler üretebilir:

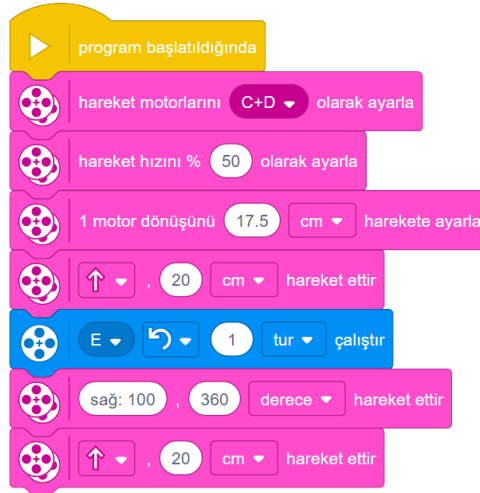
- Öncelikle robotun hangi hareketleri yapacağı planlanmalıdır. Planlanan her bir hareketin tanımı yapılmalıdır (Bu aşamada öğrenciler hareketlerin programlanmasını hızlıca deneyebilirler):
 - İlk harekette 20 cm ileri gidip durur
 - Etrafında 180 derece döner.
- Robot kolunun indirilmesi için büyük motor kullanılır.
- Kolun kutuyu kavraması sağlanır.
- Robotun kendi etrafında 180 derece dönmesi sağlanır.
- Robotun başlangıç noktasına geri gelmesi sağlanır.



Resim 2.13 Lego Parçasını Alıp Geri Getiren Robot

ÜRET

Bu bölümde öğrenciler aktif rol üstlenerek verilen problemi çözerek, robotu tasarlayıp programlamayı tamamlarlar. Rehber öğretmen öğrencilere yalnızca zorlandıkları noktalarda destek olur. Öğrenciler bilgisayarda ve robot üzerinde çalışarak gerekli yazılım çözümlerini geliştirirler. Örnek bir çözüm Resim 2.14'te verilmiştir.



Resim 2.14 Lego Parçasını Alıp Geri Getiren Örnek Program

DEĞERLENDİR

Günün sonunda öğrencilerle aşağıdaki sorular üzerinden bir tartışma yürütülür:

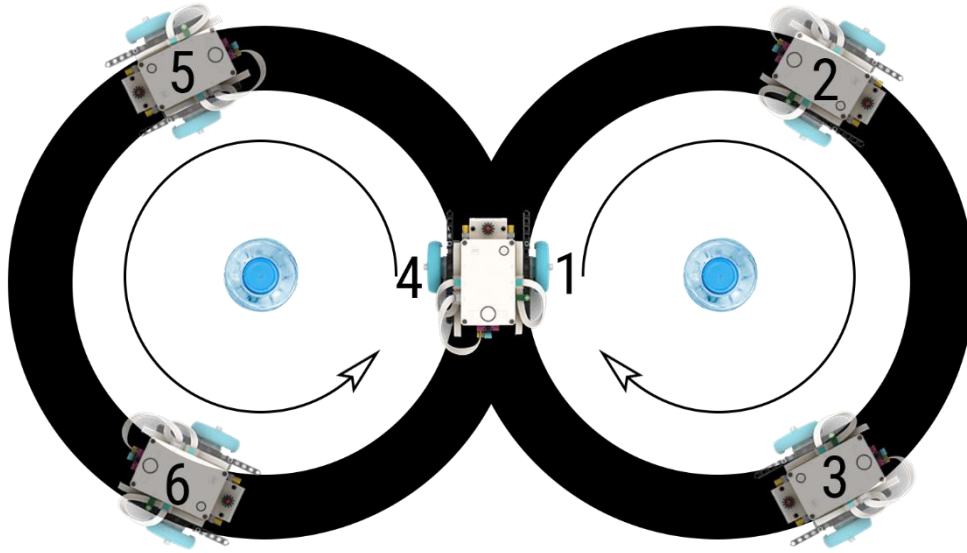
- Robotun hızının değiştirilmesi sonucunda neler oldu? (Hız-zaman ilişkisi tartışılır)
- Motor ve hareket bloklarının kullanarak robotların hareket ettirilmesinde ne gibi farklılıklar oluşuyor?
- Daha Fazla Blok Eklentilerini neden kullanma ihtiyacı hissettiniz?
- Sizce robotun hareket ettirilmesi için teker büyüklüğünün bir önemi var mıdır?
- Programınızda tur sayısını değiştirmeden sadece teker büyüklüğünün değiştirilmesi durumunda ne gibi sonuçlara yol açar?

Bu soruların cevaplarına göre farklı robotların da tasarlanabileceğine dair örnekler verilebilir.

İLAVE ETKİNLİK

Sonsuz İşareti Şeklinde Hareket Etme

Bu ilave etkinlik zaman kalması durumunda yapılabilir. Amaç, motor ve hareket bloklarını kullanarak robotların hareket etmesini farklı bir örnek üzerinde yapılmasının sağlanmasıdır. Bu ilave etkinlik için iki nesne (örneğin iki boş su şişesi olabilir) sınıf içinde müsait bir alana, aralarında 40-50 cm olacak şekilde yerleştirilir. Robotun bu nesnelere dokunmadan etraflarında "8" veya "sonsuz sembolü" çizerek hareket etmesi sağlanır. Robotun izleyeceği yol Resim 2.15'deki gibi olabilir.



Resim 2.15 Örnek Rota

Etkinliği zenginleştirmek için iki nesne arasındaki mesafe artırılabilir ya da azaltılabilir. Nesne sayısı da artırılabilir veya farklı nesneler yerleştirilebilir. Ayrıca robotun başlangıç noktasına göre algoritma ve tasarım değişiklik gösterebilir. Programı oluşturmaya başlamadan önce grupların tasarlama adımı için yukarıda bir örneği verilen tanımlama ve fikir üretme süreçlerini gerçekleştirmeleri gerekir (Rehber öğretmen kendisine verilen aşağıdaki örnek çözümü inceleyebilir ama çözümü öğrencilerle paylaşmamalıdır, görevi öğrencilerin kendilerinin yapmasına izin vermelidir).



Resim 2.16 Örnek Program

3. Hafta – Işık, Ses ve Müzik

Haftanın Amacı:

Bu hafta Hub ekranının özelliklerinin tanıtılması, ekranın ışık ve parlaklığının çeşitli desenlerle programlanması, Hub ve kullanılan cihazlar ile çalınacak seslerin programlanması ve çeşitli enstrümanların kullanılarak müzik eserlerinin programlanması hedeflenmiştir. Ayrıca, Hub ekranının programlanması ve seslerin robotun hareketi ile senkronize olması için ilgili kod bloklarının tanıtılması ve ilgili problemlerin çözümünde kullanılması amaçlanmıştır.

Haftanın Kazanımları:

- Hub ekranının özelliklerini ifade edebilir.
- Hub ekranının istenilen şekilde görünmesi için gerekli blokları program akışında kullanabilir.
- İstenilen seslerin oluşturulması için gerekli blokları program akışında kullanabilir.
- İstenilen müziklerin oluşturulması için gerekli blokları program akışında kullanabilir.
- Hub ekranın görüntüsünün düzenlenmesi, ilgili seslerin/müziğin oluşturulması ve robotun senkronize bir şekilde hareket edebilmesi için gerekli blokları program akışında kullanabilir.

Kullanılacak Malzemeler:

Robot seti ve bilgisayar.

Ekler:

Bu hafta aşağıda sıralanan programlar ekte rehber öğretmenlere sunulmuştur.

Program_3.3_Sonsuz_Dongu_ve_Donguyu_Tekrarla.llsp3

Program_3.4_Basla_ve_Geri_Sayim.llsp3

Program_3.5_Ilerlerken_Ses_Cikaran.llsp3

Program_3.6_Kare_Cizen_Robot.llsp3

Program_3.8_Ceviz_Adam.llsp3

Program_3.9_Beethoven_9._Senfoni_(Neseye_Ovgu).llsp3

Program_3.11_Dans_Eden_Robot.llsp3

Nota_Tempo_Vurus_Degerleri.pdf

GÖZLE VE UYGULA

IŞIK

Rehber Öğretmen İçeriği – Işık Bloğu

Hub ekranı, 5 satır ve 5 sütundan oluşan toplam 25 pikselden meydana gelir. Her bir piksel ayrı ayrı programlanabilir ve piksellerin parlaklığı 10 farklı seviyede ayarlanabilir. Orta düğme ışığı ise 11 farklı renkte yanabilir. Ayrıca, program aracılığıyla girilen metinler, akan metin şeklinde Hub ekranında görüntülenebilir.

Hub'ın ekranını programlamak için blok paletindeki "Işık Blokları" kategorisi kullanılır. "Işık Blokları"nda piksellerin açılması/kapatılması, piksellerin belirlenen sürede açılması, ekrana metin yazılması, piksellerin parlaklığının yüzde olarak ayarlanması, belirli bir pikselin parlaklığının ayarlanması, Hub ekran yönünün değiştirilmesi ve mesafe sensörünün ışıklarının düzenlemesi işlemleri için bloklar bulunur. Bu hafta etkinliklerde kullanılacak yeni bloklar ve her bir bloğun açıklaması aşağıda verilmiştir.

Işık Blok Paleti

Farklı parçaların ışıklarını açma, kapatma ve yoğunluğu belirleme işlemlerine olanak sağlar.

Işık Matrisini Belirli Saniyede Aç

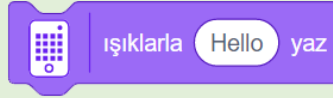
Hub ekranındaki piksellerin her birinin ışık miktarını ve ekranda görüneceği süreyi belirlemek için kullanılır. Belirtilen süre dolduğunda pikselleri kapatır.

**Işık Matrisini Aç**

Hub ekranındaki piksellerin her birinin ışık miktarını belirlemek için kullanılır. Başka bir işlem yapılmadığı sürece oluşturulan desen ekranda görünür.

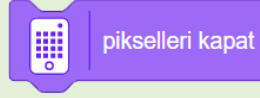
**Işıklarla Yaz**

İçerisine yazılan metni, ekranda bir seferde yalnızca bir harf olacak şekilde ışık matrisinde görüntülemek için kullanılır.



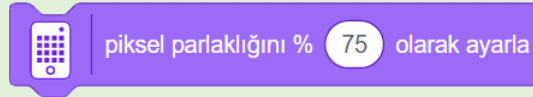
Pikselleri Kapat

Işık matrisindeki tüm ışıkları kapatmak için kullanılır.



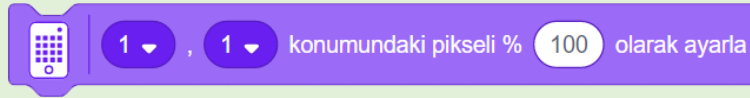
Matris Parlaklığını Ayarla

Işık matrisini kullanacak sonraki bloklar için ışık matrisinin parlaklığını ayarlamak için kullanılır. Parlaklık belirtilmediyse varsayılan değer %100'dür.



Piksel Parlaklığını Ayarla

Belirtilen pikselin parlaklığını ayarlamak için kullanılır, diğer pikselleri etkilemez.

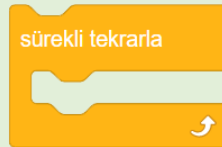


Kontrol Blokları Paleti

Bekle yapıları, döngüler ve koşullar gibi blok yürütmenin doğal akışını değiştirebilecek blokları içerir.

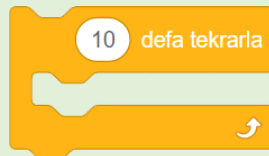
Sonsuz Döngü

İçinde yer alan blokları sırayla sürekli çalıştırmak için kullanılır. Robotun orta düğmesi veya "Tümünü Durdur" bloğu kullanılarak durdurulur.



Döngüyü Tekrarla

İçinde yer alan blokları sırayla ve belirlenen sayıda çalıştırmak için kullanılır. Döngüden sonra blok varsa çalıştırmaya devam eder yoksa program durur.

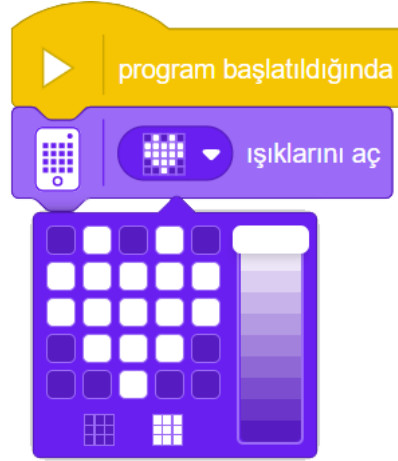


Göze: Hub Ekranı

Rehber öğretmen, Hub ekranının 5 satır ve 5 sütundan oluşan toplam 25 pikselden meydana geldiğini ve her bir pikselin ayrı ayrı programlanabildiğini ifade eder. Her pikselin yalnızca beyaz renkte yanabildiğini, parlaklığının 10 farklı seviyede ayarlanabildiğini ve orta düğme ışığının 11 farklı renkte yanabileceğini belirtir. Ayrıca, program aracılığıyla girilen metinlerin akan metin şeklinde Hub ekranında görüntülenebildiğini de aktarır.

“Işık Blok” paketinde piksellerin açılması/kapatılması, piksellerin belirlenen sürede açılması, Hub ekranına metin yazılması, piksellerin parlaklığının yüzde olarak ayarlanması, belirli bir pikselin parlaklığının ayarlanması, Hub ekran yönünün değiştirilmesi ve mesafe sensörünün ışıklarının düzenlenmesi işlemleri için blokların bulunduğunu anlatır.

Rehber öğretmen, öğrencilere öncelikle Hub ekranında kalp görüntüsü oluşturacaklarını söyler. Bunun için “Işık Blok” paletinden “pikselleri aç” bloğunun programlama tuvalindeki “program başlatıldığında” bloğunun altına sürüklenmesi ve piksellerin ışığının tek tek ayarlanması gerektiğini gösterir (Resim 3.1). Açık olan piksellerin en yüksek parlaklığa getirilmesini ister.



Resim 3.1 Işık Matrisini Aç Bloğu

Uygula: Büyüyen Kalp Efektı

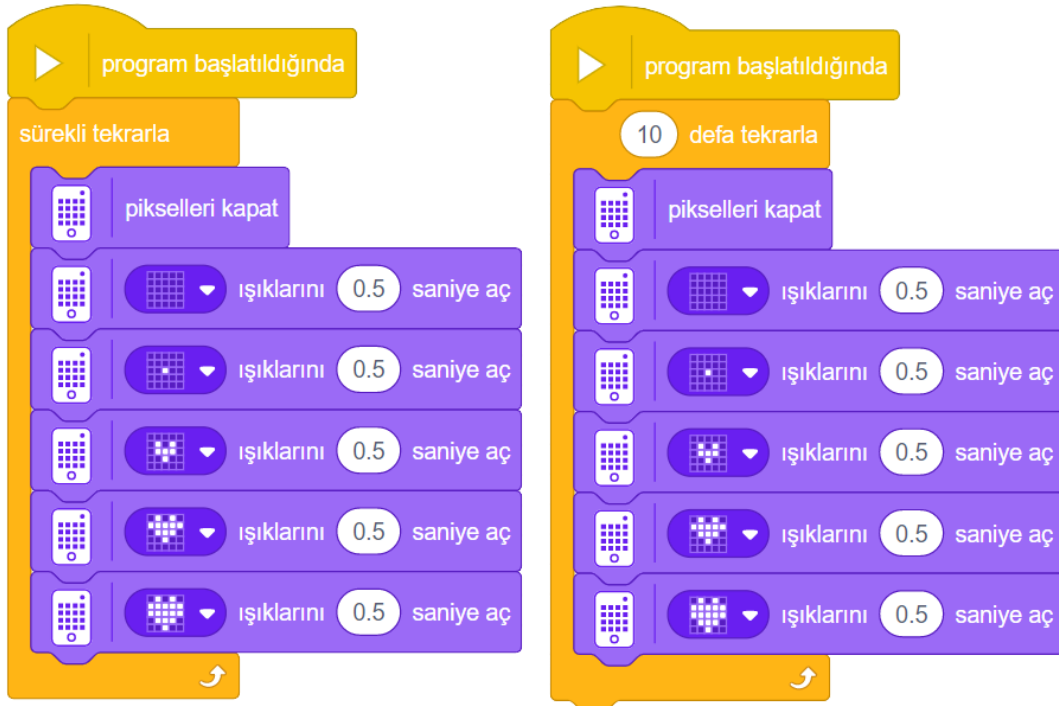
Rehber öğretmen öğrencilerden, bütün piksellerin ışığının kapalı olarak başlayıp yavaş yavaş büyüyerek bütün ekranı kaplayan bir kalbe dönüşen program akışını yapmalarını ister. Ortadaki pikselden başlayarak ve ekran tasarımları arasında belirli bir süre ekleyerek büyüyen kalp efekti yapılabilir. Örnek çözüm Resim 3.2’deki gibi olabilir.



Resim 3.2 Büyüyen Kalp Efektı Program Blokları

Gözle: Sürekli Tekrar Eden Büyüyen Kalp Efektı

Rehber öğretmen, büyüyen kalp efekti programındaki kodun bir defa çalıştığında bittiğini söyler. Efekti birden fazla tekrar etmek istediğimizde veya bu efeklin sürekli oynatılması gerektiğinde hazırlanan kodların kopyalanıp art arda eklenmesi yerine “Kontrol Blokları” paketinden “sonsuz döngü” ve “döngüyü tekrarla” bloklarının kullanılabileceğini aktarır. “döngüyü tekrarla” bloğunda istenilen sayı kadar içerdeki blokların çalıştırılacağı ve varsa “döngüyü tekrarla” bloğundan sonraki bloğun çalışacağını, başka blok yoksa bu bölümün biteceği vurgulanır. “sonsuz döngü” bloğunun ise içerisindeki blokları program açık olduğu sürece tekrarlayacağı belirtilir. Büyüyen kalp efekti programındaki kodların “sonsuz döngü” ve “döngüyü tekrarla” blokları ile çalıştırılıp, öğrencilerin de kendi kodlarını bu iki blokla çalıştırmaları istenir.



Resim 3.3 Sonsuz Döngü ve Döngüyü Tekrarla Program Kodları

Uygula: Başla ve Geri Sayım

Rehber öğretmen, öğrencilerde Hub ekranında “BASLA” yazısı aktıktan sonra, sırasıyla 3, 2, 1 ve 0 yazmasını ve sonra ekranın bütün ışıklarının 0,5 saniye ara ile 2 defa yanıp sönmesi için gerekli blokları oluşturmalarını ister. Ayrıca, programda ‘ışıklarla yaz’ bloğu içine yazılan metnin ekranda akması tamamlanmadan sonraki program bloklarının çalışmayacağını öğrencilere hatırlatır.



Resim 3.4 Başla ve Geri Sayım Program Kodları

SES

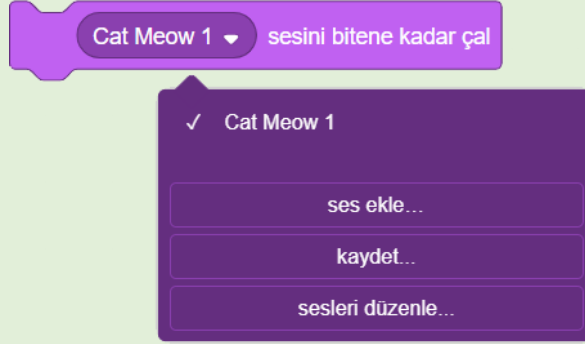
Rehber Öğretmen İçeriği – Ses Bloğu

Ses Blokları Paketi

Kullanılan cihazı ve/veya Hub’ı kullanarak sesler çalmaya olanak sağlar.

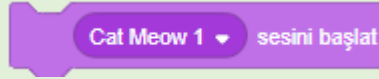
Sesi Bitene Kadar Çal

Seçilen sesi çalar ve bitene kadar sonraki bloğa geçmez. Blok ile açılan pencerede 200'den fazla ses arasından seçim yapılabilir, yeni kayıt yapılabilir veya seçilen sesler düzenlenebilir.



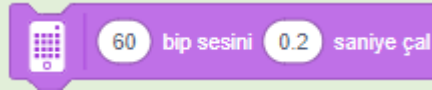
Sesi Başlat

Seçilen sesi çalar ve hemen sonraki bloğa geçer. Blok ile açılan pencerede 200'den fazla ses arasından seçim yapılabilir, yeni kayıt yapılabilir ve seçilen sesler düzenlenebilir.



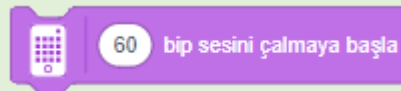
Bip Sesini Belirli bir Saniye Çal

Seçilen bip tonunu belirlenen süre için çalar.



Bip Sesini Çalmaya Başla

Seçilen bip tonunu program bitene veya durdurulana kadar çalar.



Tüm Sesleri Durdur

Çalınmakta olan (bip ve ses dosyaları gibi) tüm sesleri durdurur.

tüm sesleri durdur

Ses Seviyesini Değiştir

Çalınmakta olan sesin seviyesini çalındığı ses yüksekliğine göre azaltmak veya artırmak için kullanılır. Varsayılan ses seviyesi %100'dür.

ses düzeyini -10 değiştir

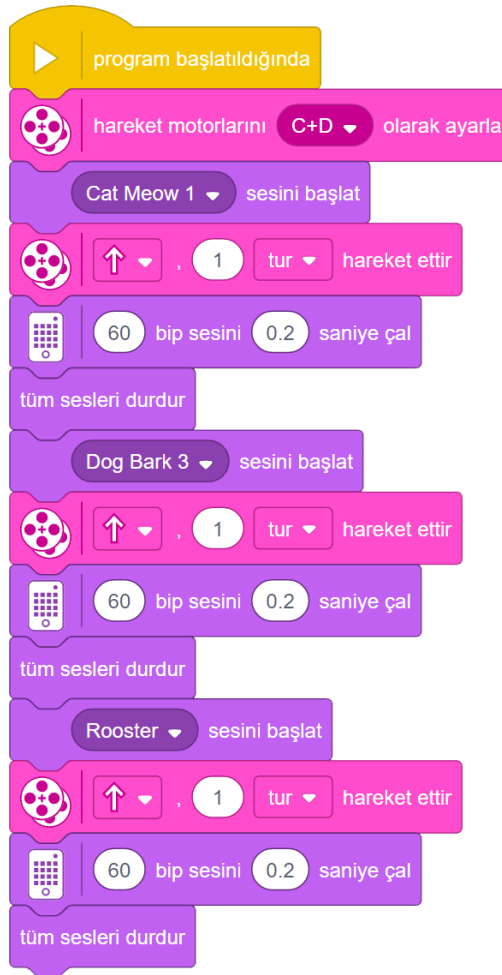
Gözele: Ses Blokları

Rehber öğretmen, 'Ses Blokları'nın öğrencilerin cihazlarından (bilgisayar) ve Hub'larından sesler çalmak için kullanıldığını belirtir. 'Ses Blokları'ndaki 'sesi bitene kadar çal', 'sesi başlat', 'bip sesini çalmaya başla', 'tüm sesleri durdur', 'ses perdesi efektini değiştir', 'ses perdesi efektini ayarla', 'ses efektini temizle', 'ses seviyesini değiştir', 'ses seviyesini ayarla' ve 'ses seviyesi' blokları gösterilir. Projelerine sadece eğlence amacıyla değil, aynı zamanda hata ayıklama amacıyla da ses blokları eklenebileceğinden bahseder. Örneğin, bir kod parçasının ne zaman tamamlandığını belirtmek için kullanılabilirler.

Not

Hub'da yalnızca 'bip' sesi blokları oynatılır; diğer sesler cihazınızda (dizüstü bilgisayar, tablet vb.) oynatılır. Bu nedenle etkinlik esnasında cihazın sesinin açık olduğundan emin olunuz. Aksi halde programda yer alan seslerden yalnızca 'bip' sesleri duyulabilir.

Robotun 3 tur düz ilerlemesi, ilerlerken önce cihazdan kedi, köpek ve horoz sesi çıkarması ve her turdan sonra da Hub'dan bip sesi çıkarması için gerekli kod bloğunun oluşturulması istenir. Resim 3.5'teki gibi bir program akışı oluşturulabilir.



Resim 3.5 İlerlerken Ses Çıkaran Robot Program Kodları

Uygula: Kare Çize Robot Aktivitesi

Öğrencilerden, robotun 20 cm'lik bir kare üzerinde ilerlemesi için bir program oluşturmaları istenir. Programı oluştururken 'döngüyü tekrarla' bloğunu kullanmaları, robot düz ilerlerken Hub'ın ilerleme yönünde ok işareti göstermesi, robot dönerken bip sesi çıkarması ve döndüğü yönde Hub ekranında ok işareti göstermesi sağlanmalıdır. Rehber öğretmen, uygulamanın sonunda başladığı yere en yakın şekilde karesini bitiren robotun oyunu kazanacağı bir aktivite yapacaklarını belirtir.

Bütün öğrenciler programlarını tamamladıktan sonra aktiviteye başlanır. Aktiviteye başlamadan önce robotun yerdeki teker izleri başlangıç çizgisi olarak çizilir. Daha sonra, sırayla bütün grupların robotlarının kareyi tamamlama aktivitesini yapmaları istenir. Yarışmayı kazanan robotun belirlenmesi için her grubun kare işlemini tamamladıktan sonra başlangıç çizgisine olan mesafeleri ölçülür ve not alınır (Örnek çözümde verilen 'belirli süreyle hareket ettir' bloğundaki mesafe değerinin, robotun hareket ettiği zemindeki materyal, pil durumu ve çözümde kullanılan bloklar gibi çeşitli nedenlerden dolayı farklılık gösterebileceği öğrencilere hatırlatılır).



Resim 3.6 Kare Çizen Robot Program Kodları

MÜZİK

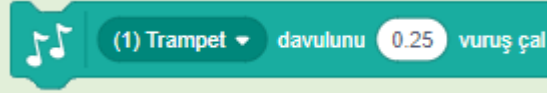
Rehber Öğretmen İçeriği – Müzik Blokları

Müzik Blokları Paleti

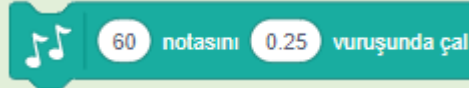
Müzikal sesler oluşturmak için kullanılırlar. Eklentiler bölümünden 'müzik' eklentisinin aktif edilerek blok paletine eklenir.

Davul Çal

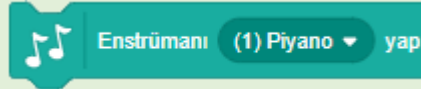
Vuruş şeklinde ölçülen bir süreyle belirtilen enstrümanı çalmak için kullanılır.

**Nota Çal**

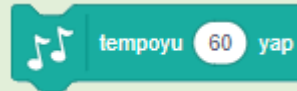
Vuruş şeklinde ölçülen bir süreyle belirtilen notayı çalmak için kullanılır.

**Enstrümanı Belirle**

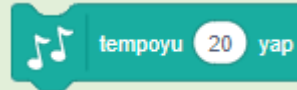
Nota Çal bloğunda kullanılacak enstrümanı belirlemek için kullanılır.

**Tempoyu Belirle**

Vuruş temposunu (hızını) belirlemek için kullanılır. 60 değeri bir vuruşun her saniye çalınacağını belirtir. Büyük değler notaların daha hızlı çalması anlamına gelir.

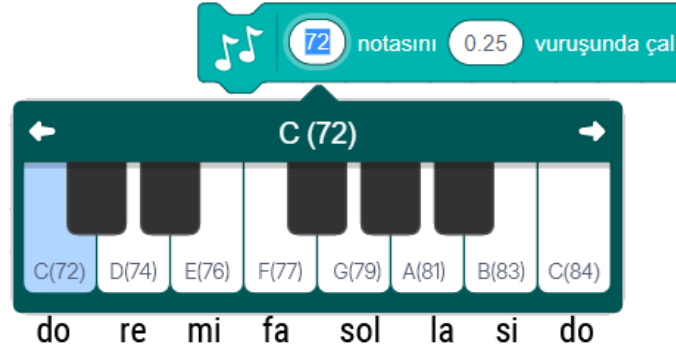
**Tempoyu Değıştir**

Vuruş temposunu belirlemek için kullanılır. 60 değeri bir vuruşun her saniye çalınacağını belirtir.

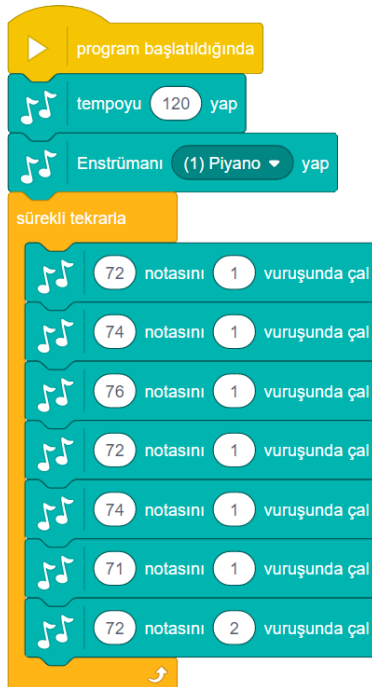
**Göster: Müzik Blokları**

Rehber öğretmen, "Müzik" blok paketine eklenti penceresi kullanılarak ulaşıldığını belirtir. Rehber öğretmen öğrencilere bu blokların müzikal sesler, ritimler, ezgiler ve örüntüler oluşturmak için kullanılabileceğinden bahseder. Bu oluşturulan müzikal seslerin tamamı öğrencinin cihazları (bilgisayar) tarafından oynatılır. "nota çal" bloğunda 72 rakamına tıklanınca açılan pencerede pişano klavyesinde 72 rakamına denk gelen Do notasının bulunduğ u oktavın diğ er notaları da ekranda görünür (C72 ile başlayan oktavın notalarının sayısal değ erleri Resim 3.7'de gösterilmektedir). Görselde sadece beyaz tuşlar sayılar ile gösterilmiş olsa da aralardaki siyah tuşlar da sayılar ile ifade edilmektedir. Çalma listenize daha kalın notaları sol tarafı gösteren oku,

daha ince notaları eklemek için sağ taraftaki oku tıklamanız gerekir. Öğrencilerin “enstrümanı belirle” bloğunu kullanarak piyano ile ceviz adam çocuk şarkısının notalarını kodlamaları istenir. Resim 3.8’de program girişi ve Ceviz Adam şarkısının nota ve vuruş sayısal değerleri (“nota çal” bloğu değerleri) verilmektedir. Her bir nota için “nota çal” bloğu komutu oluşturulmalıdır.



Resim 3.7 “Nota Çal” Bloğunda Nota ve Sayısal Değerler Gösterimi



Tempo 120			
Sıra	Nota	Sayı	Vuruş
1	do	72	1
2	re	74	1
3	mi	76	1
4	do	72	1
5	re	74	1
6	si	71	1
7	do	72	2

Resim 3.8 Ceviz Adam Şarkısı Kodları ve Nota-Vuruş Değerleri Listesi

Uygula: Beethoven - 9. Senfoni (Neşeye Övgü)

Öğrencilerin Beethoven’ın 9. Senfonisini tempo bilgileri de ekleyerek kodlamaları istenir. Program bloklarına senfoni ile çalması için bir ritim de eklenmiştir (Resim 3.9). Senfoninin “nota çal” bloğu değerleri Resim 3.9’da verilmiştir.

Not

<https://www.youtube.com/watch?v=TKegTvdEnpM> adresinden Çakabey ve Olten Filarmoni Okulu Çoksesli Çocuk Korosu’nun seslendirdiği şarkının videosu öğrencilere izletilebilir.

Tempo 120			
Sıra	Nota	Kod	Vuruş
1	mi	76	1
2	mi	76	1
3	fa	77	1
4	sol	79	1
5	sol	79	1
6	fa	77	1
7	mi	76	1
8	re	74	1
9	do	72	1
10	do	72	1
11	re	74	1
12	mi	76	1
13	mi	74	1
14	re	74	1
15	re	74	2
16	mi	76	1
17	mi	76	1
18	fa	77	1
19	sol	79	1
20	sol	79	1
21	fa	77	1
22	mi	76	1
23	re	74	1
24	do	72	1
25	do	72	1
26	re	74	1
27	mi	76	1
28	re	74	1
29	do	72	1
30	do	72	2

Resim 3.9 Beethoven'in 9. Senfonisi Müzik Program Kodları ve Nota-Vuruş Değerleri Listesi

TASARLA

Tasarla etkinliğinde rehber öğretmen, öğrencilerden robot tasarımlarında herhangi bir değişikliğe gitmeden, robotlarını dans edecek şekilde programlamaları ister. Robot dans ederken hareketler, müzik ve Hub ekranındaki görseller senkronize bir şekilde kullanılmalıdır. Gruplardan tartışma yöntemiyle dans eden robotun kodunun nasıl yazılacağı üzerinde fikir üretmeleri istenir. Gruplar çözümü kendileri bulmalıdır, çözümün tam olarak verilmesi öğrencilerin yaratıcılıklarını olumsuz yönde etkileyebileceğinden tavsiye edilmez. Gerekirse rehber öğretmenden yardım alınabilir. Dans eden robotu tasarlama süreci aşağıda örnek olarak verilen tanımlama ve fikir üretme başlıklarına benzemelidir.

Tanımlama: Öncelikle öğrencilerin, robotun dans hareketleri için neler gerektiğini belirlemeleri gereklidir. Bunun için; robotun dans stiline, hareketlerine, çalacak notaya ve Hub ekranında yer alacak görsele karar vermeleri gerekir. Öğrenciler bu adımları maddeler hâlinde yazmalıdır.

Örneğin:

- Belirlenen bir parçanın notalarını çalacak,
- Nota ve hareketlerin birlikte hareket etmesini sağlamak için ritim belirlenecek,
- Çalınan notalara paralel olarak dans edecek (sağ, sol, ileri, sağ, sol, geri hareket edecek),
- Hub ekranında robotun hareketine paralel olarak sağa sola dönüş okları; ileri gidince ileri, geri gidince geri oku gösterecek,
- Dans sonlandırma hareketi, notası ve görüntüsü yapılacak.

Fikir üretme: Bu aşamada öğrenciler, tanımlamada belirlenen işlemlerin nasıl yapılabileceği ile ilgili fikir yürütmelidir. Öğrenciler aşağıdaki maddelere benzer fikirler üretebilir:

Örneğin:

- Öncelikle her bir harekette Hub ekranı görüntüsü veya bilgisayarın çıkaracağı sesler/müzikler (nota ve vuruş değerleri) belirlenir. Öğrencilerin internetten müzik parçaları için nota ve vuruş değerleri indirmelerine veya kendi müziklerini programlamalarına izin verilmelidir.
- Sonra robotun hangi hareketleri yapacağı planlanmalıdır. Planlanan her bir hareketin tanımı yapılmalıdır (Örneğin robot önce 0,5 saniye sağa doğru döner ve durur. Sonra iki katı hızlı bir şekilde yine 0,5 saniye sağa doğru döner ve durur.). Bu aşamada öğrenciler hareketlerin programlanmasını hızlıca deneyebilirler.
- Hareket ve müzikle birlikte Hub ekranında hangi görsellerin gösterileceği de belirlenmelidir.
- Bu sürecin sonunda öğrenciler müziği, dans hareketlerini, Hub görüntülerini ve durum ışığı bilgilerini birlikte anlatabilmelidir.

Öğrenciler bu süreçte farklı çözümler gerçekleştirebilirler. Tüm grupların aynı veya benzer sonuçlar üretmesi zorunlu değildir. Ayrıca, öğrencilerin farklı fikirleri test etmeleri rehber öğretmenler tarafından desteklenmelidir.

Not

Rehber öğretmenler, Tasarla Üret bölümlerinde verilen kod örneklerini ve ekran görüntülerini öğrencilerle paylaşmamalıdır. Tasarla Üret bölümündeki paylaşımlar, rehber öğretmenlerin etkinliklere hazırlanmasında kolaylık sağlamak amacıyla sunulmuştur. Öğrenciler, Tasarla Üret bölümlerinde sunulan problemlere kendileri çözüm üretmelidir. Rehber öğretmenler, öğrencilere problemleri çözebilmeleri için yeterince zaman tanımalı ve doğrudan çözümü vermeden rehberlik sağlamalıdır. Tasarla Üret etkinliklerinde tam çözüm hiçbir zaman öğrencilere verilmemelidir.

ÜRET

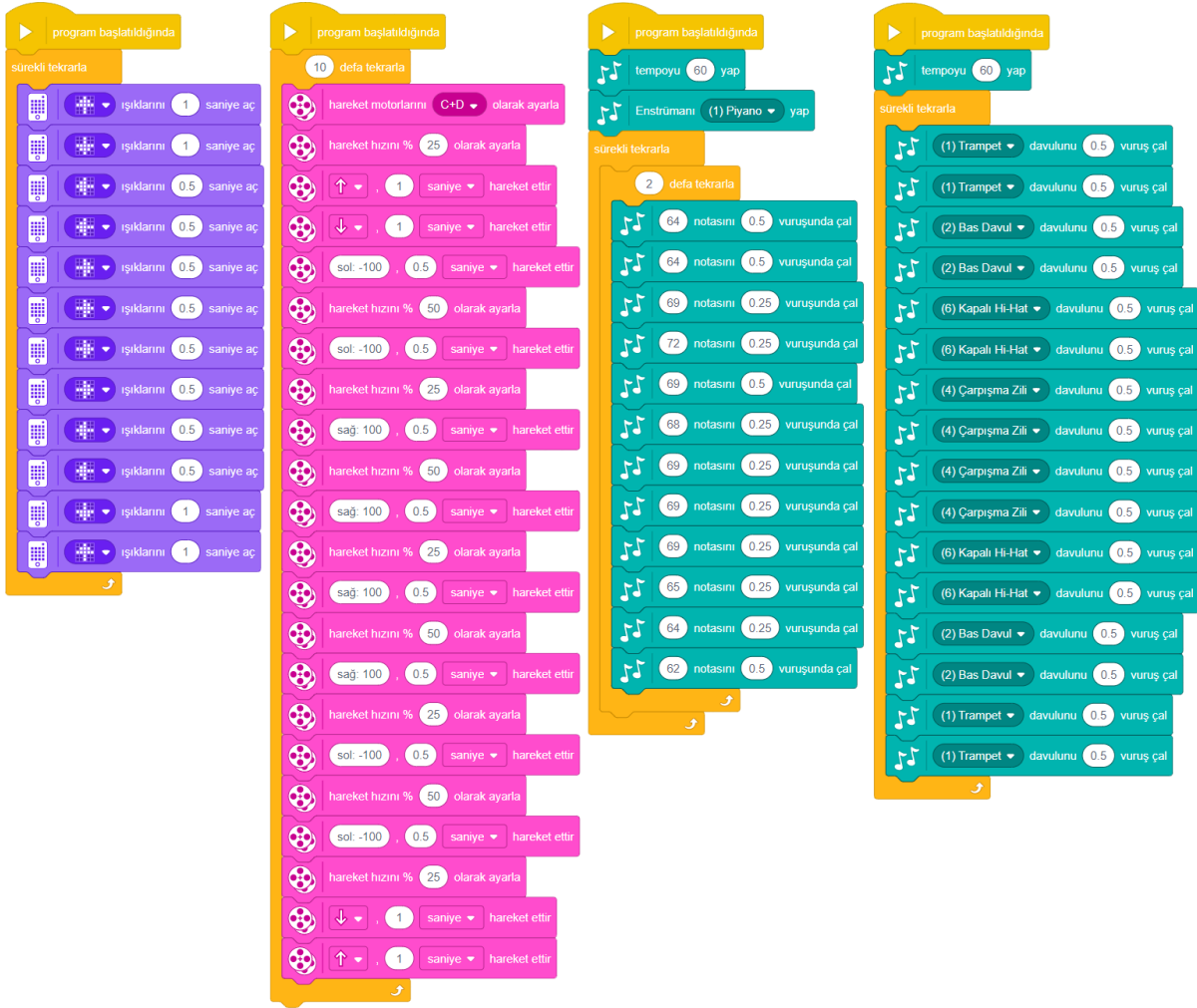
Öğrenciler, tasarlama aşamasında tanımlama ve fikir üretme adımlarını tamamladıktan sonra bilgisayar ve robotlarla çalışarak gerekli programı oluştururlar. Bu bölümde öğrenciler aktiftir. Rehber öğretmen, ihtiyaç duyulduğunda öğrencileri yönlendirir veya takıldıkları noktalarda yardımcı olur; yani rehber öğretmen yalnızca ihtiyaç anında destek sağlamalıdır.

Üret aşamasında, öğrencilerden bir önceki adımda tasarladıkları dans eden robot planını kullanarak probleme daha yapılandırılmış bir çözüm önerisi geliştirmeleri istenir. Öğrenciler, bilgisayar ve robot başında çalışarak gerekli yazılım çözümlerini geliştirirler. Burada öğrencilerin en çok dikkat etmesi gereken konu, ses, hareket ve Hub ekranı görüntüsü senkronizasyonudur.

Püf noktalar (Kendilerinin keşfetmesine veya sorarak öğrenmelerine imkân vermek için bunları başlangıçta öğrencilere söylemek uygun değildir.):

- Hareketin süresi, Hub ekranı ve müzik bölümlerinin ekranda bekleme süresi aynı olmalıdır.
- “*nota çal*” (Müzik Blokları), “*ışık matrisini belirli bir saniye aç*” (Işık Blokları) ve “*belirli süreyle hareket ettir*” (Hareket Blokları) bloklarındaki süreler senkronizasyon için paralel kullanılabilir.
- Hareket, müzik ve ekran görüntüsü için ayrı “*program başlatıldığında*” blokları kullanılabilir.
- Önce çalınacak müziğin programlanması, sonra hareket veya Hub ekranının programlanması senkronizasyonun sağlanmasını kolaylaştırabilir.

Örnek program kodları ve müzik olarak kullanılan şarkının nota değerleri Resim 3.10 ve Resim 3.11’de verilmiştir. Örnek programda müzik, hareket ve ışık senkronizasyonunu sağlamak için bloklarda süreler kullanılmıştır. Ayrıca örnek programda döngü sayısı bir olarak belirtilmiştir. Öğrenciler de kendi robotlarında kullandıkları müzik, hareket, ışık ve tempo senkronizasyonunu sağladıktan sonra “*döngüyü tekrarla*” bloğundaki değeri değiştirerek dansın süresini uzatabilirler.



Resim 3.10 Dans Eden Robot Program Kodları

Tempo 60			
Sıra	Nota	Kod	Vuruş
1	mi	64	0.5
2	mi	64	0.5
3	la	69	0.25
4	do	72	0.25
5	la	69	0.5
6	sol#	68	0.25
7	la	69	0.25
8	la	69	0.25
9	la	69	0.25
10	fa	65	0.25
11	mi	64	0.25
12	re	62	0.5

Resim 3.11 Dans Eden Robot Nota-Vuruş Değerleri Listesi

DEĞERLENDİR

Günün sonunda rehber öğretmen öğrencilere aşağıdaki soruları yönelterek öğrenilenlerin birlikte özetlenmesini sağlamalıdır.

- Size bugün Tasarla ve Üret adımımda verilen problemi nasıl tanımlarsınız? (problemi kendi cümleleri ile ifade etme).
- En çok hangi görevde zorlandınız? Bu zorlukların üstesinden nasıl geldiniz? (Problemin çözümü için hangi stratejileri kullandınız ve neden bu stratejileri seçtiniz?) Yeteri kadar tartışma ortamı oluşmazsa rehber öğretmen aşağıdaki soruları kullanarak tartışma ortamı yaratmaya çalışır.
 - Dans adımlarına nasıl karar verdiniz, bunları programlarken ne tür sıkıntılar yaşadınız ve bunların üstesinden gelmek için neler yaptınız?
 - Müzik, Hub ekranı, hareket ve ritme nasıl karar verdiniz; bunları programlarken ne tür sıkıntılar yaşadınız ve bunların üstesinden gelmek için neler yaptınız?
- Kullandığınız yöntemler bu sıkıntıları gidermede başarılı oldu mu?

İLAVE ETKİNLİK

Karşılıklı Dans Eden Robotlar

Dans eden robot etkinliğini tamamlayan gruplardan, robotlarını karşılıklı (tercihen 2 robot) dans ediyormuş gibi kodlamaları istenir. Karşılıklı dans eden robotların, dans etkinliğindeki programladıkları dansı partner robota göre düzenlemeleri gerekmektedir. Her iki robotun da önceki dansları, partnerleri için yeniden düzenlenmelidir.

4. Hafta - Kuvvet Sensörü

Haftanın Amacı:

Bu haftaya başlarken duyu organlarımız ile analogi yapılarak sensör kavramı öğrencilere aktarılacaktır. Lego SPIKE Prime robot setinde yer alan kuvvet sensörü öğrencilere tanıtılarak, kuvvet sensörünün; basılma, sert basılma ve bırakılma parametreleri ve ölçülen kuvvet miktarının farklı programlarda kullanılması sağlanacaktır. Kuvvet sensörü ile ilgili kod blokları tanıtılarak, ilgili problemlerin çözümünde kod bloklarının kullanılması amaçlanmıştır.

Haftanın Kazanımları:

- Öğrenciler kuvvet kavramını ifade edebilir.
- Öğrenciler kuvvet sensörünün basılma, sert basılma ve bırakılma durumlarını, program akışında kullanabilir.
- Öğrenciler kuvvet sensörü kullanarak, kuvvet ölçümü yapabilir.
- Öğrenciler kuvvet sensörü ile ölçülen değerleri, robotun motor gücü girdisi olarak kullanabilir.
- Öğrenciler sensörlerden aktarılan verilerin grafiğini çizdirebilir.
- Öğrenciler farklı hızlarda ilerleyen robotun, çarpışma anında oluşturduğu kuvveti inceleyebilir.
- Öğrenciler birden fazla değişkenin grafiklerini yorumlayabilir.

Kullanılacak Malzemeler:

Robot seti ve bilgisayar.

Ekler:

Konu anlatımında ekran görüntüleri verilen örnek programlar, ayrıca dijital formatta rehber öğretmenlere sunulmuştur. Örnek program dosyalarının numaralandırılmasında, konu anlatımındaki resim numaraları temel alınmıştır. Bu hafta aşağıda sıralanmış programlar ekte rehber öğretmenlere sunulmuştur.

Program_4.3_Kuvvet_Yuzde_Hareket.llsp3
Program_4.4_Kuvvet_Newton_Hareket.llsp3
Program_4.5_Kuvvet_Newtonx10_Hareket.llsp3
Program_4.6_Oyun_Newtonx10_cm.llsp3
Program_4.9_Kuvvet_Engele_Carpinca_Duran.llsp3
Program_4.10_Carpisma_Testi.llsp3
Program_4.11_Tasarla_Uret_Kuvvet_Motor_Grafik.llsp3
Program_4.13_Oyun_NewtonXSure_cm.llsp3

Kuvvet Sensörü

Rehber Öğretmen İçeriği – Kuvvet Sensörü

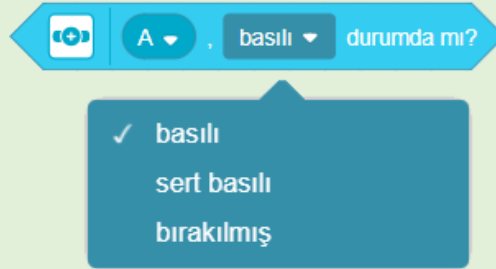
Kuvvet sensörü, EV3 robot setinde bulunan dokunma sensörlerine birçok yönden benzer. Kuvvet sensörü, bir dokunma sensörü gibi basıldığında ve bırakıldığında durumlarını algılayabilir. Ancak kuvvet sensörü, dokunma sensöründen farklı olarak 10 Newton'a kadar kuvveti ölçebilir. Ayrıca sert basıldığında (5N üzeri kuvvetle basıldığında) durumunu da algılayabilir.

Newton, kuvvet birimi olup simgesi N'dir. Terim, fizik bilimine yaptığı katkılar nedeni ile İngiliz bilim adamı Isaac Newton'un adı ile anılır. Kütlesi 1 kg olan bir cismin hızını, 1 saniye içerisinde 1 m/s arttırmak için o cisme uygulanması gereken kuvvet, newton cinsinden $1N = 1kg.m/s^2$ olarak tanımlanır.

Bu hafta etkinliklerde kullanılacak yeni bloklar ve her bloğun açıklaması aşağıda verilmiştir.

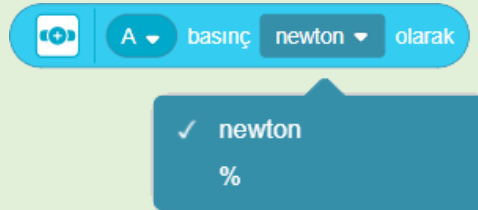
Sensörler Blok Paleti

Basılı mı?



Bu blok, bir "Boole Bloğu" olduğundan doğru (true) veya yanlış (false) değerlerini alır. Kuvvet sensörü basılmış (0 N'dan daha büyük bir kuvvet uygulanmış) sert basılmış (5 N'dan daha büyük bir kuvvet uygulanmış) veya serbest bırakılmış (hiçbir kuvvet uygulanmamış) olduğunda doğru sonucu verir.

Basınç



Bu blok kuvvet sensörüne uygulanan basıncın değerini Newton veya yüzde olarak alır. Kuvvet sensörünün teknik özellikler kitapçığında 2,5 - 10 Newton'u tespit edebildiği belirtilmiştir. Newton olarak ölçülen basınç değeri maksimum 10N, yüzde olarak 100'e karşılık gelir.

Değişkenler Blok Paleti

Değişkenler, programlarda sayıları, metinleri veya diğer bilgilerin tutulmasına olanak tanır. Değişkenler paletinde, yeni değişkenler oluşturabilir, mevcut değişkenler programlarda

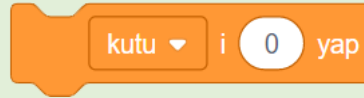
kullanılabilir ve değiştirilebilir. Değişkenler paletinde 'Bir Değişken Oluştur' düğmesi tıklandığında açılan pencereden değişkenin ismi girilerek yeni bir değişken oluşturulur. Oluşturulan yeni değişken ismi ile üç farklı program bloğu listeye eklenir.

Değişken

kutu

Bir değişken oluşturulduğunda, değişkenin ismi ile bir blok oluşur. Örneğin 'kutu' isimli bir değişken oluşturulduğunda, yukarıdaki gibi bir blok oluşur ve bu blok değişkenin aldığı değeri yansıtır. Bu blok, diğer blokların içerisine yerleştirilerek kullanılabilir. Örneğin robotun değişken değeri kadar tur, cm, saniye ileri gitmesi sağlanabilir.

Değişkeni Şuna Ayarla



Bu blok, seçilen değişkene istenilen değerin atanmasını sağlar. Örneğin sensörden ölçülen değerin saklanması için kullanılabilir.

Olaylar Blok Paleti

Basınç Şu Olduğunda



Bu "Şapka Bloğu" kendisine bağlı olan tüm blokların kuvvet sensörüne basıldığında, sert basıldığında, sensör bırakıldığında veya uygulanan basınçta herhangi bir değişiklik algılandığında çalışmasını sağlar.

Bu blok sadece belirtilen olay durumunda tetiklenir. Kuvvet sensöründe basıncın değişmemesi halinde blok yeniden tetiklenmez.

Sensörler Blok Paleti

Kronometre

kronometre

Bu blok program çalışmaya başladığı andan itibaren geçen sürenin saniye cinsinden değerini alır. Program her başlatıldığında kronometre değeri sıfırlanır.

Kronometreyi Sıfırla

kronometreyi sıfırla

Bu blok, sayesinde programın akışı içerisinde istenilen durumda kronometre değeri sıfırlanır.

Blok Eklentileri – Çizgi Grafiği

Çizgi grafiği blokları SPIKE Prime Uygulaması arayüzünde varsayılan olarak açık değildir. Çizgi grafiği bloklarına ulaşabilmek için, blok paletinde sol alt köşede yer alan "blok eklentilerini göster" düğmesi tıklanarak, çizgi grafiği kutucuğunun seçilmesi gerekmektedir.

Çizgi Grafiğini Temizle

çizgi grafiğini temizle

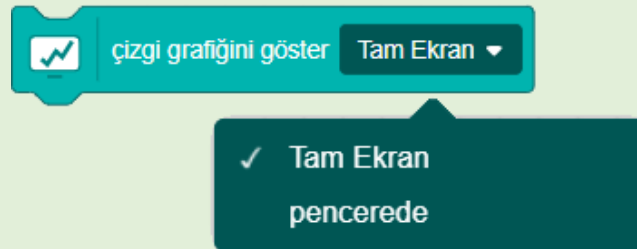
Bu blok, çizgi grafiği ekranını temizler. Önceki ölçümlere ait tüm verileri siler.

Değeri Çizgiye İşaretle



Bu blok, girilen değere veya değişkene (sensör veya kronometre verisi gibi zamanla değişen değerlere) ait verilerin zamana bağlı grafiğini istenilen renkte çizgi grafiği olarak çizilmesini sağlar.

Çizgi Grafiğini Tam Ekran Göster



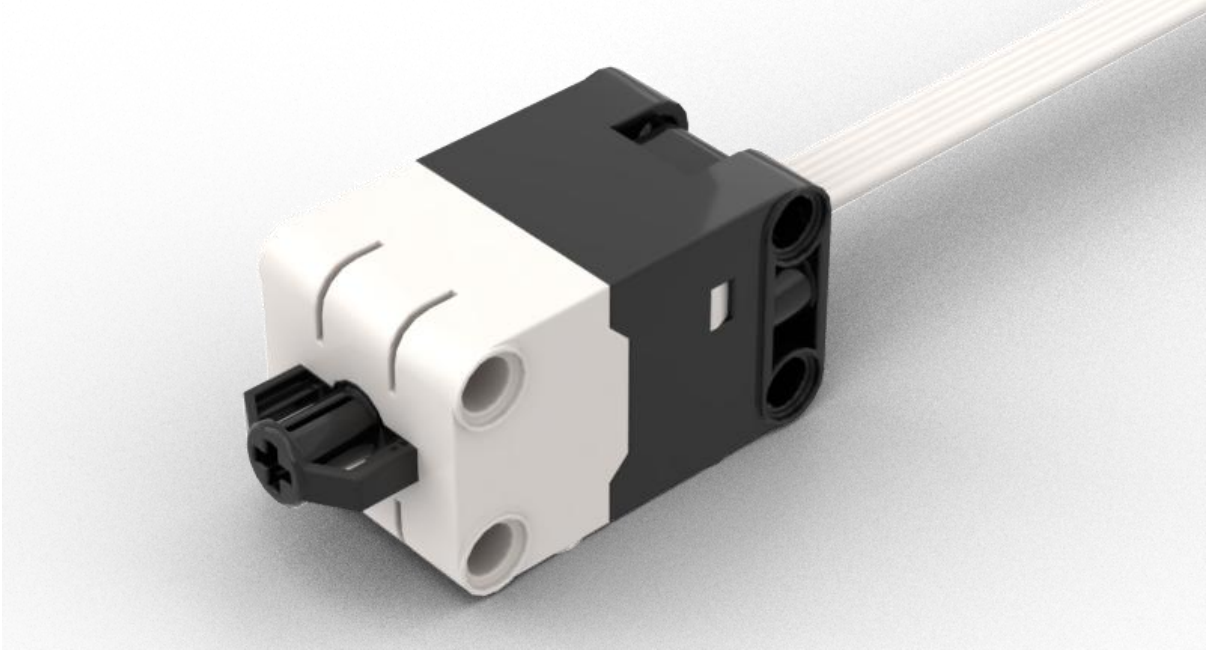
Bu blok, çizgi grafiğinin tam ekran ya da yüzen bir pencerede açılmasını sağlar.

Not

*Etkinliklerde hazırlanması planlanan programların ekran görüntüleri, ilgili konu içerisinde sunulmuştur. Ayrıca program dosyaları da rehber öğretmenlerle paylaşılmıştır. Bu paylaşımlar rehber öğretmenlerin etkinliklere hazırlanmasında kolaylık sağlamak içindir. Gözle etkinliklerinde rehber öğretmen ilgili programları öğrencilere açıklayarak adım adım hazırlamalıdır. Tasarla ve üret etkinliklerinde ise rehber öğretmen programları ve ekran görüntülerini öğrencilerle kesinlikle **paylaşmamalıdır**.*

GÖZLE VE UYGULA**Gözle: Kuvvet Sensörü**

Rehber öğretmen, duyu organlarımızla bir analogi kurarak robotların çevrelerini sensörleri aracılığıyla algıladığını açıklar. İlerleyen haftalarda, öğrencilerin robotun farklı sensörleriyle çalışacaklarını ve bu sensörlerin robotu programlarken nasıl kullanılacağını öğreneceklerini belirtir.



Resim 4.1 Kuvvet Sensörü

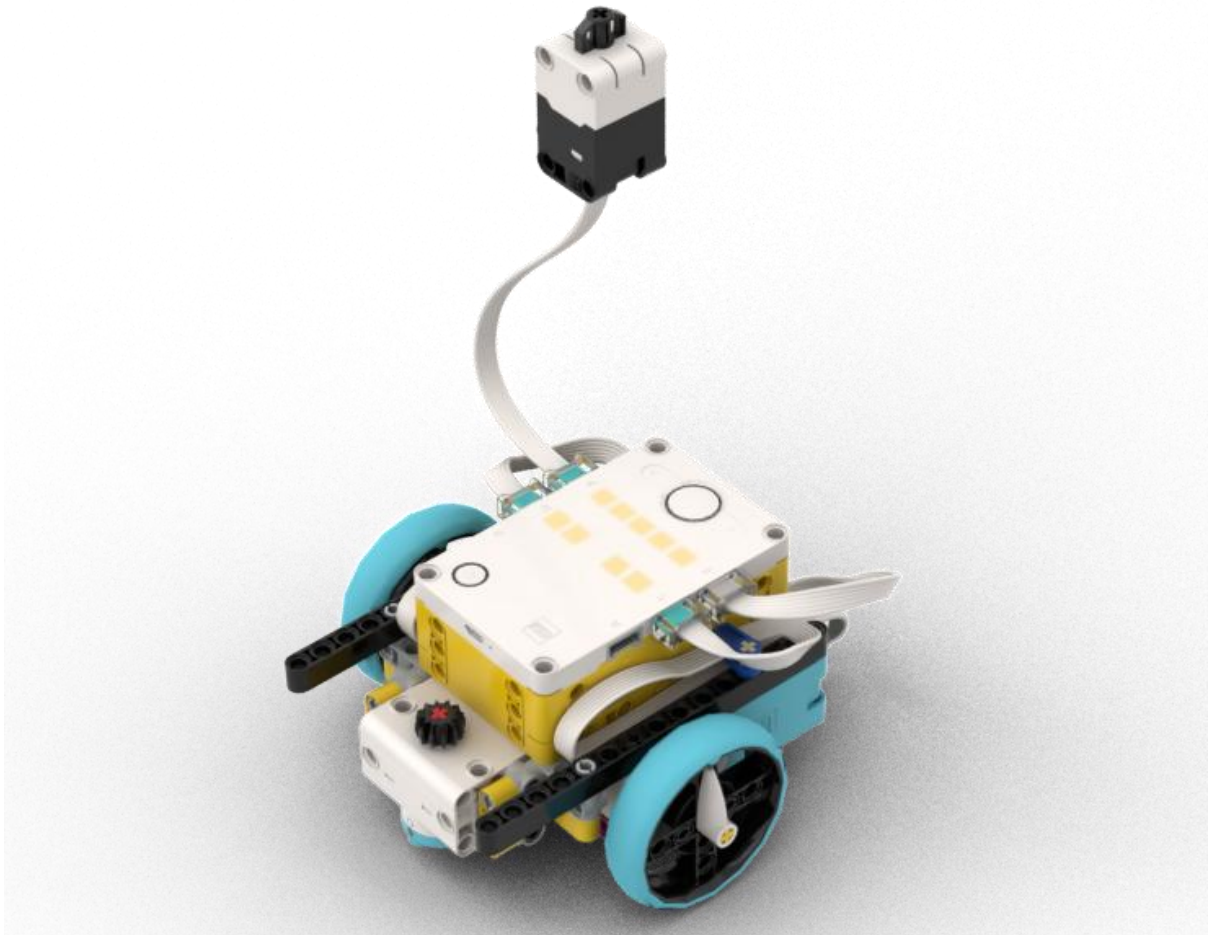
Bu hafta kuvvet sensörü ile başlayacaklarını belirterek, robot setindeki kuvvet sensörünü gösterir ve özelliklerini açıklar. Ardından, kuvvet sensörünü Resim 4.2’de görüldüğü gibi Sürüş Modeli tasarımında Hub’ın F portuna bağlamalarını ister. Ayrıca, E portunda büyük motorun bağlı olduğundan emin olmalarını ister.

Robot açıkken ve ekranda kalp sembolü görünürken, açma/kapama düğmesine kısa bir süre basıldığında kuvvet sensörünün bağlı olduğu F portu ile büyük motorun bağlı olduğu E portu arasında ışıklarla bir bağlantı oluşacaktır. Bu durumda, kuvvet sensörüne dokunulduğunda büyük

motor dönmeye başlayacaktır. Uygulanan kuvvetin büyüklüğüne bağlı olarak, kuvvet sensörüne ne kadar fazla kuvvet uygulanırsa, büyük motorun da o kadar hızlı döndüğü gözlemlenecektir.

Uygula: Kuvvet Sensörü ile Motorların Çalıştırılması

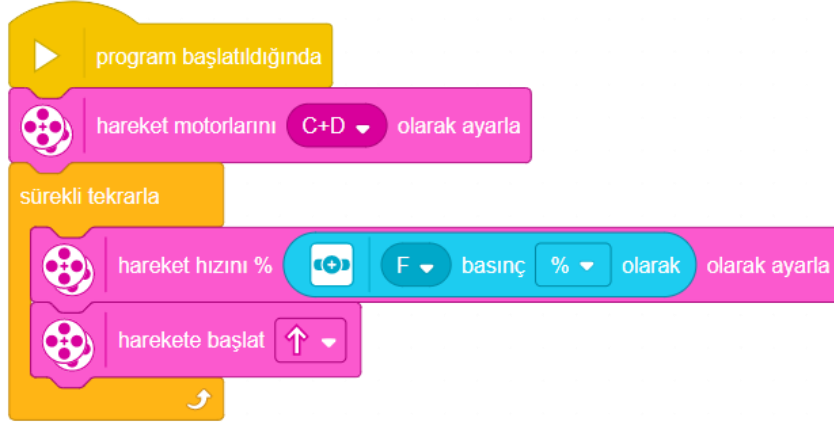
Rehber öğretmen, öğrencilerden anlatılanları uygulayarak büyük motoru kuvvet sensörü ile çalıştırmalarını ister. Benzer şekilde, diğer motorların bağlı olduğu portların karşısındaki portlara herhangi bir sensör takıldığında, sensörden okunan verinin motorun dönme hızını ayarlayacağını açıklar. Öğrenciler, kuvvet sensörü ile denemeler yaptıktan sonra, bir kez daha açma/kapama düğmesine kısa bir süre basarak program moduna (kalp simgesinin olduğu moda) geçerler.



Resim 4.2 Kuvvet Sensörü ile Motor Hareketi

Göze: Kuvvet Sensörü ile İlk Program

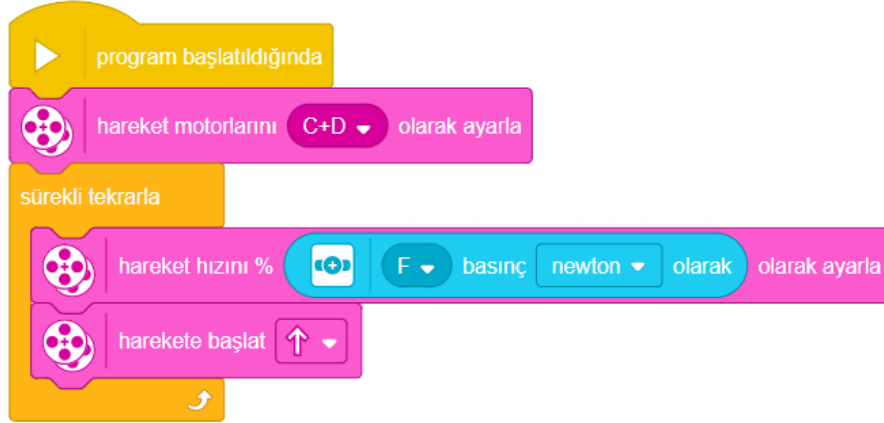
Rehber öğretmen, öğrencilere bir önceki uygulamada yalnızca bir motorun çalıştırılabildiğini belirtir. İki motorun birden çalıştırılarak, örneğin robotun kuvvet sensörüne uygulanan basınç miktarına göre ileriye doğru hareket hızının ayarlanabildiği programı öğrencilere anlatır. Ayrıca, rehber öğretmen, programlarında basınç değerini yüzde olarak kullanmalarını ister. Örnek program, Resim 4.3'te sunulmuştur.



Resim 4.3 Kuvvet Sensörü Yüzde Hareket Örnek Program

Uygula: Kuvvet Sensörü ile Yüzde ve Newton

Rehber öğretmen, öğrencilerden anlattığı programı uygulamalarını ister. Tüm öğrenciler, kuvvet sensöründen algılanan yüzde değeri ile robotun hızının ayarlandığı programı tamamladıktan sonra, aynı programı bu sefer yüzde yerine Newton kullanarak çalıştırmaları istenir. Programda yapılan değişiklik, Resim 4.4'te görülmektedir.



Resim 4.4 Kuvvet Sensörü Newton Hareket Örnek Program

Rehber öğretmen öğrencilerle her iki programda elde edilen sonuçları tartışır. Tartışmayı aşağıdaki sorularla yönlendirebilir:

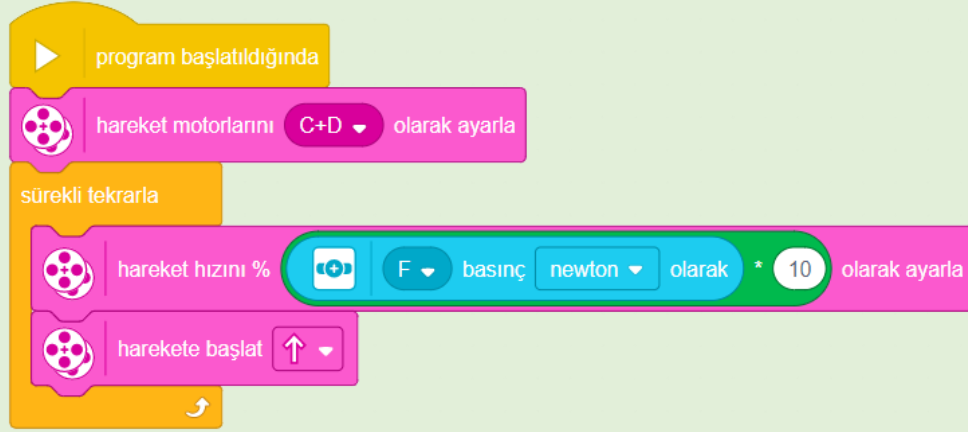
- Aynı kuvvet sensörü kullanıldığı halde, basınç Newton olarak ayarlandığında robot neden daha yavaş hareket etti?
- Basıncın yüzde olarak ayarlandığında robotun maksimum hızı, Newton olarak ayarlandığındaki robotun hızının kaç katı olabilir?
- Basınç Newton olarak ayarlı iken, robotun yüzde olarak ayarlandığı hızda gidebilmesi için programda ne gibi değişiklikler yapılmalıdır.

Rehber öğretmen, öğrencilerin programlarında gerekli değişiklikleri yaparak uygulamalarını ister.

Rehber Öğretmen İçeriği – Kuvvet Sensörü

Basınç Newton olarak ölçüldüğünde, kuvvet sensörü ile maksimum 10 Newtonluk bir kuvvet ölçülebildiğinden, robotun hareket hızı maksimum %10 olabilir.

Basınç yüzde olarak ölçüldüğünde, 10 Newtonluk basınç %100 olarak ölçülecek ve robotun hareket hızı maksimum %100 olabilecektir. Örnek programda da (Resim 4.5) görüldüğü gibi Newton olarak ölçülen basıncın 10 ile çarpılması gerekecektir.

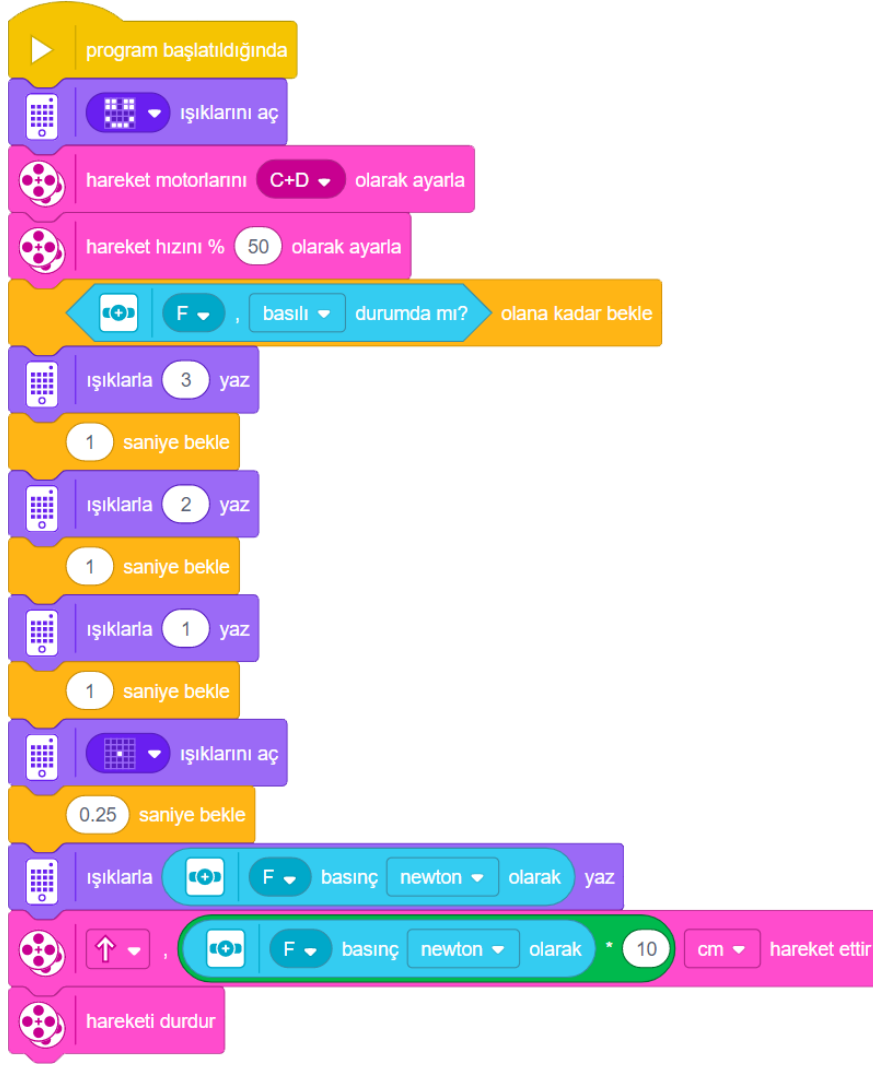


Resim 4.5 Kuvvet Sensörü 10 Newton Hareket Örnek Program

Gözle ve Uygula: Kuvvet Sensörü Oyunu

Rehber öğretmen, Resim 4.6'da gösterilen programı öğrencilerle paylaşarak aynı programı yapmalarını ister. Ardından, rehber öğretmen programın nasıl çalıştığını öğrencilere anlatır. Program çalıştırıldığında, robotun ekranında gülen yüz şeklinde ışıklar yanacaktır. Öğrenciler, robota takılı olan kuvvet sensörüne kuvvet uyguladıkları anda, robot ekranında 3'ten geriye doğru sayılmaya başlayacak ve sıfır anında ekranda bir piksel yanacaktır (Program slotlarını gösteren sıfır rakamı ile karışmaması için sadece bir pikselin yanması ile sıfır durumu gösterilmiştir). Robot, Hub ekranında tek pikselin yandığı anda kuvvet sensöründen ölçülen basınç miktarının on katı santimetre ileri gidecektir. Hareket esnasında ölçülen kuvvetin Newton cinsinden değeri ekranda görülecektir.

Öğrenciler, bir hedef uzaklık belirleyerek örneğin 70 cm (belirlenecek hedef uzaklık 10 cm'nin katları olmalıdır) uzaklığa bir Lego parçası yerleştireceklerdir. Ardından, programı çalıştırarak kuvvet sensörüne uygulayacakları basınç ile robotun istenilen miktarda giderek hedefe ulaşmasını sağlayacaklardır. Grup üyeleri, sırayla deneme yaparak robotu hedefe ulaştırmalıdır. Sonrasında, farklı uzaklıklarla aynı uygulama gerçekleştirilebilir.



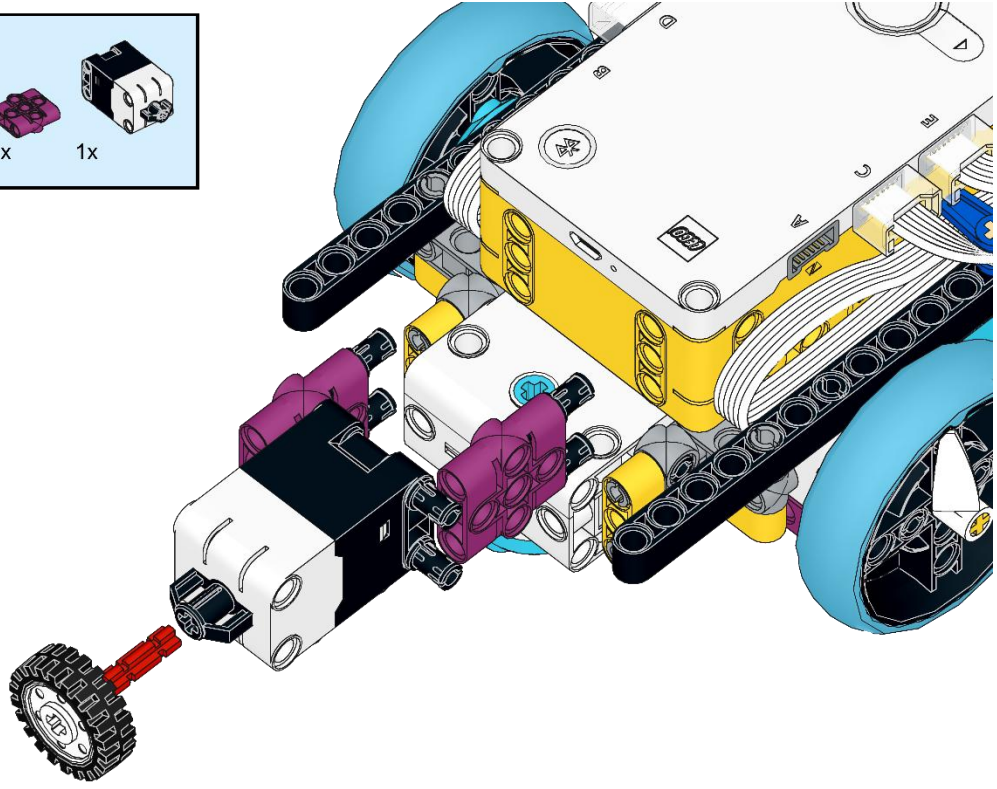
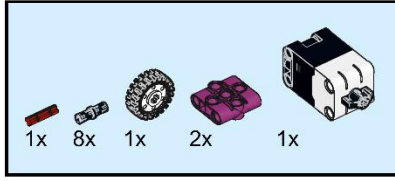
Resim 4.6 Kuvvet Sensörü Oyunu Programı

Not

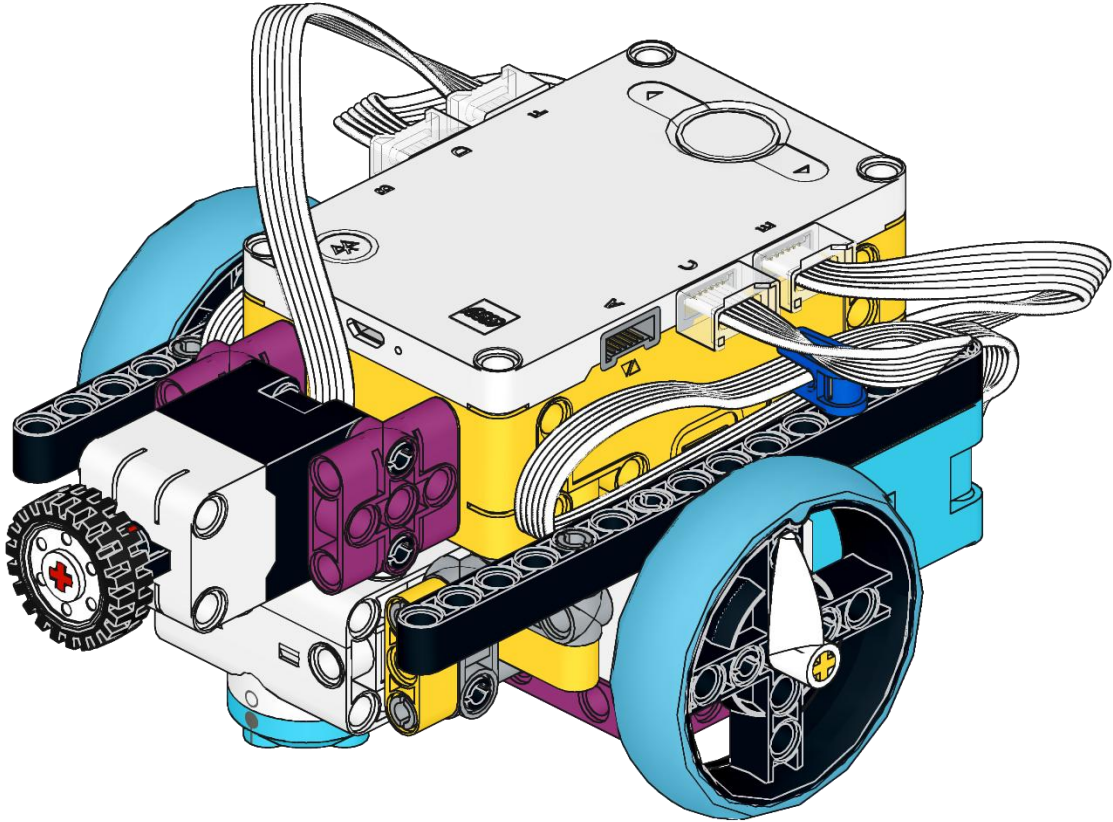
Rehber öğretmen, öğrencilerden uygulanan kuvvet miktarını yorumlamalarını isteyebilir. Günlük yaşantımızda uzunluk ve ağırlık gibi kavramlarla ilgili bir öngörümüz varken, kuvvet konusunda aynı düzeyde bir öngörümüz genellikle yoktur. Örneğin, oyunda uyguladıkları 7N'lık kuvveti baz alarak, bir pet şişeyi sıkıştırmak için en az kaç Newtonluk bir kuvvet gerektiğini tartışabilirler.

Gözle ve Uygula: Engele Çarpınca Duran Robot

Rehber öğretmen öğrencilere Resim 4.7 ve Resim 4.8'da görüldüğü gibi dokunma sensörünü sürüş modeline birleştirmelerini ve robotlarını ileri doğru giderken bir engele çarpması durumunda duracak şekilde programlamalarını ister. Örnek program Resim 4.9'da verilmiştir.



Resim 4.7 Kuvvet Sensörü Montajı Birinci Adım



Resim 4.8 Kuvvet Sensörü Montajı İkinci Adım



Resim 4.9 Engele Çarpınca Duran Robot Örnek Programı

Gözle ve Uygula: Çarpışma Testi

Rehber öğretmen öğrencilere arabalarla yapılan çarpışma testleri hakkında bilgileri olup olmadığını, öncesinde herhangi bir çarpışma testi videosu izleyip izlemediklerini sorar. Video paylaşım sitelerinden otomobillerin çarpışma testi videolarını öğrencilere izletir.

Not

youtube.com sitesinden Euro NCAP sözcükleri aratılarak uygun birkaç video izletilebilir. Örnek olarak aşağıda birkaç video bağlantısı paylaşılmıştır.

https://www.youtube.com/watch?v=2indFTVpQ_Y

<https://www.youtube.com/watch?v=vqfKw6ypvDg>

Rehber öğretmen öğrenciler ile çarpışma testlerinin neden yapıldığını, otomobillerin hasar almasına neyin sebep olduğunu, otomobillerin hızının çarpışmanın şiddetine olan etkisini öğrencilerin tartışmalarını sağlar.

Rehber öğretmen, Resim 4.10'da verilen programı öğrencilerle paylaşır ve programı açıklar. Öğrencilerden, programdaki hareket hızını %20, %40, %60, %80 ve %100 olarak değiştirmelerini ve her bir hız değeri için grafikte okunan maksimum kuvvet miktarını kaydetmelerini ister. Program, bir engele (duvara) doğru düz bir şekilde ilerleyen robotun çarpışma anında kuvvet sensörü verilerinin grafiğinin çizilmesini sağlayacaktır. Ayrıca, program robot çarpışma sonrasında engelden uzaklaştığında (öğrenciler robotu aldıklarında) motorları durduracak ve kuvvet sensörü grafiğini tam ekran olarak açacaktır.

Rehber öğretmen öğrenciler ile buldukları sonucu tartışır. Robotun hızının çarpışma anında oluşan kuvvete etkisi sorgulanır.



Resim 4.10 Çarpışma Testi Programı

TASARLA

Tasarla etkinliğinde rehber öğretmen, öğrencilerden robot tasarımlarında herhangi bir değişiklik yapmadan, düz olarak ilerleyen robotun bir engelle çarpması durumunda kendi etrafında dönerek engelden uzaklaşmasını sağlayan bir program hazırlamalarını ister. Robot engelle çarpınca yönünü değiştirme hareketini sürekli olarak yapmalı ve bu esnada kuvvet sensörü ile motor verilerinin grafiği çizilmelidir. Ardından, rehber öğretmen öğrencilerden oluşturdukları bu grafiği yorumlamalarını ister.

Tanımlama: Öğrenciler kendilerinden istenilenleri belirlemeli, robotun neler yapması gerektiğini tanımlamalı ve işlem adımlarını maddeler halinde sırlamalıdır.

Örneğin:

- Robot sürekli hareket edecek
- Robot bir engelle çarpıp çarpmadığını sürekli kontrol edecek
- Robot engelle çarptığı anda yönünü değiştirecek
- Robot sürekli kuvvet sensörü ve motor verilerini kayıt edecek
- Robot kuvvet sensörü ve motor verilerinden hangilerini kayıt edecek

Fikir üretme: Öğrenciler tanımlanan görevleri robotun gerçekleştirebilmesi için, hangi bloklarının kullanılacağına ve algoritmanın nasıl olması gerektiğine dair fikirler üretmelidir.

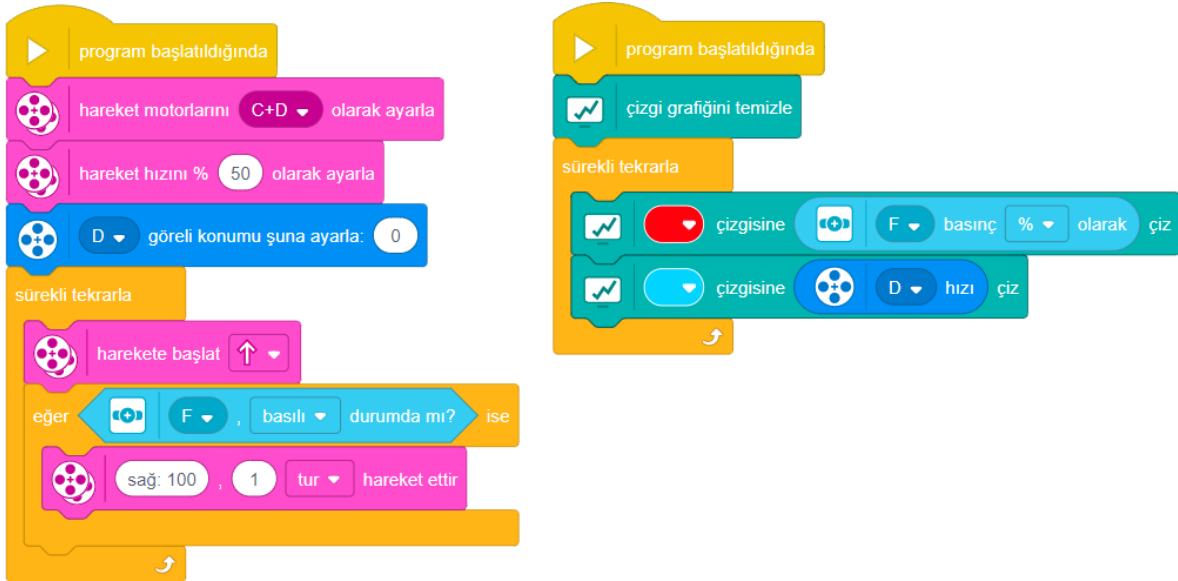
Örneğin:

- Robot sürekli hareket edecek ise sonsuz döngü ("*sürekli tekrarla*" bloğu) kullanılmalıdır.
- Robot bir engele çarpıp çarpmadığını kuvvet sensörü ile algılayabilir, "*basılı mı*" bloğu kullanılmalıdır.
- Robot engele çarptığı anda yönünü değiştirecek ise koşula bağlı bir hareket olduğundan "*eğer*" bloğu kullanılmalıdır.
- Robot sürekli kuvvet sensörü ve motor verilerini kayıt edecek ise aynı zamanda sürekli hareket edeceği için bu işlemler farklı yığıtlarda gerçekleştirilmelidir.

Öğrenciler bu süreçte farklı çözümler üretebilirler. Tüm grupların aynı veya benzer sonuçlar üretmesi zorunlu değildir. Ayrıca, öğrencilerin farklı fikirleri test etmeleri rehber öğretmenler tarafından desteklenmelidir. Örneğin, motorun konum verilerinin kaydedilmesi durumunda grafiği yorumlamak zorlaşabilir; bu gibi durumlarda rehber öğretmen, motorun hız veya güç verilerinin kaydedilmesini teşvik etmelidir.

ÜRET

Öğrenciler programın yapısı ile ilgili fikirleri ve tasarımları tamamladıktan sonra bilgisayar ve robotlarla çalışarak gerekli programı ve grafiği oluştururlar. Örnek program kodları ve programın çalıştırılması ile oluşan grafik Resim 4.11 ve Resim 4.12'de sunulmuştur. Resim 4.12'de sunulan grafikte, kırmızı çizgi kuvvet sensöründen ölçülen verilerin, mavi çizgi ise D motorunun hız verilerinin grafiğidir. Bir çarpışma gerçekleştiği anda, kırmızı çizgide tepe noktaları oluşmakta ve eşzamanlı olarak motor ters yönde dönmeye başladığından mavi grafikte negatif yönde bir artış oluşmaktadır. Rehber öğretmen diğer motorun grafiğinin nasıl olabileceğini öğrencilerin tahmin etmesini isteyebilir. Sonrasında C motorunun hız verisinin de farklı bir renkte grafiğinin çizilmesini ve yorumlanmasını isteyerek etkinliği zenginleştirebilir.



Resim 4.11 Kuvvet Sensörü ve Motor Veri Grafiği Örnek Programı

Çizgi Grafiği



Resim 4.12 Kuvvet Sensörü ve Motor Veri Grafiği

Not

Rehber öğretmen, Tasarla Üret bölümlerinde verilen program örneklerini ve ekran görüntülerini öğrencilerle paylaşmamalıdır. Tasarla Üret bölümündeki paylaşımlar, rehber öğretmenlerin etkinliklere hazırlanmasında kolaylık sağlamak amacıyla sunulmuştur. Öğrencilerin, Tasarla Üret bölümlerindeki problemlere kendileri çözüm üretmeleri gerekmektedir. Rehber öğretmenler, öğrencilere problemleri çözebilmeleri için yeterince zaman tanımalı ve doğrudan çözümü vermeden rehberlik sağlamalıdır.

DEĞERLENDİR

Günün sonunda rehber öğretmen, öğrencilere aşağıdaki soruları yönelterek bu hafta içeriğinin birlikte özetlenmesini sağlamalıdır. Bu sorular, öğrencilerin öğrendiklerini pekiştirmelerine ve konular arasındaki bağlantıları daha iyi anlamalarına yardımcı olacaktır. Ayrıca, öğrencilerin düşüncelerini ifade etmeleri ve grup içinde etkileşimde bulunmaları için bir fırsat sunar. Böylece, öğrenme süreci daha etkileşimli ve anlamlı hale gelecektir.

- Robot setinde bulunan kuvvet sensörünün özellikleri nelerdir?
- Günlük hayatta kullanılan dokunma/kuvvet/ağırlık sensörlerine örnekler verebilir misiniz? (Örneğin, otomobilin kapısı açık kaldığında veya asansörlerde yük fazla olduğunda uyarı verilmesi gibi).
- Bu gün öğrendiğiniz kod blokları hangileridir? Görevleri nelerdir?
- Kuvvetin nesneler üzerinde nasıl bir etkisi olabilir? Örnekler veriniz?
- Sizce bu hafta en zor görev hangisiydi? Zorluğun üstesinden nasıl geldiniz?

İLAVE ETKİNLİK*Kuvvet Sensörü ve Zaman Oyunu*

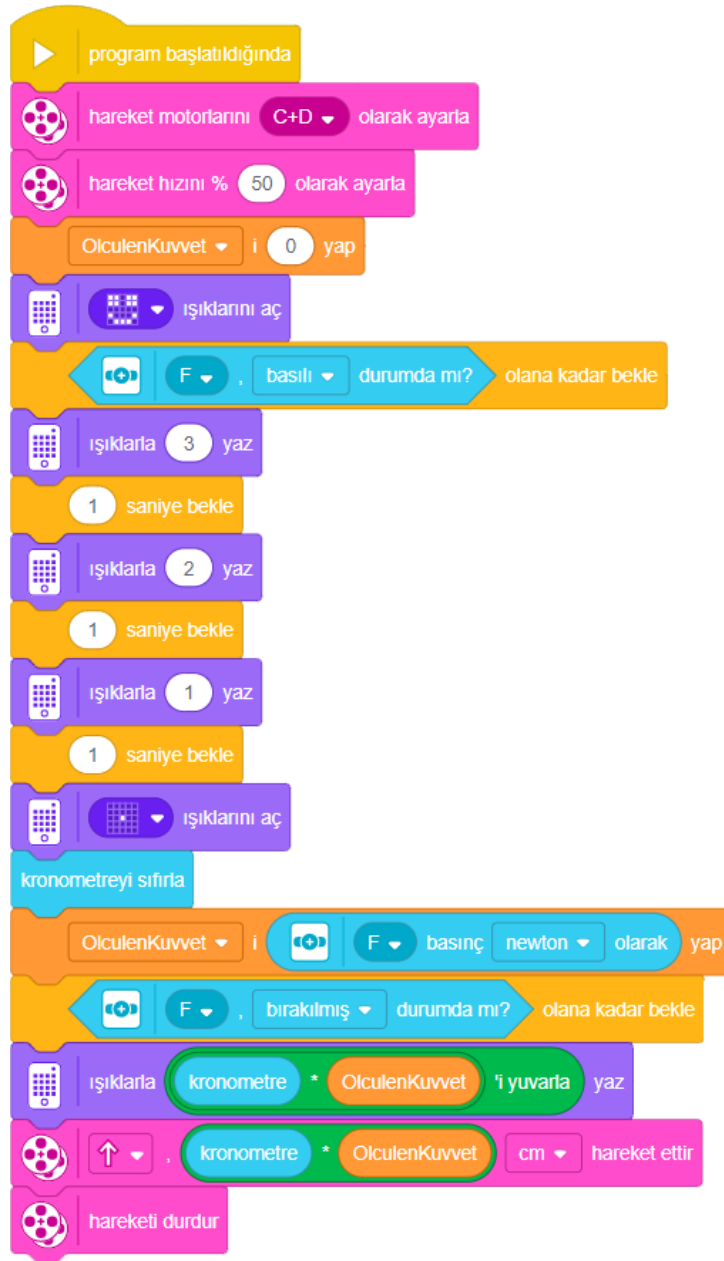
Rehber öğretmen öğrencilerden kuvvet sensörünü robotun ön kısmından sökerek, “Kuvvet Sensörü Oyunu”nda olduğu gibi serbest bir konuma almalarını ister (Resim 4.2). Rehber öğretmen “Kuvvet Sensörü Oyunu”nda robotun yalnızca 10’un katlarında hareket edebildiğini, çünkü kuvvet sensörünün uygulanan basıncı tam sayı olarak ölçebildiğini ve 10 ile çarpımı sonucunda robotun sadece 10’un katları uzunluklarda (70, 80, 90 santimetre) ilerleyebildiğini hatırlatır.

Var olan programı, ekranda tek piksel yandıktan sonra uygulanan basınç ile kuvvet sensörü bırakılncaya kadar geçen sürenin (saniye) çarpımı kadar ileri gidecek şekilde geliştirmelerini ister. Rehber öğretmen robotun basınç ve süre ölçümünü ekrandaki tek piksel yandığı anda başlayacağını ve kuvvet sensörü bırakılncaya kadar geçen süreyi hesaplamasını ve basınç sensörü bırakılınca harekete geçmesi gerektiğini öğrencilere açıklar. “Kuvvet Sensörü Oyunu”nda olduğu gibi, robotun gideceği mesafe Hub ekranına yazdırılmalıdır.

Rehber öğretmen “Kuvvet Sensörü Oyunu” programını tekrar açarak öğrencilerin hatırlamasını sağlayabilir. Bu oyun için hazırlanacak programın bir örneği program Resim 4.13’te verilmiştir. Öğrenciler benzer bir programı hazırladıktan sonra “Kuvvet Sensör Oyunu”nda olduğu gibi bir hedef belirleyip oynayabilirler. Bu defa hedefin 10 un katları olması gerekmemektedir, örneğin hedef 63 cm olarak belirlenebilir.

Not

Rehber öğretmen, öğrencilere "kronometre" bloğunu ve değişken tanımlamayı etkinlik öncesinde anlatabilir. Ayrıca, rehber öğretmen ihtiyaç duyulması halinde programla ilgili daha fazla yardımda bulunabilir.



Resim 4.13 Kuvvet Sensörü ve Zaman Oyunu Örnek Programı

5. Hafta - Mesafe Sensörü

Haftanın Amacı:

Bu haftanın amacı, öğrencilerin mesafe (distance) sensörünün çeşitli kullanım şekillerini kavrayarak, sensörü farklı görevler için programlarken gerekli düzenlemeleri yapabilmelerini sağlamaktır.

Haftanın Kazanımları:

- Öğrenciler, mesafe sensörünün çalışma mantığını ve kullanma yöntemlerini ifade eder.
- Öğrenciler, mesafe sensörünü farklı amaçlar için programlama adımlarını oluşturabilir.
- Öğrenciler, robotun farklı mesafelerdeki nesnelere göre davranması (hareket etmesi, ses çıkarması, hızını ayarlaması, Hub ekranında farklı simgeler göstermesi) için gerekli programlama adımlarını oluşturabilir.

Ekler:

Konu anlatımında ekran görüntüleri verilen örnek programlar, ayrıca dijital formatta rehber öğretmenlere sunulmuştur. Örnek program dosyalarının numaralandırılmasında, konu anlatımındaki resim numaraları temel alınmıştır. Bu hafta aşağıda sıralanmış programlar ekte rehber öğretmenlere sunulmuştur.

Program_5.4_Belirli_Bir_Mesafe_Kadar_Ilerleme.llsp3
Program_5.5_Istenilen_Mesafe_Kadar_Geri_Gitme.llsp3
Program_5.6_Engelle_Yaklaştıkça_Yavaşlayan_Robot.llsp3
Program_5.7_Yaklasan_Nesne_Olursa_Ses_Cikar
Program_5.8_Park_Sensoru.llsp3
Program_5.10_Bos_Alana_Park_Et.llsp3
Program_5.11_Ondeki_Nesneyi_Takip_Et.llsp3

Mesafe Sensörü

Rehber Öğretmen İçeriği – Mesafe Sensörü

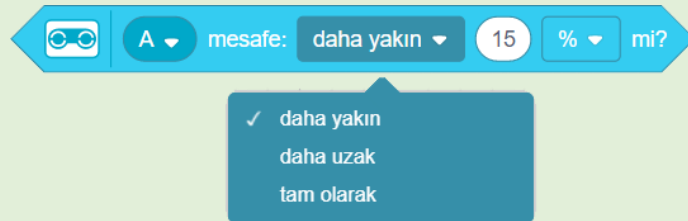
Mesafe sensörü, ileriye doğru yüksek frekanslı ses dalgaları gönderir; bu ses dalgaları insanlar tarafından duyulamaz. Gönderilen ses dalgaları, karşıdaki nesneye çarparak geri yansır ve sensör tarafından algılanır. Ses dalgasının gönderilmesi ile geri dönüşü arasındaki süre, mesafe sensörünün nesneye olan uzaklığını hesaplamak için kullanılır. Algılama aralığı yaklaşık 200 cm'dir; bu mesafeden daha uzaktaki cisimleri algılayamaz. Mesafe sensörü, uzaklık ölçümünü yüzde, santimetre veya inç cinsinden gerçekleştirebilir. Ayrıca, sensörün üzerinde dört adet programlanabilir ışıklı bölüm bulunmaktadır.



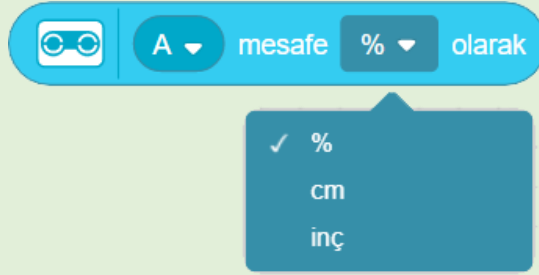
Bu hafta etkinliklerde kullanılacak mesafe sensörü ile ilgili sözcük blokları ve açıklamaları aşağıda verilmiştir.

Sensörler Blok Paleti

Mesafe Şu mu?



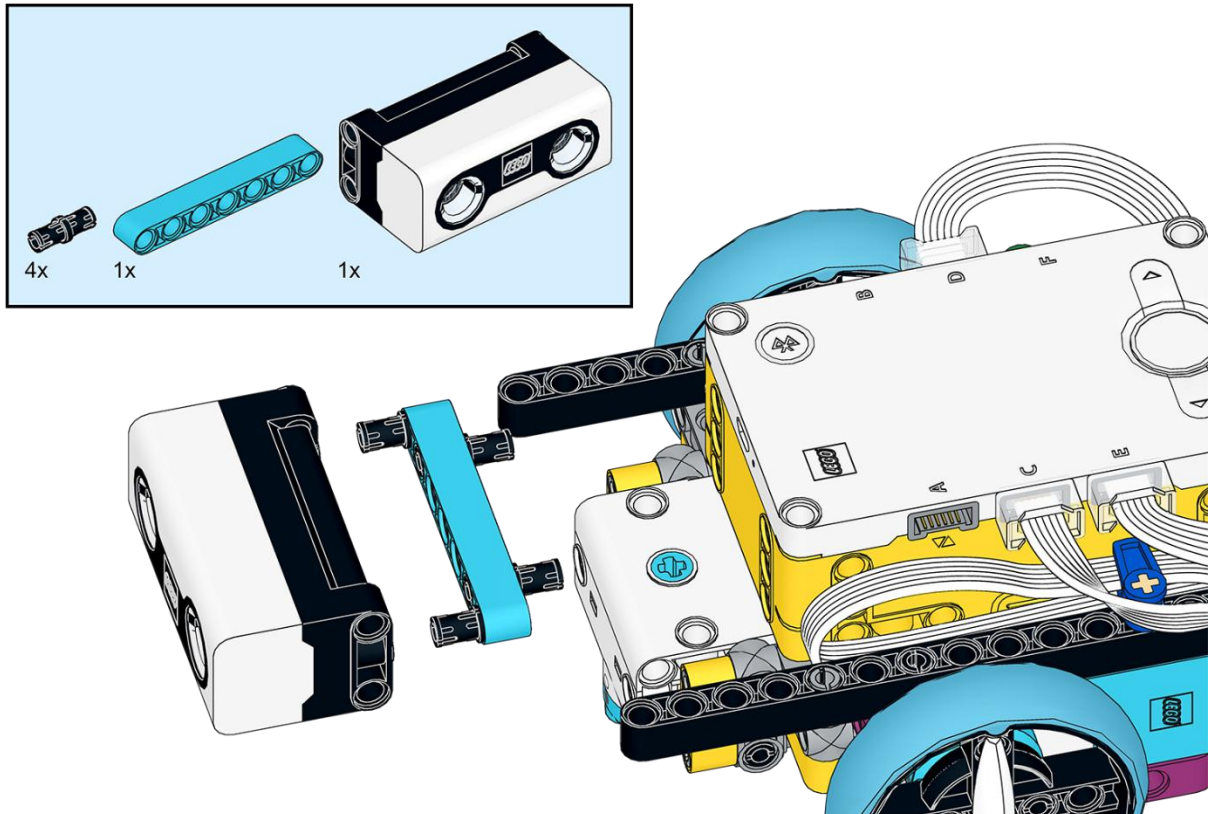
Bu blok, mesafe sensörünün santimetre, inç veya yüzde olarak belirtilen mesafeden yakın (<), uzak (>) veya eşit (=) olması durumunda "doğru (true)" sonucunu verir.

Mesafe

Bu blok, mesafe sensörünün algıladığı mevcut mesafeyi santimetre, inç veya yüzde olarak bildirir. Sensörün aralığı 0-200 cm'dir.

GÖZLE VE UYGULA**Göze: Mesafe Sensörü**

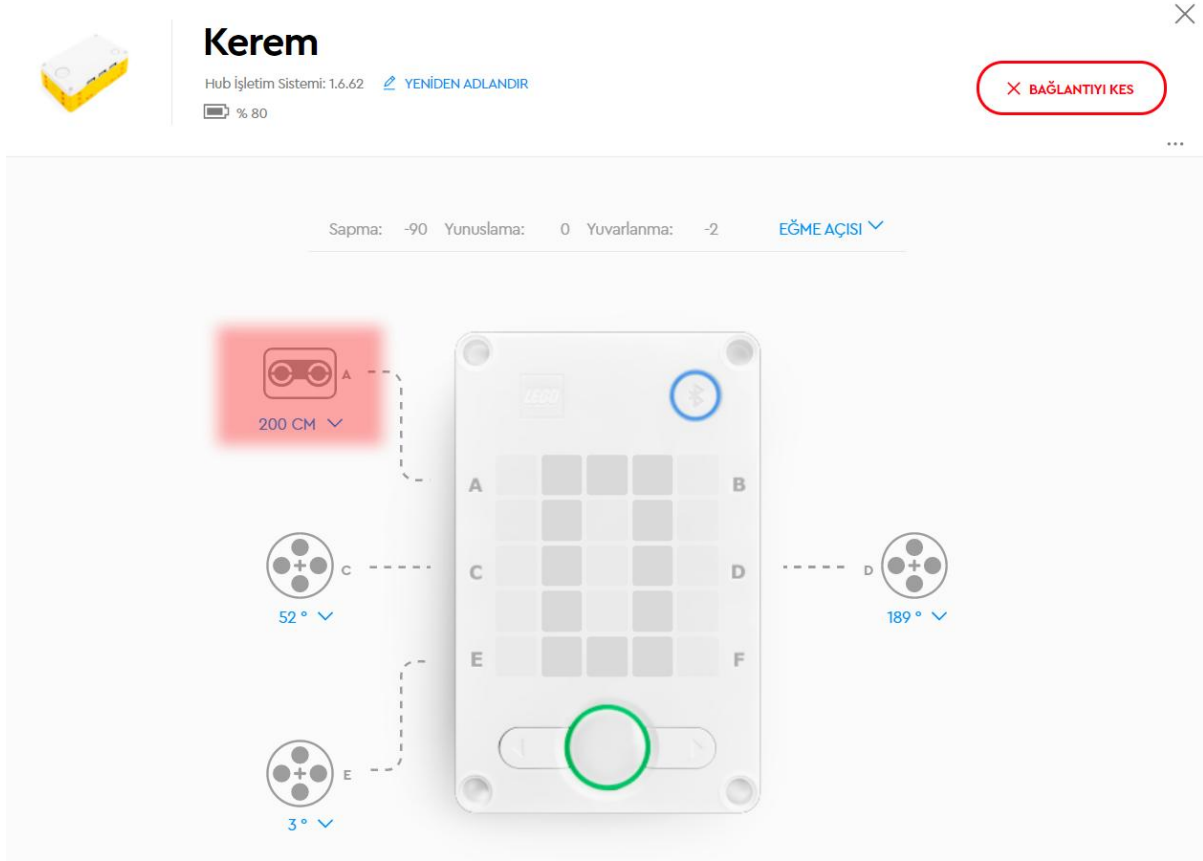
Rehber öğretmen, bu hafta yeni bir sensöre geçileceğini duyurarak öğrencilerin dikkatini çeker. Ardından, mesafe sensörünün özelliklerini detaylı bir şekilde anlatır; bu sensörün nasıl çalıştığını, algılama aralığını ve uygulama alanlarını örneklerle açıklar. Öğrencilere mesafe sensörünün Resim 5.1'deki gibi sürüş modeli robotuna nasıl entegre edileceğini gösterir.



Resim 5.1 Mesafe Sensörü Montajı

Rehber öğretmen, öğrencilere SPIKE yazılımını açmalarını ve yeni bir proje oluşturmalarını ister. Ardından, robotlarıyla Bluetooth veya kablo ile bağlantı kurmalarını ister. Bağlantılar

tamamlandıktan sonra, A portuna mesafe sensörünün takıldığını kontrol etmek için aşağıdaki 'Hub Bağlantısı Aç' seçeneğini kullanmalarını söyler.



Resim 5.2 A Portunda Mesafe Sensörü

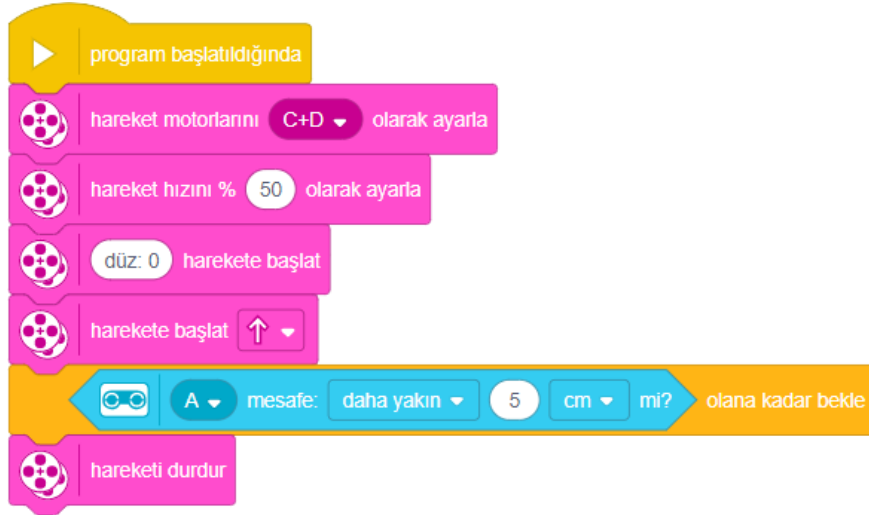
Mesafe sensörü üzerindeki ışıkların yakılması için, Resim 5.3'de gösterilen programın oluşturulması gerektiği açıklanır. Bu süreçte öğrencilere "ışıklarını yak" bloğunun temel işlevleri kısaca anlatılır. Öğrencilerin, öğrendiklerini pekiştirmek amacıyla farklı denemeler yapmaları için cesaretlendirilir.



Resim 5.3 Mesafe Sensörünün Işıklarının Yakılması

Göze: Belirli Bir Mesafeye Kadar İlerleme

Rehber öğretmen, robotun A numaralı porta bağlı mesafe sensöründen gelen uzaklık değeri 5 cm'den küçük olana kadar ilerlemesini ve bu mesafe 5 cm'den az olduğunda durmasını sağlayan Resim 5.4'deki programı hazırlar.

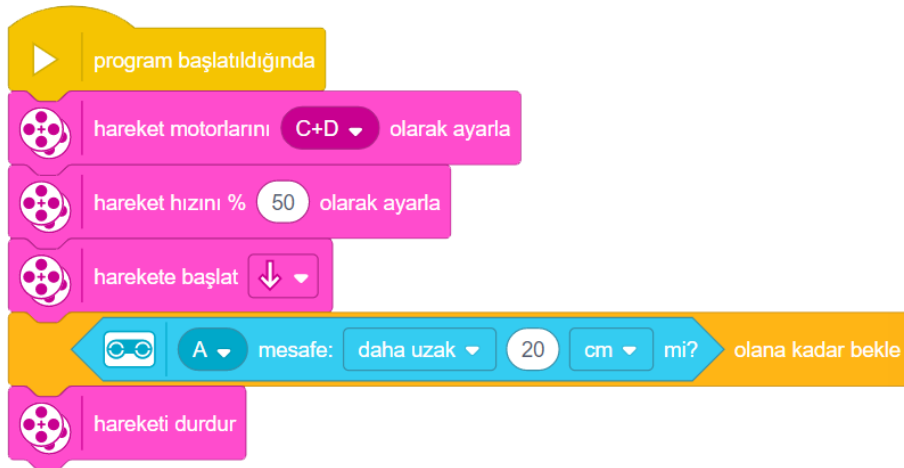


Resim 5.4 Belirli Bir Mesafe Kadar İlerleme

Bu adımda, programın nasıl yazıldığı ekrana yansıtılarak öğrencilere gösterilir ve çalışma mantığı açıklanır. Rehber öğretmen, öğrencilerden yukarıdaki blokları SPIKE programlama alanına sürükleyip bırakmalarını ve robotu çalıştırmalarını ister. Kod çalıştıktan sonra, nesne ile mesafe sensörü arasındaki mesafe ölçülür ve öğrencilerden buldukları değeri (ilk örnek için 5 cm) bir kâğıda yazmaları istenir. Ardından, aynı işlemler 10 cm, 15 cm ve 20 cm için tekrarlanır. Öğrencilere elde ettikleri verilerin anlamı sorulur ve robotun verilen değerde tam olarak duramayacağı, hata payı bulunduğu açıklanır.

Uygula: İstenilen Mesafe Kadar Geri Gitme

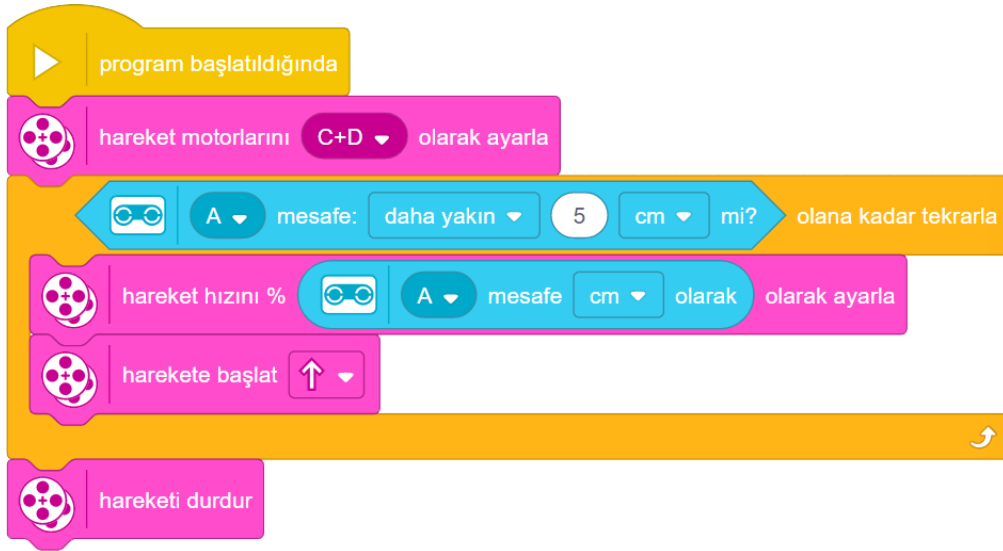
Bu uygulamada, öğrencilerden mesafe sensörünü kullanarak robotlarının, önündeki engelden 20 cm geriye gittikten sonra durmasını sağlayacak bir program oluşturmaları istenir. Öğrenciler, Resim 5.5’de gösterilen programı oluşturmaları için yönlendirilir. Programlarını hazırladıktan sonra çalıştırmaları ve sonuçları gözlemlemeleri istenir. Robot çalıştırıldıktan sonra, öğrencilerden aldıkları mesafeyi ölçmeleri ve bu mesafeyi 20 cm’ye olabildiğince yaklaştırmak için programda değişiklik yapmaları talep edilebilir.



Resim 5.5 İstenilen Mesafe Kadar Geri Gitme

Uygula: Engele Yaklaştıkça Yavaşlayan Robot

Bu etkinlikte, karşısındaki engele yaklaştıkça yavaşlayan ve belirli bir mesafe kaldığında durabilen bir robot programı oluşturulacaktır. Öğrencilere, mesafe sensörü ile ölçülen uzaklık değerinin robotun hareketini sağlayan motorların güç değeri olarak aktarıldığında, mesafe azaldıkça motorların gücünün de azalacağı ve böylece robotun engele yaklaştıkça yavaşlayacağı detaylı bir şekilde açıklanır. Resim 5.6'da görülen programın, engele yaklaştıkça yavaşlayan ve 5 cm mesafede durabilen bir robot için hazırlanabilmesi amacıyla öğrencilere yeterli süre tanınır. Gerekli görüldüğünde, programın mantığı tekrar gözden geçirilerek öğrencilere daha iyi kavratılması sağlanır.



Resim 5.6 Engele Yaklaştıkça Yavaşlayan Robot

Gözle: Robota Yaklaşan Nesne Olursa Ses Çıkar

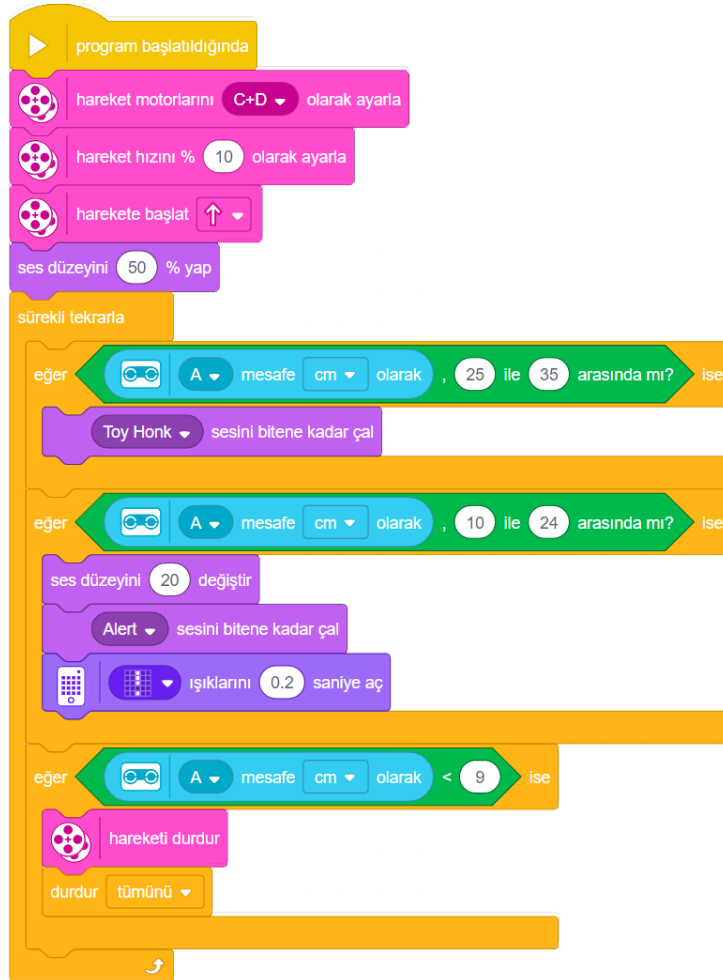
Rehber öğretmen, Sürüş Modeli robotuna 5 cm'den daha az bir mesafede bir nesne yaklaştığında ses çalan bir program yazacağını duyurur. Öncelikle, mesafe sensörünün düzgün çalıştığını kontrol eder ve programlama alanına 'sürekli tekrarla' döngüsünü sürükleyip bırakır. Bu döngünün kullanım amacını açıklayarak, programın sürekli olarak mesafe değerini kontrol etmesi gerektiğini vurgular. Ardından, mesafenin 5 cm'den az olup olmadığını kontrol edecek 'eğer ise' bloğunu ekler. Çalmak istediği sesi kütüphaneden seçerek kodunu tamamlar ve koşul doğru olduğu sürece sesin çalındığını öğrencilere gösterir.



Resim 5.7 Yaklaşan Nesne Olursa Ses Çıkaran Robot

Uygula: Park Sensörü

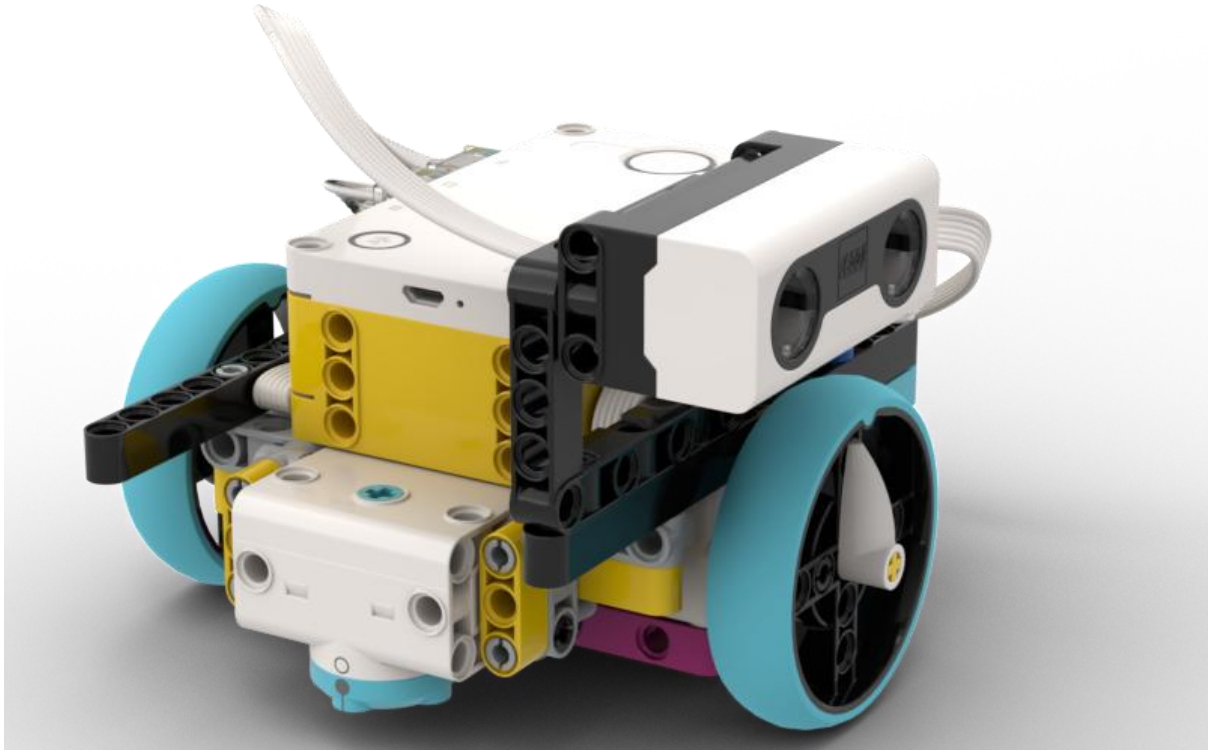
Bu etkinlikte, öğrencilerden robotu taşıtlardaki park sensörlerine benzer bir şekilde programlamaları istenir. Robot, belirli bir mesafeye yaklaştığında uyarı vermeli, mesafe azaldıkça ses seviyesi yükselmeli ve Hub ekranında '!' işareti görüntülenmelidir.



Resim 5.8 Park Sensörü Programı

TASARLA VE ÜRET

Bu etkinlikte, öğrencilerin robotun yol kenarında boş bir alana, park eden araçlar gibi, hareket halindeyken sağ veya sol tarafından (mesafe sensörünün takıldığı yönde) park etmesini sağlamaları amaçlanmaktadır. Öğrencilerden, Resim 5.9'daki gibi mesafe sensörünü yan tarafa bakacak şekilde monte etmeleri istenir. Düz bir şekilde ilerleyen robotun üzerindeki mesafe sensörü, bir boşluk algıladığında, robotun ilgili yöne dönerek paralel park etmesi beklenmektedir. Bu aşamada, gerektiğinde rehber öğretmen onlara destek sağlayarak süreci kolaylaştırabilir. Ders programı boyunca, tasarım sürecinde olduğu gibi, öğrencilerin aşağıda örnek olarak verilen adıma benzer bir süreç gerçekleştirmeleri gerekecektir.



Resim 5.9 Mesafe Sensörünün Yana Takılması

Tanımlama: Öğrencilerin öncelikle istenilen işlemler için neler gerektirdiğini belirlemesi ve bunları sıralaması gerekir. Örneğin;

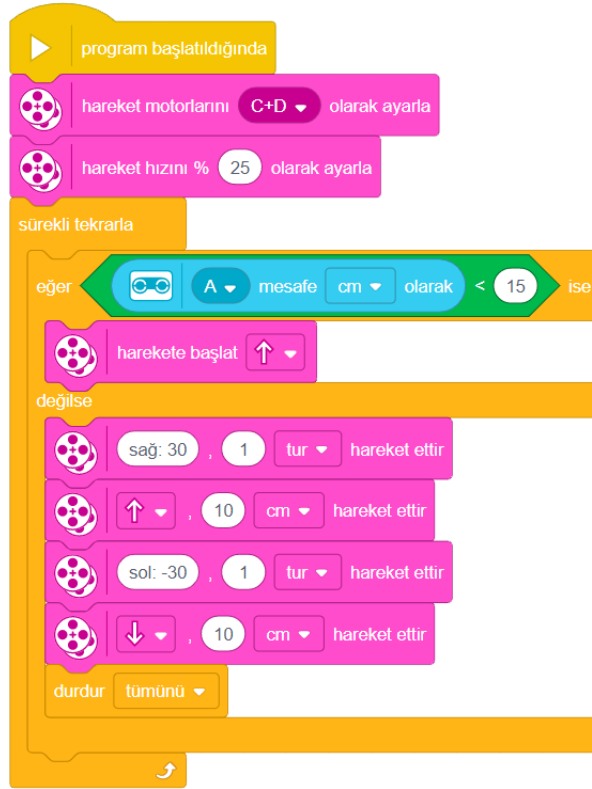
- Robot mesafe sensörü ile duvar veya kapalı alanı arasındaki mesafeyi sürekli kontrol etmeli.
- Robot düz ilerlemeli.
- Robot hareket halinde iken duvar tarafında boşluk olması durumunda ilgili tarafa dönerek park etmelidir.

Fikir üretme: Bu aşamada öğrencilerin yukarıda belirlenen işlemlerin nasıl yapılabileceği ile ilgili fikir yürütmesi gerekir. Örneğin, öğrenciler aşağıdaki maddelere benzer fikirler üretebilir.

- Duvarla arasındaki mesafeyi sürekli kontrol etmek için mesafe sensörü ve döngü kullanılabilir.
- Düz ilerlerken boşluk olduğunu anlamak için kontrol ifadesi kullanılabilir.

- Robotun boşluğa doğru dönmesi sağlanmalıdır.
- Dönme tamamlandıktan sonra robotun park etmesi sağlanmalıdır.

Rehber öğretmen gözetiminde, öğrenciler çözüm için tasarımlarını yaptıktan sonra bilgisayar ve robot başında çalışarak istenilen görevi yerine getirmelidir. Bu süreçte, örnek bir program Resim 5.10'da sunulmuştur.



Resim 5.10 Boş Alana Park Et Örnek Programı

DEĞERLENDİR

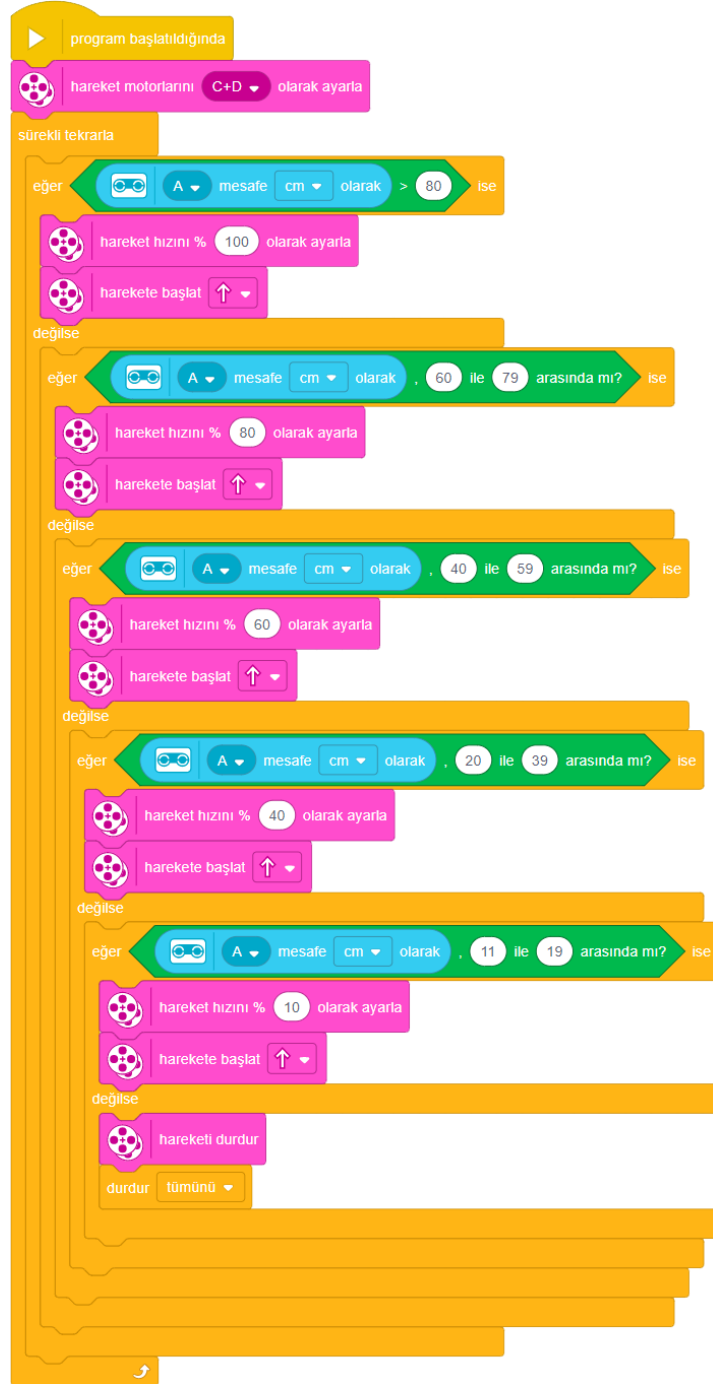
Bu aşamada hedef, her haftada olduğu gibi öğrencilerin öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşüncelerini sağlamaktır. Bu sayede öğrenciler; problem çözme yetenekleri, dersin konusu ve kendileri ile ilgili gözlemler yaparak öğrendikleri yeni konuları ve kendilerini değerlendirebilir. Öğrencilerden şu soruları yanıtlamaları istenir:

- Bugün 'tasarla ve üret' adımı size verilen problem nedir? Problemi kendi cümlelerinizle ifade edin.
- En çok hangi görevde zorlandınız? Bu zorlukların üstesinden nasıl geldiniz?
- Problemi çözerken ne gibi sıkıntılar yaşadınız ve bunların üstesinden gelmek için neler yaptınız?
- Kullandığınız yöntemler bu sıkıntıları gidermede başarılı oldu mu?
- Grup arkadaşınızla anlaşmazlığa düştüğünüz durumlar oldu mu ve bunların üstesinden gelmek için neler yaptınız?
- Grup arkadaşınızdan ne/neler öğrendiniz?

İLAVE ETKİNLİK

Öndeki Aracı/Nesneyi Takip Eden Robot

Bu etkinlikte, robotun önündeki bir nesneyi takip etmesi sağlanacaktır. Başka bir deyişle, robotun önündeki nesne ilerledikçe, robotun da bu nesneye paralel olarak hareket etmesi beklenmektedir. Resim 5.11'deki örnek program kodunda görüldüğü gibi, kontrol ifadeleri kullanılarak robot ile nesne arasındaki mesafe arttıkça robotun hızının artması sağlanmalıdır. Eğer robot ile nesne arasındaki mesafe istenilen seviyeden çok yakınsa (10 cm'den az), robotun durması gerekmektedir.



Resim 5.11 Öndeki Nesneyi Takip Eden Robot Programı

6. Hafta - Hareket Sensörü

Haftanın Amacı:

Bu haftanın amacı öğrencilerin hareket (gyro) sensörünü kullanarak, Hub'ın üç eksenindeki dönme hareketlerini programlarında kullanabilmelerini sağlayacak temel bilgi ve becerileri kazandırmaktır. Öğrencilerin Hub'ın yunuslama, yuvarlanma ve sapma açılarını kullanarak farklı programlar yapabilmeleri amaçlanmıştır. Ayrıca Hub'ın ön, arka, üst, alt, sağ ve sol yönlerinin yukarı olması durumunda ve Hub'ın sallanma, tıklatılma ve düşme durumlarının mantık operatörlerinde kullanılması ve program yığıtlarının çalıştırılması amacıyla kullanılması örnekendirilecektir.

Haftanın Kazanımları:

- Öğrenciler hareket sensörünün nasıl çalıştığını ifade edebilirler.
- Öğrenciler hareket sensörü ile farklı eksenlerdeki dönme açısını tespit ederek program içerisinde kullanabilirler.
- Öğrenciler robotun herhangi bir ekseninde istenilen açı miktarında dönmesini sağlayabilirler.
- Öğrenciler Hub'ın herhangi bir yönünün yukarıda olması koşulunu mantık işlemlerinde ve program yığıtlarının başlatılmasında kullanabilirler.
- Öğrenciler Hub'ın sallanma, tıklatılma ve düşme durumlarını koşullarda ve yığıtların başlatılmasında kullanabilirler.

Kullanılacak Malzemeler:

Robot seti, bilgisayar ve çalışma alanı (mat).

Ekler:

Konu içerisinde ekran görüntüleri sunulan programlar ayrıca SPIKE uygulaması formatında (.llsp3 dosyası olarak) rehber öğretmenlere sunulmuştur. Ekte sunulan örnek programların numaraları, konu içerisindeki resim numaralarıyla eşleştirilmiştir. Bu hafta aşağıda sıralanan programlar, çalışma alanı (mat) ve video dosyası ekte rehber öğretmenlere sunulmuştur. Rehber öğretmen etkinlik öncesinde her grup için bir tane Mat_6_A4.pdf dosyasının çıktısını almalıdır.

Program_6.7_90_Derece_Hareketi.llsp3
Program_6.9_Kare_Hareketi.llsp3
Program_6.10_Ok_Isareti_Yukari.llsp3
Program_6.12_Su_Terazisi_Isiklari.llsp3
Program_6.15_Tasarla_Uret_Hedefe_Kilitlen.llsp3
Program_6.17_Sallaninca_Rastgele_Sayi.llsp3
Program_6.19_Yonunu_Kaybetmeyen_Robot.llsp3
Video_6.1_Tasarla_Uret_Gorev.mp4
Mat_6_A4.pdf

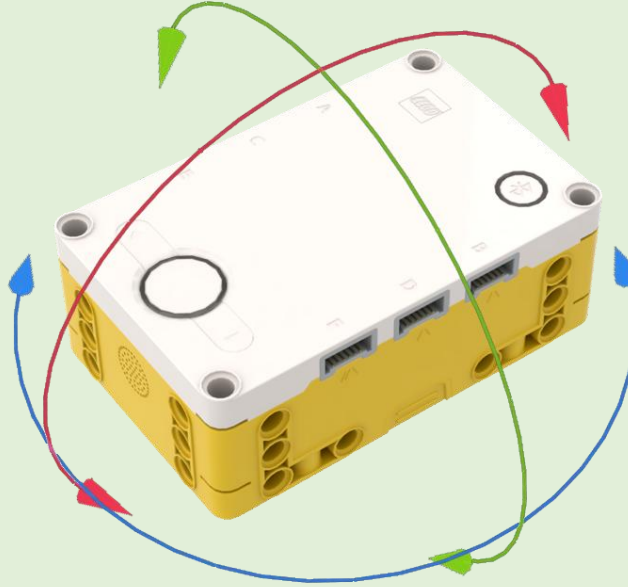
Hareket Sensörü

Rehber Öğretmen İçeriği – Hareket Sensörü

Gyro sensörü olarak da isimlendirilen hareket sensörü açısal dengenin korunması ilkesi ile çalışan ve yön ölçüm işlemlerinde kullanılan sensörlerdir. Günümüzde akıllı telefonlar, navigasyon sistemleri, oyun kumandaları ve robotlar gibi birçok teknolojik üründe kullanılmaktadırlar.

SPIKE Prime robot setinde hareket sensörü Hub ile bütünleşiktir. EV3 robot setinde olduğu gibi ayrı bir sensör olarak robot setinde yer almamaktadır. Ayrıca EV3 robot setinde yaşanan kayma problemi (robotun dönme hareketi durduğu halde, sensörün açı değerinin artmaya devam etmesi) SPIKE Prime setinde giderilmiştir.

Ayrıca EV3 setinde yalnızca bir eksenle açı ölçülebiliyor iken, SPIKE Prime setinde altı eksenli hareket sensörü bulunmaktadır. SPIKE Prime robot setindeki hareket sensörü, üç eksenli ivmeölçer ve üç eksenli jiroskop sensörlerini içermektedir. Böylece bütünleşik hareket sensörü ile üç eksenli dönme hareketinin açısal değeri ve ivmesi ölçülebilmektedir. Ayrıca sensör Hub'ın sallandığını, dokunulduğunu ve düştüğünü algılayabilmektedir. Resim 6.1'de üç eksenli dönme hareketi gösterilmiştir.



Resim 6.1 Hub'ın Üç Eksendeki Dönme Hareketi

Bu hafta etkinliklerde kullanılacak hareket sensörü ile ilgili kod blokları ve açıklamaları aşağıda verilmiştir.

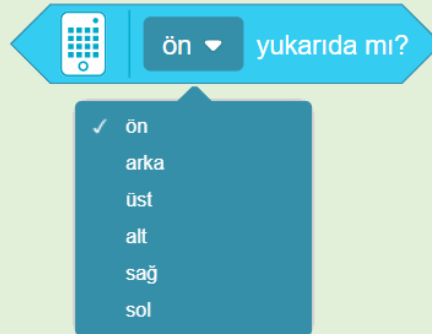
Sensörler Blok Paleti

Eğildi mi?



Bu blok bir “Boole Bloğu” olduğundan, Hub’ın seçilen yönlerde (dört yön oklarından herhangi biri), herhangi bir yönde (sol üst köşedeki oklar simgesi) veya düz durması (sağ üst köşedeki Hub simgesi) durumlarında doğru (true) sonucunu verir.

Hub yönü şu mu?



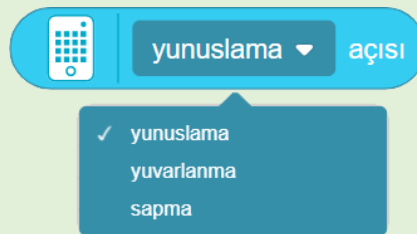
Hub seçilen yönü yukarı geldiğinde doğru (true) sonucunu verir. Hub’ın ekranın bulunduğu yönü **üst**, pilin takıldığı yüzeyi **alt**, şarj girişinin olduğu yönü **arka**, hoparlörünün olduğu yönü **ön**, B,D,F girişlerinin olduğu yön **sağ** ve A,C,E girişlerinin olduğu yön ise **soldur**. Resim 6.2’de Hub’ın yönleri gösterilmiştir.



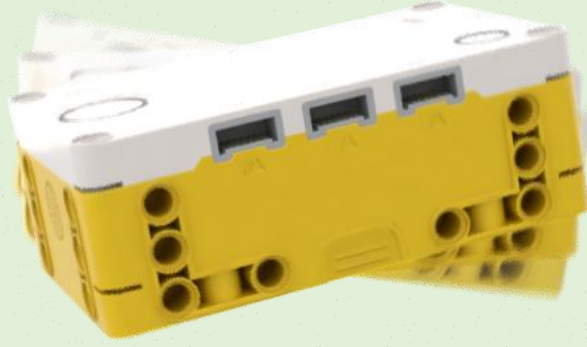
Resim 6.2 Hub'ın Yönleri

Sallanıyor mu?

Bu blok Hub'ın belirlenen sallanma, tıklatılma ve düşme hareketlerini gerçekleştirdiğinde doğru (true) sonucunu verir.

Hub Yunuslama Yuvarlanma Sapma Açısı

Bu blok, Hub'ın **yunuslama**, **yuvarlanma** ve **sapma** açı değerlerini verir. Hub'ın üç eksenindeki dönmesini ifade eden yunuslama (Resim 6.3), yuvarlanma (Resim 6.4) ve sapma (Resim 6.5) hareketleri aşağıda sunulmuştur.



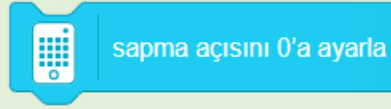
Resim 6.3 Yunuslama



Resim 6.4 Yuvarlanma



Resim 6.5 Sapma

Hub Sapma Açısını 0 Olarak Ayarla

Bu blok Hub'ın o anki sapma açısını sıfır (0) olarak ayarlar. Bu blok çalıştığı anda Hub'ın ön-arka yönünün doğrultusu sıfır derece olarak kabul edilecektir.

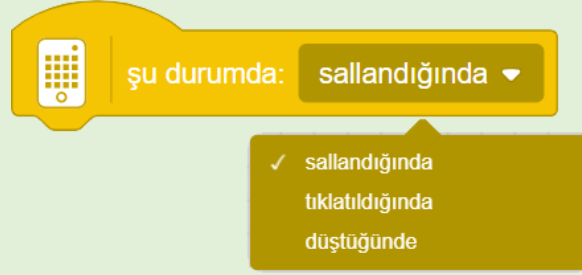
Olaylar Blok Paleti**Eğildiğinde**

Bir "Şapka Bloğu" olan "eğildiğinde" bloğu, Hub'ın düz bir konumda iken, belirlenen yönlerde bir hareketin gerçekleşmesi durumunda kendisine bağlı olan tüm blokları çalıştırır.

Bu blok sadece belirtilen olay durumunda tetiklenir. Hub yeni bir yönde eğilmediği sürece yeniden tetiklenmez.

Hub Yönü Yukarı Baktığında

Bu blok seçilen yönün yukarı olması durumunda, bağlı olan tüm blokları çalıştırır. Bu blok sadece belirtilen olay durumunda tetiklenir. Tekrar tetiklenmesi için Hub'ın yönünün değişmesi gereklidir.

Hub Sallandığında

Bu blok, belirlenen durumun (sallanma, tıklatılma veya düşme) gerçekleşmesi durumunda kendisine bağlı olan tüm blokları oynatır. Hub'ın hareketinin değişmemesi durumunda blok yeniden tetiklenmez.

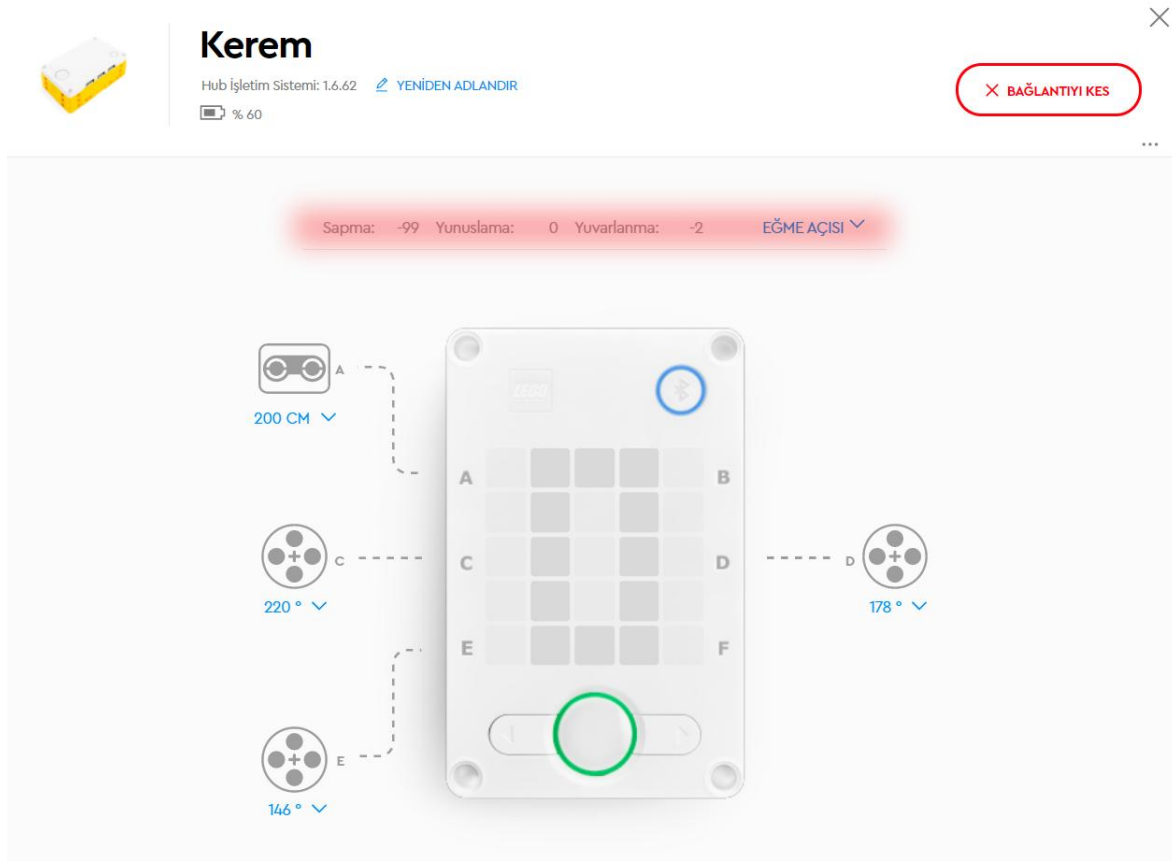
Not

*Ekte sunulan örnek programlar ve bu programlara ait ekran görüntüleri, rehber öğretmenlerin etkinliklere hazırlık sürecinde kullanmaları amacıyla hazırlanmıştır. Rehber öğretmenin görevi, gözle etkinlikleri sırasında sunulan programları öğrencilere açıklayarak adım adım gerçekleştirmektir. Üret etkinlikleri için sunulan örnek programlar ve ekran görüntüleri **kesinlikle öğrencilerle paylaşılmamalıdır**. Tasarla ve üret etkinliklerinde, öğrencilerin verilen probleme kendi çözümlerini üretmeleri beklenmektedir. Bu süreçte rehber öğretmen, doğrudan çözümü sunmadan öğrencilere rehberlik etmelidir.*

GÖZLE VE UYGULA*Gözle ve Uygula: Robotun 90 Derece Döndürülmesi*

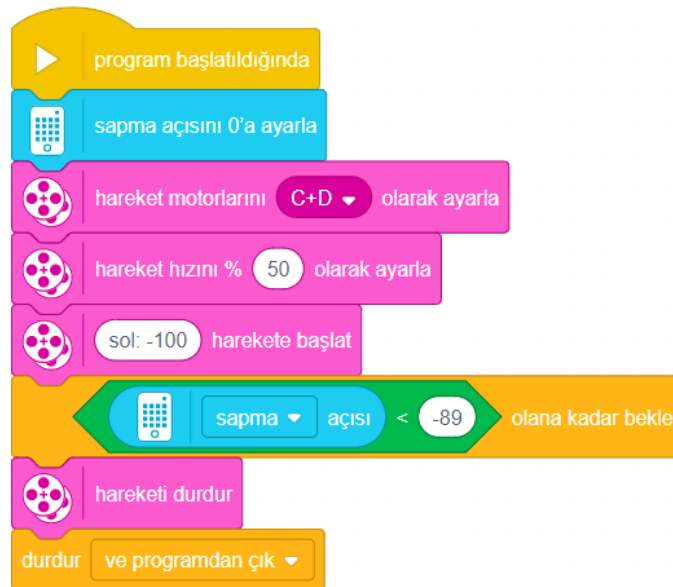
Rehber öğretmen öğrencilere bu hafta robotun hareket sensörünü kullanarak etkinlikler yapacaklarını belirtir. Hareket sensörünün Hub'ın içerisinde bütünleşik olduğunu ve diğer sensörler gibi robotlarına birleştirmelerine gerek olmadığını belirtir.

Rehber öğretmen öğrencilerden SPIKE Prime Uygulamasını açmalarını ve robotlarını Bluetooth veya USB kablosu ile bilgisayar bağlantısını yapmalarını ister. Hub bağlantı penceresini açarak sapma, yunuslama ve yuvarlanma açılarının robotun hareketine göre nasıl değiştiğini gözlemlenmelerini ister. Rehber öğretmen bu hareketleri açıklamadan öğrencilerin keşfetmelerine olanak sağlar. Resim 6.6'da robotun dönme hareketlerine göre Eğme Açısının (sapma, yunuslama ve yuvarlanma açılarının) değişimi gösterilmiştir.



Resim 6.6 Eğme Açısının Gözlemlenmesi

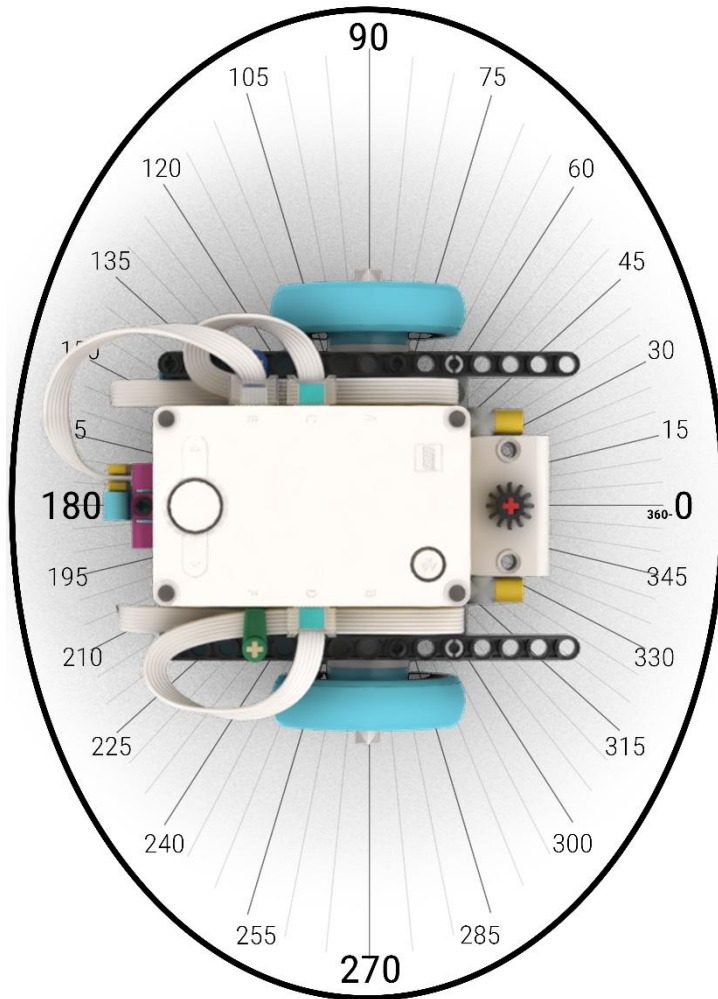
Rehber öğretmen Sürüş Modeli robotunun 90 derece dönmesini sağlayan Resim 6.7’de sunulan programı hazırlayarak Resim 6.8’de gösterildiği gibi ekte sunulan çalışma alanı (açıölçer) üzerinde çalıştırır.



Resim 6.7 Doksan Derece Dönme Örnek Program

Not

Çalışma alanı (açıölçer) öğrencilerde kavram yanlışlığına neden olmamak için ortaokul matematik derslerindeki açılar konusu temel alınarak hazırlanmıştır. Sürüş Modeli tasarımında ise robotun sapma açısı açıölçerin tersi yöndedir. Robotun sapma açısı, saat yönünde artmakta, saatin tersi yönde azalmaktadır. Bu nedenle Resim 6.7’de sunulan program negatif açı değeri ile hazırlanmıştır. Rehber öğretmen programda neden negatif açı değeri kullanıldığını öğrencilere açıklamalıdır.



Resim 6. 8 Doksan Derece Dönme Çalışma Alanı

Rehber öğretmen, öğrencilerden aynı programı hazırlayıp çalışma alanı üzerinde robotlarının kaç derece döndüğünü ölçmelerini ve bu ölçümü kaydetmelerini ister. Ayrıca hareket hızını artırarak ve azaltarak farklı ölçümler yapmalarını ister. Sapma açısı değerinin neden 90 dereceden daha büyük olduğu tartışılır. Robotun tam olarak 90 derece dönebilmesi için programda nasıl düzenlemeler yapılabileceği öğrenciler ile tartışılır.

Rehber Öğretmen İçeriği – Robot Neden 90 Dereceden Fazla Döner

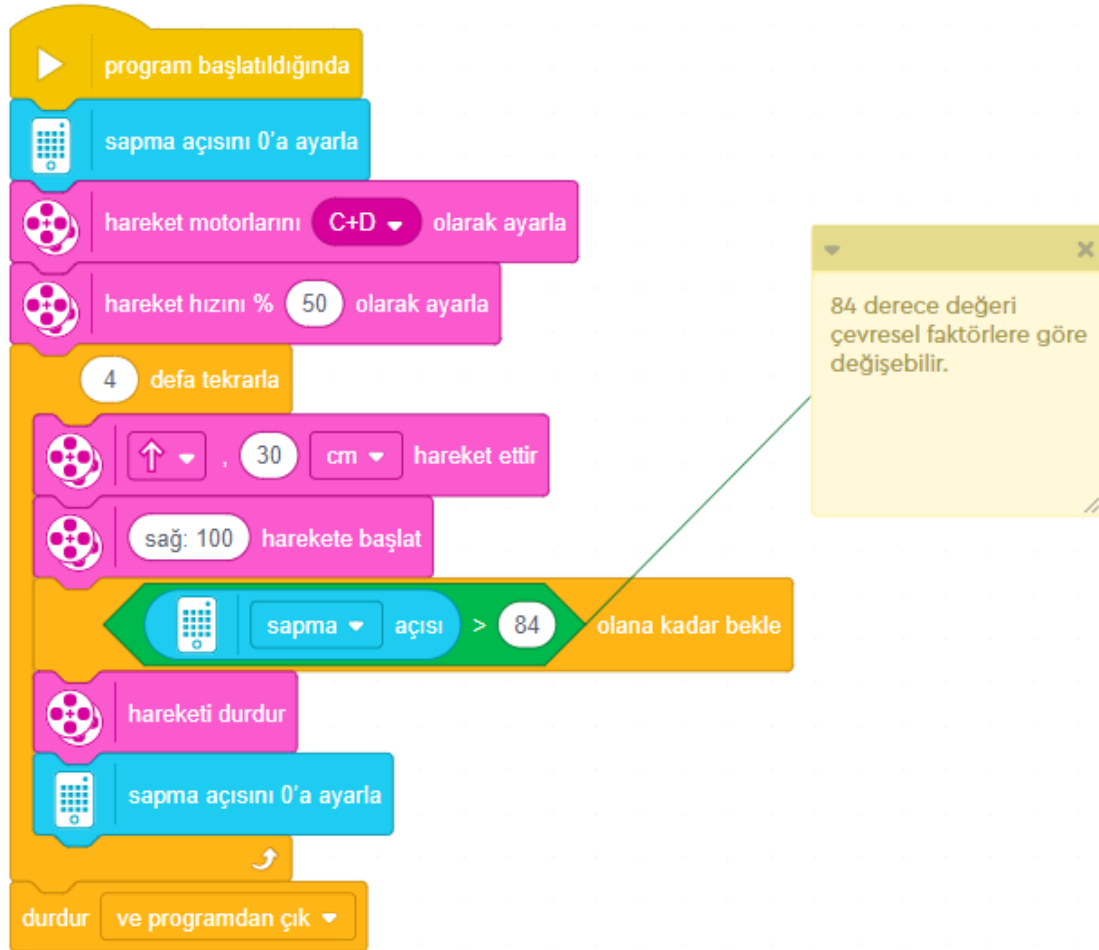
Sensör tam olarak 90 dereceyi ölçtüğü anda bu verinin aktarılması ve motorlara durma komutunun gönderilmesi zaman alır. Bu nedenle robot 90 dereceden fazla döner.

Robot kendi etrafında dönerken oluşan dönme kuvveti, motorlar durdurulduktan sonra da robotun biraz daha dönmesine sebep olur. Dönme kuvvetinin etkisini azaltmak için robotun daha yavaş dönmesi sağlanabilir.

Robotun 90 dereceden daha az bir açı ile dönecek şekilde programlanması bir çözüm olabilir.

Uygula: Kare Şeklinde Hareket

Rehber öğretmen öğrencilerden bir kenarı 30 cm olan bir kare şeklinde hareket edecek robotun programını hazırlamalarını ister. Robotun başlangıç noktasında, hareketini tamamlaması gerektiğini bu nedenle köşelerdeki dönme hareketlerinin tam 90 derece olması gerektiğini belirtir. Örnek program Resim 6.9'da sunulmuştur.



Resim 6.9 Kare Şeklinde Hareket Örnek Program

Gözle ve Uygula: Sürekli Yukarıyı Gösteren Ok

Rehber öğretmen öğrencilere Hub'ın yönünü algılayabildiğini belirtir ve “Hub yönü şu mu?” bloğu ile yukarıya bakan yönün kontrol edilebildiğini ve kontrol sonucunda doğru veya yanlış sonuçlar üretebildiğini belirtir. Hub'ın altı yönünü Resim 6.2'de gösterildiği gibi öğrencilere anlatır.

Rehber öğretmen öğrencilerden “Hub yönü şu mu?” bloğunu kullanarak robotun ekranında sürekli yukarıyı (sınıfın tavanını) gösterecek bir okun sembolü bulunan programı yapmalarını ister. Robot yan veya dik konuma geldiğinde ekrandaki ok değişerek yukarıyı (sınıfın tavanını) göstermeye devam etmelidir. Örnek program Resim 6.10'da sunulmuştur.



Resim 6.10 Ok İşareti Yukarı Örnek Program

Gözle ve Uygula: Yunuslama, Yuvarlanma Işıkları

Rehber öğretmen öğrencilere daha öncesinde su terazisi görüp görmediklerini ve su terazisinin ne amaçla kullanıldığını sorar. Tüm öğrencilerin su terazisinin nasıl çalıştığını anladığından emin olduktan sonra benzer şekilde robotun eğimini gösteren bir uygulama gerçekleştireceklerini belirtir.



Resim 6.11 Su Terazisi

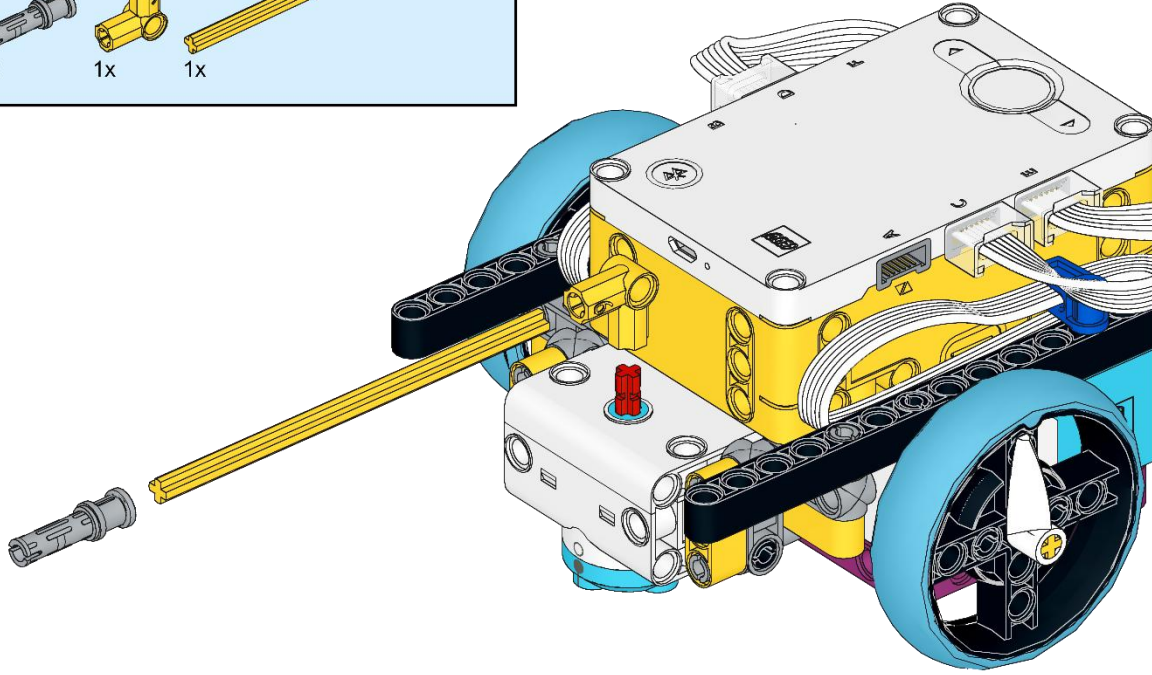
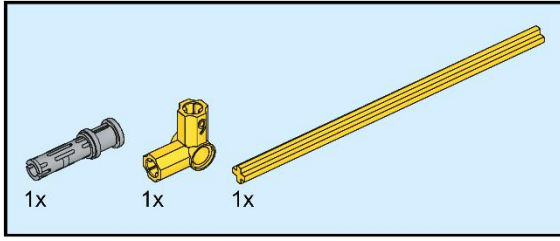
Robotun yunuslama veya yuvarlanma hareketi yapması durumunda bu hareketlerin yönünü ışıkları ile gösteren robot programının nasıl yapılabileceğini öğrencilere açıklar. Şapka bloklarından “eğildiğinde” bloğunu öğrencilere anlatır. Yukarı yönlü eğilme gerçekleştiğinde merkezde bulunan ışıkların, yukarıya doğru hareketi ve robot düz konuma geldiğinde tekrar merkeze doğru hareketini rehber öğretmen yaparak öğrencilere anlatır (Resim 6.12). Yapılan programı test eder. Öğrencilerden diğer yönler (aşağı, sağ ve sol) için programı tamamlamalarını ister.



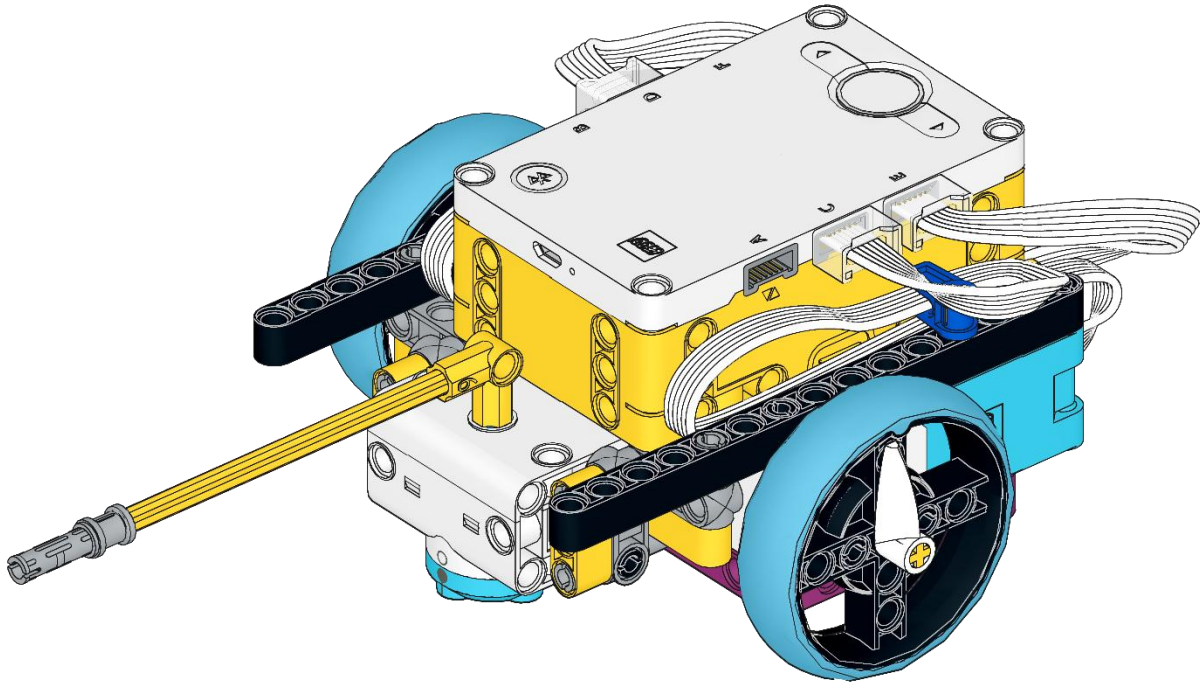
Resim 6.12 Su Terazisi Örnek Programı (Bir Bölümü)

TASARLA

Rehber öğretmen öğrencilerden Resim 6.13 ve 6.14'te görüldüğü gibi sürüş modeline hedef yön kolunu eklemelerini ister.



Resim 6.13 Hedef Yön Kolu Montajı Birinci Adım



Resim 6.14 Hedef Yön Kolu Montajı İkinci Adım

Tasarla etkinliği için rehber öğretmen öğrencilerden durmakta olan robotun hedef yön kolu çevrilerek hedefin yönü gösterildikten sonra sağ veya sol tuşa basılmasının ardından robotun yönünü hedefe doğru çeviren programı hazırlamalarını ister. Rehber öğretmen ekte sunulan **Video_6.1_Tasarla_Uret_Gorev.mp4** video dosyasını öğrencilere izleterek görevin daha kolay kavramasını sağlayabilir.

Ayrıca rehber öğretmen programın çalışması esnasında, hedef yön kolunun döndürülme miktarı, robotun kendi etrafında dönme miktarı gibi verilerin grafiklerinin oluşturulmasının programlama sürecindeki hataların giderilmesinde kolaylık sağlayacağını belirtir. Öğrencilerin bu süreçteki değişkenlerin (hedeflenen açı miktarı, büyük motorun dönme miktarı, robotun sapma açısı vb.) grafiklerini yorumlamalarını ister.

Tanımlama: Öğrenciler robotun istenilen görevi sorunsuz bir şekilde gerçekleştirebilmesi için gerekli görev adımlarını birer birer tanımlamalıdır. Robotun gerçekleştireceği işlemler sırası ile yazılmalıdır. Bu aşamada öğrenciler kalem kâğıt kullanarak tanımlama sürecini gerçekleştirmelidirler. Tanımlama süreci sonunda öğrenciler aşağıda bir örneği sunulmuş olan bir görev listesi hazırlamalıdır.

- Program çalıştırıldığında robotun konumu temel alınacak. Robot göreceli bir hareket gerçekleştirecek.
- Hedef kolu sağa veya sola el ile çevrilerek ucu bir hedefi gösterecek şekilde ayarlanacak.
- Robotun sağ veya sol tuşuna basılacak.
- Robot kendi etrafında dönerek yönünü hedefe doğru çevirecek.

Fikir üretme: Bu aşamada öğrenciler tanımlanan görevleri gerçekleştirecek algoritma adımları ve kullanılması gereken kod blokları hakkında fikirler üretirler. Tanımlama sürecinde belirlenen adımlar bu süreçte daha somut hale getirilir.

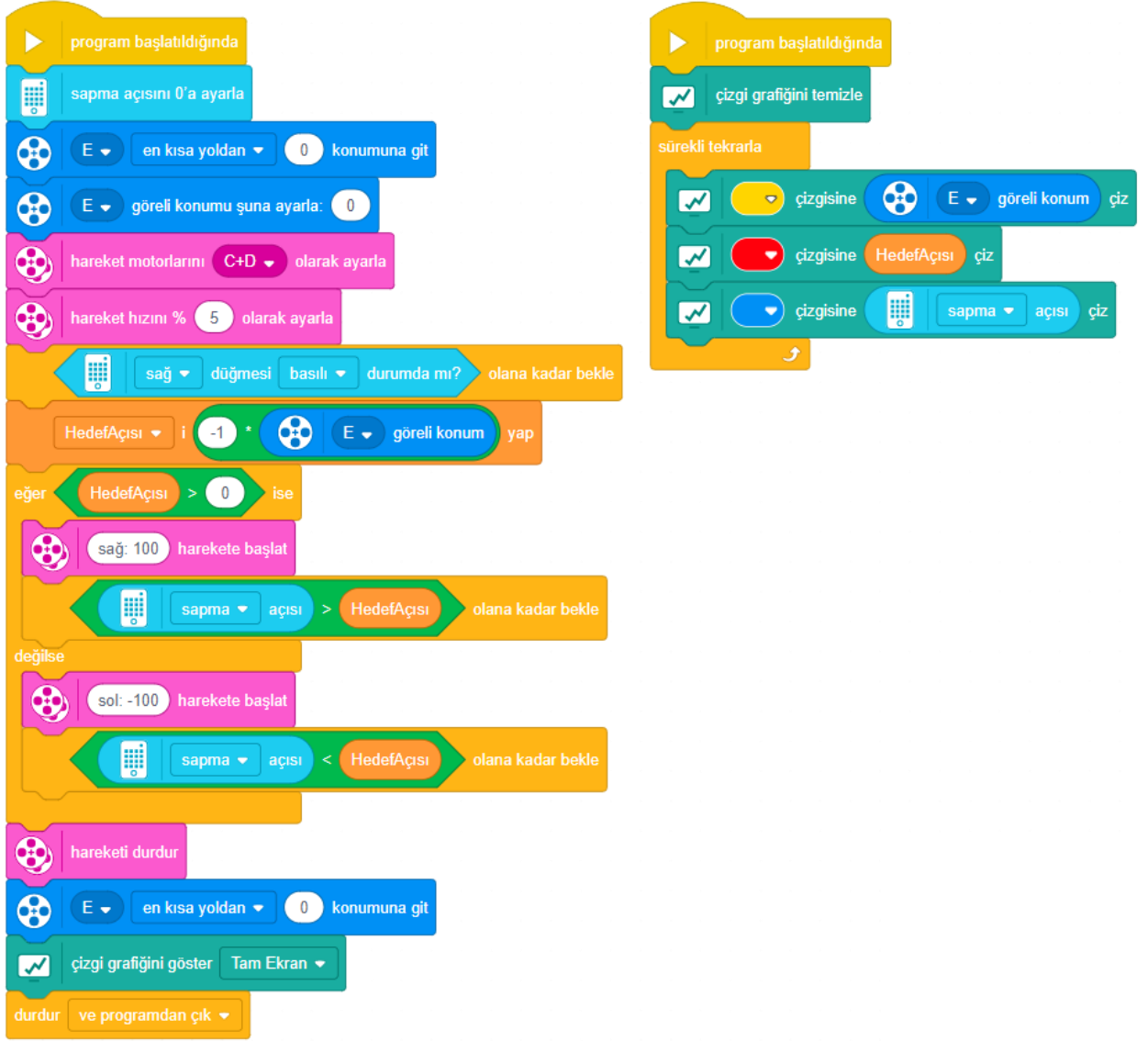
Örneğin:

- Hedef kolu büyük motorun dönme açısının sıfır olduğu konumunda olmalıdır.
- Büyük motorun göreceli konumu sıfır olmalıdır.
- Hedef kolunun sağa veya sola ne kadar çevrildiğini nasıl bulabiliriz.
- Robotun sağ veya sol tuşuna basılınca kadar beklemesini nasıl sağlayabiliriz.
- Robotun kendi etrafında dönme miktarı ile hedef kolunun dönme miktarının eşit olması gereklidir.
- Hedef kolu dönme açısı, robotun kendi etrafında dönme açısına nasıl eşitlenebilir.

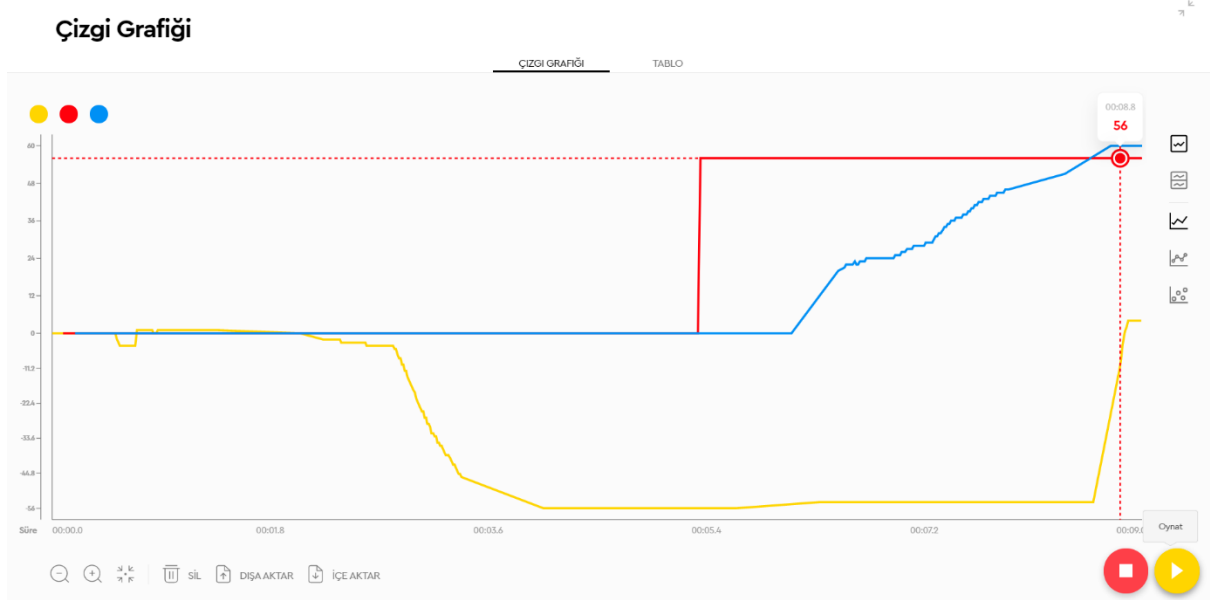
Öğrenciler bu süreçte farklı çözümler gerçekleştirebilirler. Tüm grupların aynı veya benzer sonuçlar üretmesi zorunlu değildir. Ayrıca, öğrencilerin farklı fikirleri test etmeleri rehber öğretmenler tarafından desteklenmelidir.

ÜRET

Öğrenciler bilgisayar ve robotlar ile çalışarak fikir üretme sürecinin çıktılarını program kodlarına dönüştürürler. Hazırladıkları programları robotlar ile test ederek kodlarındaki sorunları düzeltir ve kodlarını geliştirirler. Örnek program kodları ve programın çalıştırılması ile oluşan grafik Resim 6.15 ve Resim 6.16'da verilmiştir.



Resim 6.15 Hedefe Kilitlen Örnek Programı



Resim 6.16 Hedefe Kilitlen Değişkenleri Veri Grafiği

Rehber Öğretmen İçeriği – Hedefe Kilitlen Etkinliği Açıklaması

Bu etkinlikte hedef yön kolu ile belirlenen yön ve açı değeri kadar, robotun kendi etrafında dönmesi sağlanacaktır. Hedef yön kolunun program başladığı anda sıfır (0) konumuna alınması önemlidir. Böylece hedef yön kolunun ne kadar döndüğü daha kolay tespit edilebilir. Temel olarak hedef yön kolu ile robota kaç derece döneceği verisinin girişi yapılmaktadır.

Sürüş modeli tasarımına göre, büyük motorun (E girişine takılı) ters durmasından dolayı, hedef yön kolunun saat ibresi yönünde döndürülmesi büyük motorun görel konumunu negatif yönde artırırken, aynı yönde robotun döndürülmesi sapma açısını pozitif yönde artırmaktadır. Bu nedenle örnek programda büyük motorun görel konumu eksi bir (-1) ile çarpılmıştır. Bu durumun öğrencilere açıklanmasında fayda vardır. Hub bağlantı penceresi açılarak, bilgisayara bağlı robotun sapma açısı ve büyük motorun görel konumunun ters yönlü değişimi öğrencilere gösterilebilir.

Ayrıca öğrencilerin program sürecinde kullanılan değişkenlerin grafiğini yorumlayabilmeleri oldukça önemlidir. Örnek program için (sarı, büyük motor görel konum; kırmızı hedeflenen dönme miktarı; mavi, robotun gerçekte kendi etrafında dönme miktarı) değişkenlerin ne zaman ve neden grafikteki değerlere ulaştığı tartışılmalıdır.

DEĞERLENDİR

Rehber öğretmen Resim 6.17’de verilen programı her grubun robotlarına yükleyerek çalıştırmalarını ister. Program robot sallandığında rastgele bir sayı üretilip bu sayıyı ekranda 3 saniye göstermektedir. Grup sayısına göre 1- 10 aralığı değiştirilebilir.



Resim 6.17 Sallandığında Rastgele Sayı Üret Programı

Her bir gruba bir sayı değeri verilir. Herhangi bir grup robotlarını sallayarak, cevap verecek grubu seçer. Cevap veren grup robotunu sallayarak bir sonraki grubu seçer. Tüm sorular cevaplanıncaya kadar değerlendir etkinliğine bu şekilde devam edilir.

Rehber öğretmen gruplara aşağıdaki ve benzeri soruları yöneltir.

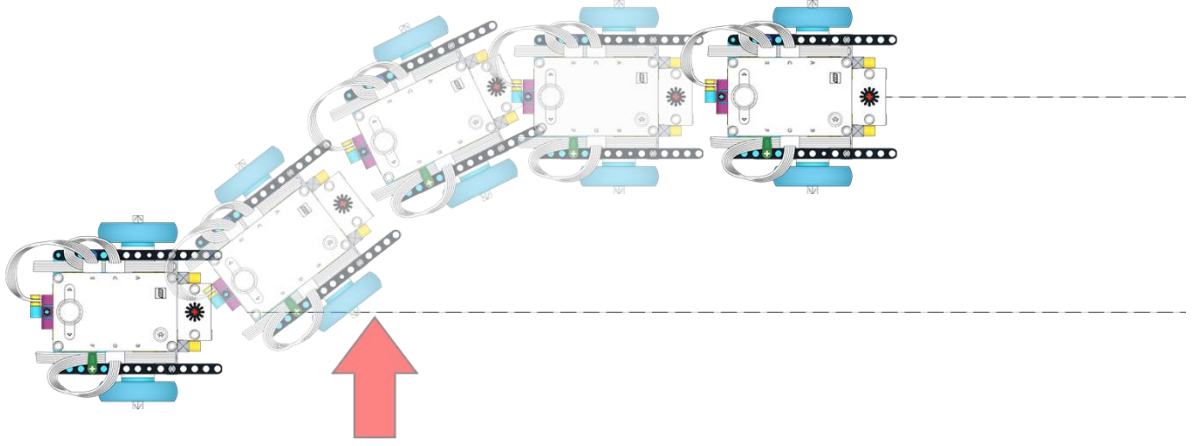
- Yunuslama hareketi nedir? Açıklayınız.
- Yuvarlanma hareketi nedir? Açıklayınız.
- Sapma hareketi nedir? Açıklayınız.
- Hub içerisinde bulunan hareket sensörü ne amaçla kullanılır?
- Hub içerisinde bulunan hareket sensörünün özellikleri nelerdir?
- Hareket sensörünün günlük hayattaki kullanımlarına örnekler veriniz.
- Bugün öğrendiğiniz kod blokları hangileridir? Görevleri nelerdir?
- Sizce bu haftanın en zor görevi hangisiydi? Zorluğun üstesinden nasıl geldiniz?

İLAVE ETKİNLİK

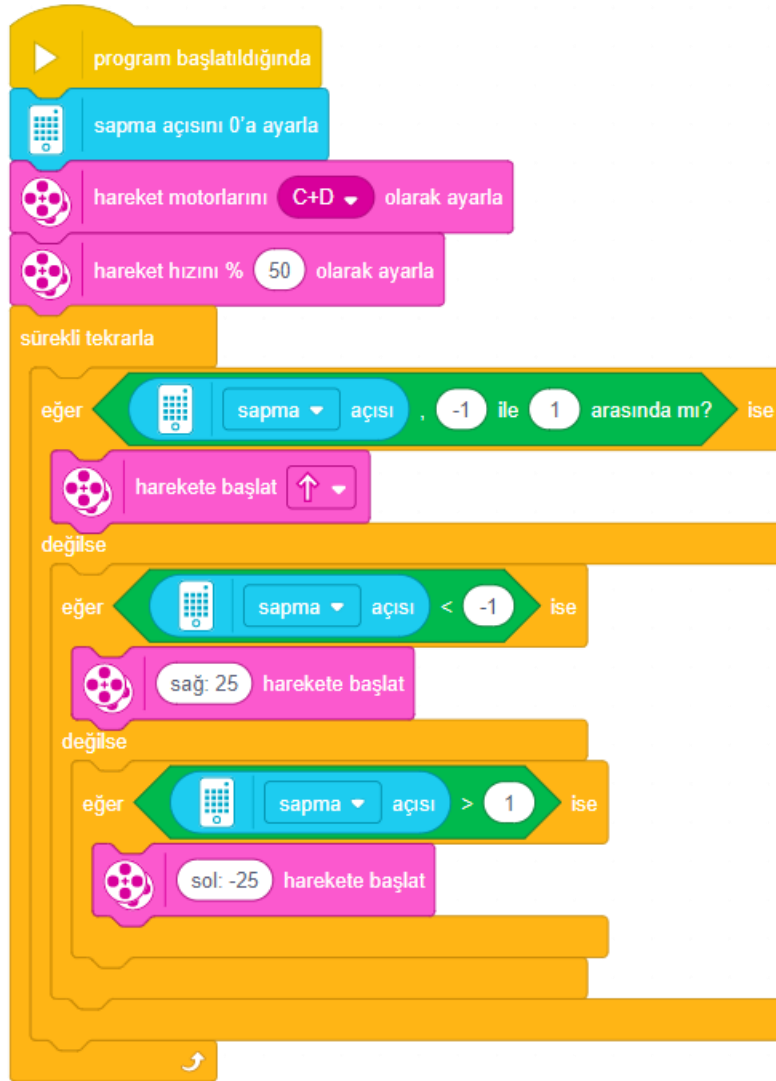
Yönünü Kaybetmeyen Robot

Bu etkinlikte öğrencilerden doğrusal olarak ilerleyen robotun dışarıdan yapılacak bir müdahale sonrasında tekrar önceki doğrultusuna dönmesini sağlayan bir robot tasarımları istenir.

Öğrencilerden robotlarını doğrusal ilerleyecek şekilde programlamaları istenir. Fakat doğrusal ilerlerken yönünü sağa veya sola döndürecek bir müdahale sonrasında ilerlediği doğrultuyu tekrar başlangıç doğrultusuna ayarlayabilecek bir robot olması gerektiği belirtilir. Örneğin robot başlangıçta kuzeye doğru doğrusal ilerlerken, rehber öğretmenin müdahalesi ile yönü doğuya çevrilen robotun tekrar yönünü kuzeye çevirmesi gerektiği anlatılır. Resim 6.18’de robotun beklenen hareketi gösterilmiştir. Resim 6.19’da örnek program sunulmuştur.



Resim 6.18 Yönünü Kaybetmeyen Robot



Resim 6.19 Yönünü Kaybetmeyen Robot Örnek Programı

7. Hafta- Renk Sensörü

Haftanın Amacı:

Bu haftanın amacı, öğrencilerin renk sensörünü kullanarak, renk okuma ve yansıyan ışık ölçme özelliklerini kullanabilmelerini sağlayacak temel bilgi ve becerileri kazandırmaktır. Öğrencilerin karşılaşılan renkleri okuyarak ve yansıyan ışığı kullanarak farklı programlama işlemlerini gerçekleştirebilmeleri amaçlanmıştır.

Haftanın Kazanımları:

- Öğrenciler renk sensörünün nasıl çalıştığını anlatabilirler.
- Öğrenciler renk algılama ve yansıyan ışığı ölçme özelliklerinin nasıl kullanıldığını gösterebilirler.
- Öğrenciler robotun sensörün önündeki renge göre işlem yapması için gerekli program bloklarını oluşturabilirler.
- Öğrenciler yansıyan ışığa göre robotun hareketlerini düzenlemeyi, yerdeki izleri takip edebilmesini veya renklerine göre nesneleri sayabilmesini sağlamak için gerekli programları oluşturabilirler.

Kullanılacak Malzemeler:

Robot seti, bilgisayar ve çalışma alanı çıktıları.

Ekler:

Bu hafta aşağıda sıralanan programlar ve çalışma alanı çizimlerini içeren dosyaların bilgileri ekte rehber öğretmenlere sunulmuştur.

Program_7.3_Karsilasilan_Rengin_Numarasini_Hub_Ekraninda_Gosterme.llsp3

Program_7.4_Calisma_Alanindan_Cikmayan_Robot.llsp3

Program_7.7_Yansiyan_Isigin_Degerini_Hub_Ekraninda_Gosterme.llsp3

Program_7.8_Eliptik_Bir_Yorungeyi_Takip_Eden_Robot.llsp3

Program_7.9_Ucuncu_Cizgiyi_Gecince_Duran_Robot.llsp3

Ek_7.7_Siyah_Beyaz_Gecis.pdf

Ek_7.9_Uc_Siyah_Cizgi.pdf

Mat_7_A3.pdf

Mat_7_A4.pdf

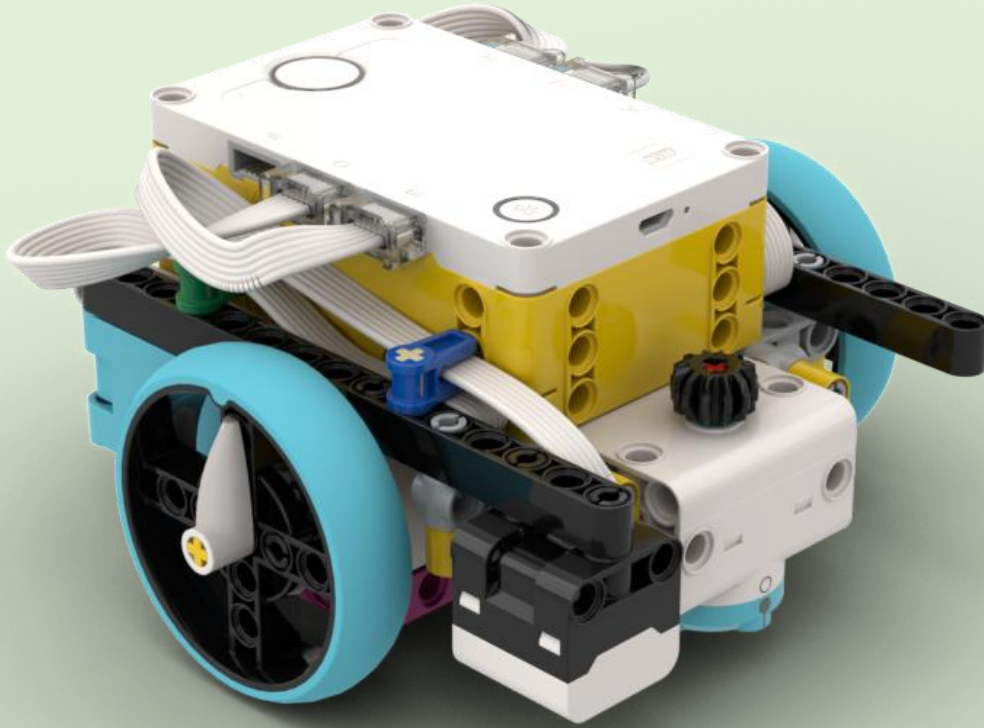
Renk Sensörü

Rehber Öğretmen İçeriği – Renk Sensörü

Renk sensörü rengi, yansımayı, ortam ışığını veya beyaz ışık emisyonunu algılayabilir. Sensör ayrıca ışık kaynağı olarak da kullanılabilir. Renk sensörü, optimum okuma mesafesi 16 mm (nesne boyutuna, rengine ve yüzeyine bağlı olarak değişebilir) olan renk okuma özelliği kullanıldığında 8 renk (siyah (0), mor (1), mavi (3), açık mavi (4), yeşil (5), sarı (7), kırmızı (9), beyaz (10)) ve renksiz (-1) değerlerini ölçebilmektedir.

Ayrıca, yine optimum okuma mesafesi 16 mm (nesne boyutuna, rengine ve yüzeyine bağlı olarak değişebilir) olan yansıtıcılık algılama özelliği sayesinde "0" (yansıtma) ile "100" (çok yansıtıcı) değerleri arasında ölçüm yapabilmektedir. Yansıtma beyaz ışık kullanılarak ölçülmektedir.

Sensör, Python modunda ortam ışığını veya beyaz ışık emisyonunu algılayabilir. Ortam ışığı yoğunluğu kullanıldığında renk sensörü herhangi bir ışık göndermez. Doğrudan bulunduğu ortamdan gelen ışığın yoğunluğu ile ilgilenir. Bu yoğunluk değeri yine "0" (en karanlık) ile "100" (en aydınlık) arasındadır.

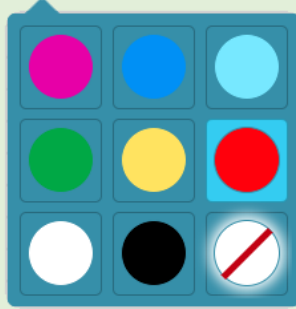


Resim 7.1 Renk Sensörü

Bu hafta etkinliklerde kullanılacak renk sensörü ile ilgili sözcük blokları ve açıklamaları aşağıda verilmiştir.

Sensörler Blok Paleti

Sensör blokları renk, mesafe, kuvvet ve jiroskop sensöründen bilgi alır.

Şu renkte mi?

- Siyah – 0
- Mor – 1
- Mavi – 3
- Açık Mavi – 4
- Yeşil – 5
- Sarı – 7
- Kırmızı – 9
- Beyaz – 10
- Renk yok – -1

Renk sensörü belirli renkleri algılayınca doğru sonucunu verir. Algılanabilecek renkler ve sayısal değerleri yukarıda verilmiştir.

Renk

- Siyah – 0
- Mor – 1
- Mavi – 3
- Açık Mavi – 4
- Yeşil – 5
- Sarı – 7
- Kırmızı – 9
- Beyaz – 10
- Renk yok – -1

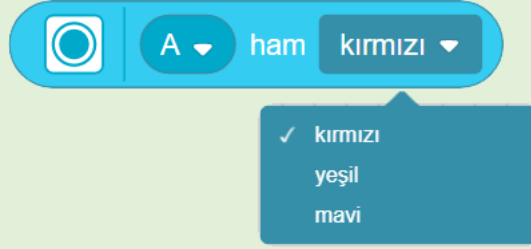
Renk sensörü tarafından tespit edilen renk değerini almak için kullanılır. Sensör tarafından tespit edilebilecek renkler yukarıda verilmiştir.

Yansıyan ışık şu mu?

Renk sensörüne geri yansıyan ışığın belirtilen yüzdeden küçük, büyük veya eşit olması durumunda doğru sonucunu verir.

Yansıyan ışık

Renk sensörüne geri yansıyan ışığın mevcut değerini verir.

Daha Fazla Sensör Blok Paleti**Ham renk**

Renk sensöründen ham kırmızı, yeşil ya da mavi renk okuması değerini verir. Renk değer aralığı 0-255 olarak ölçülebilir.

Kontrol Blok Paleti

Bekle yapıları, döngüler ve koşullar gibi blog yürütmenin doğal akışını değiştirebilecek blokları içerir.

Eğer öyle değilse şunu yap

Bu blok, belirtilen boole koşulunun doğruluğunu kontrol eder. Koşul doğru ise ilk boşluk içindeki bloklar oynatılır, varsa bu bloktan sonraki komut yığıtından devam eder. Komutun yanlış olması halinde ikinci boşluktaki komutlar oynatılır, varsa sonraki komut yığıtından devam eder.

GÖZLE VE UYGULA**Gözle ve Uygula: Karşılaşılan Rengin Numarasını Hub Ekranında Gösterme**

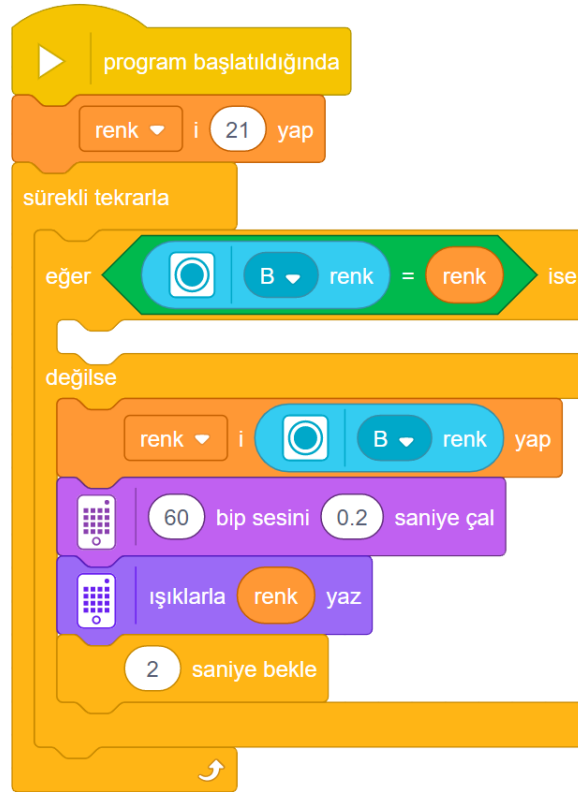
Bu uygulamada, sabit duran bir robotun renk sensörüne gösterilen farklı cisimlerin renklerinin numaralarını, sensöre gösterilen renk her değiştiğinde bip sesi çıkararak bir defa Hub ekranına yazmasını sağlayan bir program oluşturulacaktır. SPIKE yazılımı sensörü gördüğü renklerin sayısal kodlarını "Programlama Tuvali" ekranında göstermektedir. Aşağıdaki ekran görüntüsünde renk sensörünün önüne siyah, kırmızı ve yeşil renkli nesne getirildiğinde "Programlama Tuvali" ekranındaki renk sensörünün görüntüsü ayrı ayrı gösterilmiştir.



Resim 7.2 Renk Sensörünün Programlama Tuvalinde Görüntü

Programda renk sensörünün önündeki renk değiştirildiğinde Hub'ın ekranında sadece bir defa renk kodunun yazılması ve bip sesi çıkarılması için *renk* adlı bir değişken kullanılmıştır. İlk renkte programın çalışması için program başladığında *renk* değişkenine renk kodlarında olmayan bir değer atanmıştır. Renk sensörünün önüne her farklı renk geldiğinde programın bir defa çalışması için yeni okunan rengin değeri *renk* değişkenine atanmıştır. Bu sayede sensörün önündeki renk değişmediği sürece robotun bip sesi çıkarmaması ve önündeki rengin kodunu Hub ekranına yazmaya devam etmesi sağlanmıştır.

İstenilen işlemi yapmak için “eğer öyle değilse şunu yap” bloğu kullanılmıştır. Renk sensörünün okuduğu değer *renk* değişkenine eşit olduğunda *boole koşulu* doğru olduğu için herhangi bir işlem yapılmamıştır. Renk sensörünün okuduğu değer *renk* değişkeninden farklı olduğunda *boole koşulu* yanlış olur. *boole koşulu* yanlış olduğu için “eğer öyle değilse şunu yap” bloğundaki ikinci bölümde *renk* değişkeninin değeri renk sensörün okuduğu yeni değer yapılmış, Hub ekranına renk sensörünün okuduğu yeni değer yazılmış ve renk sensörünün yeni renk okumaması için 2 saniye durması için bekle komutu kullanılmıştır.



Resim 7.3 Karşılaşılan Rengin Numarasını Hub Ekranında Gösterme

Gözele: Yansıyan Işık Şiddeti

Rehber öğretmen öğrencilere renk sensörünün ışık şiddetini yansıyan ışık şiddeti ve ortam ışığı şiddeti olarak da ölçebildiğini belirtir. Yansıyan ışık şiddetini ölçmek için renk sensörünün karşısındaki yüzeye beyaz bir ışık gönderdiğini ve o yüzeyden yansıyan ışığın şiddetini "0" ile "100" arasında bir değer olarak ölçtüğünü anlatır. "0" en karanlık, "100" ise en aydınlık değeri ifade eder. Ölçülen değer sıfıra ne kadar yakınsa yüzeyden yansıyan ışık o kadar azdır, yani karanlıktır. Ölçülen değer 100'e ne kadar yakınsa yüzeyden yansıyan ışık o kadar fazladır, yani aydınlıktır.

Yansıyan ışık ile işlem yaparken "*yansıyan ışık*" ve "*yansıyan ışık şu mu*" blokları kullanılabilir. "*yansıyan ışık*" bloğu geri yansıyan ışığın değerini renk sensörüne bildirir. "*yansıyan ışık şu mu*" bloğu ise renk sensörüne geri yansıyan ışık belirtilen yüzdeden büyük, ona eşit veya daha az olduğunda "*doğru*" değerini döndürür.

Uygula: Çalışma Alanının İçinden Çıkmayan Robot

Rehber öğretmen öğrencilerden çalışma alanı içine bırakılan robotun çalışma alanının içinden çıkmadan hareket etmesini sağlayan programı hazırlamalarını ister. Öncelikle "*yansıyan ışık bu mu*" bloğunu kullanarak, beyaz ve siyah bölgelerin yansıyan ışık değer aralıklarının öğrenilmesi gerekir. Değer aralıklarının kullanılan kâğıdın özelliğine, sensörün kâğıda uzaklığına, ortamdaki ışığın miktarına ve robota/renk sensörüne göre değişiklik gösterebildiği öğrencilere hatırlatılır. Programın çalıştırılabilmesi için öncelikle öğrencilerin "*ışıklarla yaz*" bloğunu kullanarak siyah ve beyaz yüzeylerin yansıyan ışık aralıklarını belirlemeleri gerektiği söylenir.

Rehber öğretmen programın nasıl çalışması gerektiğini öğrencilerle birlikte belirlemeye çalışır. Daire içine bırakılan robotun daire dışına çıkmadan hareket edebilmesi için aşağıdaki listedeki adımlara benzer bir yapılacaklar listesi belirlenir;

- i. yansıyan ışığın rengi siyah olana kadar düz ilerlemesi,
- ii. siyah rengi algıladığında durması,
- iii. siyah rengin algılanması ve mevcut ivmesi ile durması sürecinde robotun siyah daire çizgilerinin dışına çıkmaması için hızının düşük olması,
- iv. belirli bir açı ile geri gitmesinin sağlanması ve
- v. ilk adımdan devam etmesi gerekir.

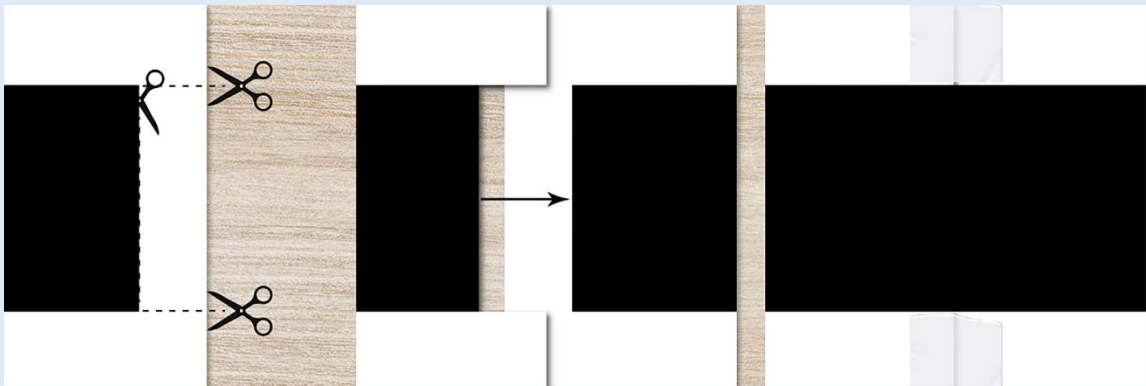
Rehber öğretmen, Resim 7.4'te görüldüğü gibi örnek bir programı öğrencilerle birlikte adım adım hazırlar.



Resim 7.4 Çalışma Alanının İçinden Çıkmayan Robot Programı

Not

Çalışma alanına, A3 ebatında çıktı imkânı varsa Mat_7_A3.pdf dosyasının çıktısı alınarak temin edilebilir. Çalışma alanı iki adet A4 ebatlarındaki çıktıların (Mat_7_A4.pdf) birleştirilmesi ile oluşturulduğunda, baskı esnasında oluşan kenar boşlukları birleştirilen kenarlardan biri makas ile kesilerek veya siyah kurşun kalemle boyanarak elips şeklindeki rotanın kesintisiz devam etmesi sağlanabilir. Ayrıca iki A4 kâğıdı yapıştırılarak etkinlik esnasında rotanın bozulması önlenir. Resim 7.5'te çalışma alanının kenarları kesilerek birleştirme, Resim 7.6'da çalışma alanının kurşun kalemle boyanarak birleştirme adımları gösterilmiştir.



Resim 7.5 Çalışma Alanı Kesilerek Birleştirme Adımları



Resim 7.6 Çalışma Alanı Boyanarak Birleştirme Adımları

Uygula: Yansıyan Işığın Değerini Hub Ekranında Gösterme

Rehber öğretmen, öğrencilerden robot “Ek_7.7_Siyah_Beyaz_Gecis.pdf” dosyasının çıktısı üzerinde hareket ederken yansıyan ışığın değerini Hub ekranında gösteren programı yazmalarını ister. Ayrıca robot beyaz alana ulaşınca, robotun durması gerekmektedir. Robot her iki yönde de gönderilerek yansıma miktarındaki değişiklik gözlenmelidir. Gözlem esnasında robotun hızının yavaş olması gerektiği öğrencilere hatırlatılmalıdır. Ayrıca robot beyaz rengi gördüğünde duracağı için robot göreve beyaz alandan başlatılmamalıdır.

Öğrencilerin Resim 7.7’de görülen programa benzer bir çözüm üretmeleri beklenmektedir. Rehber öğretmen öğrencilere takıldıkları noktalarda yardımcı olur, çözüm yolu gösterir. Etkinlik sonrasında rehber öğretmen öğrencilerle gözlem sonuçlarını tartışır.



Resim 7.7 Yansıyan Işığın Değerini Hub Ekranında Gösteren Örnek Program

Not

Derste kullanmak üzere ekte verilen "Ek_7.7_Siyah_Beyaz_Gecis.pdf" dosyasının çıktısı alınmalıdır. Dosyada beyazdan başlayarak koyu siyah tona kadar ilerledikçe artan siyah renkler bulunur.

TASARLA VE ÜRET

Öğrencilerden çalışma alanına çizilmiş elips şeklindeki yörüngeyi takip eden bir program hazırlamaları istenir. Robotu elipsin üzerinde bir turu en hızlı tamamlayan grubun yarışmayı kazanacağı, bu yüzden robotun bir turu daha hızlı tamamlaması için programları üzerinde düzenlemeler yapmaları gerektiği hatırlatılır. Öğrencilere çalışma alanındaki siyah çizgiyi takip etmek için robotun renk sensörünü kullanacağı hatırlatılır. Tasarlama sürecinde öğrencilerin eliptik bir yörüngeyi takip eden robot için aşağıda örnek olarak verilen iki adıma benzer bir süreci gerçekleştirmesi gerekir;

Tanımlama: Öğrencilerin öncelikle eliptik yörüngeyi takip edecek robotun yapması gereken işlemleri belirlemesi ve maddeler hâlinde yazması gerekir. Örneğin:

- Robot, renk sensörü siyah bölgedeyken (sensör siyah şerit üzerinde) sabit hızda hareket edecek,
- Renk sensörü beyaz bölgeye gelene kadar robot hareketine devam edecek,
- Renk sensörü beyaz bölgeye geldiğinde, renk sensörü siyah bölgeye gelene kadar robot sabit hızda hareket edecek ve
- Robot durdurulana kadar renk sensörü eliptik yörüngeyi üstünde dönmeye devam edecek.

Fikir üretme: Bu aşamada öğrencilerin tanımlamada belirlenen işlemlerin nasıl yapılabileceği ile ilgili muhtemel çözümler hakkında fikir yürütmesi gerekir. Örneğin, öğrenciler aşağıdaki maddelere benzer fikirler üretilebilir.

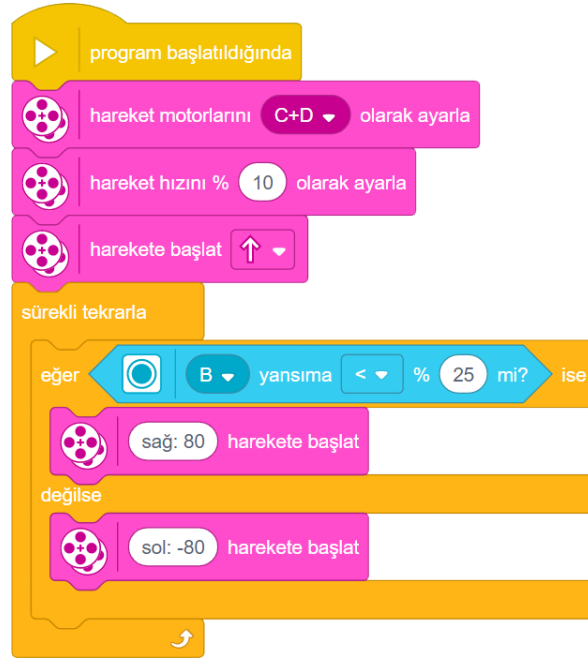
- Robotun renk sensörün beyaz çizgiye geldiğini algılayabilmesi için eşik değeri belirlenir (siyah bölgeden beyaz bölgeye geçtiğini anlamak için "yansıyan ışık" bloğu kullanılır).
- Robot ileri doğru küçük bir hızla hareket eder (beyaz rengi algıladığında renk sensörünün durma sürecinde elipsin dışına çıkmasını önlemek için).
- Renk sensörünün siyah çizgi üzerinden çıkma durumu sürekli olarak kontrol edilir ("yansıyan ışık şu mu" bloğu ile belirlenen eşik değeri kullanılarak beyaz bölgeye geçtiği an belirlenir. Renk sensörünün beyaz bölgeye geçme durumu "şu renkte mi?" bloğu kullanılarak da kontrol edilebilir.)
- Renk sensörü beyaz bölgeye geldiğinde siyah çizgi üzerine doğru ("direksiyonla harekete başla" bloğu kullanılarak) robot yönlendirilir. Buradaki hareketin renk sensörünün dairenin dışına çıkmasını engelleyecek kadar küçük olması gerekir.
- Bu adımlar robot durduruluncaya kadar tekrarlanır.

Bu programın amacı, renk sensörü beyaz bölgeye geldiğinde onu siyah bölgeye yönlendirerek elipsin üzerinde robotun zikzaklar çizerek hareket etmesini sağlamaktır. Daire içinden çıkmayan robot etkinliğinde olduğu gibi renk sensörünün beyaz bölgede olduğunu anlaması için "yansıyan

ışık" bloğu kullanılarak bir eşik değeri belirlenmelidir. Eşik değerini her grubun kendi programlarında bulup kullanmaları gerekir.

ÜRET

Tasarımlarını geliştirdikten sonra öğrenciler bilgisayar ve robot ile çalışma alanına çizilmiş elips şeklindeki yörünge üzerinde çalışarak istenilen program kodunu oluştururlar. Öğrencilerin aşağıdaki Resim 7.8’de görülen programa benzer bir çözüm üretmeleri beklenmektedir. Eliptik yörüngeyi en hızlı şekilde tamamlayan robotun grubunu belirlemek için grupların süreleri ölçülür ve kaydedilir.



Resim 7.8 Eliptik Yörüngeyi Takip Eden Robot Programı Örneği

Not

*Rehber öğretmenler Tasarla Üret bölümlerinde verilen kod örneklerini ve ekran görüntülerini öğrenciler ile **paylaşmamalıdır**. Tasarla Üret bölümündeki paylaşımlar rehber öğretmenlerin etkinliklere hazırlanmasında kolaylık sağlamak içindir. Öğrenciler Tasarla Üret bölümlerinde sorulan problemlere kendileri çözüm üretmelidir. Rehber öğretmenler öğrencilere problemleri çözebilmeleri için yeterince zaman tanımalı ve direk çözümü vermeden rehberlik sağlamalıdır.*

DEĞERLENDİR

Günün sonunda rehber öğretmen öğrencilere aşağıdaki soruları yönelterek bu hafta içeriğinin birlikte özetlenmesini sağlamalıdır.

- Robot setinde bulunan renk sensörünün özellikleri nelerdir?
- Günlük hayatta kullanılan renk sensörlerine örnekler verebilir misiniz? (Örneğin renk sensörleri nesneleri sıralama ve eşleştirme işlemlerinde kullanılabilir. Ayrıca, alışveriş

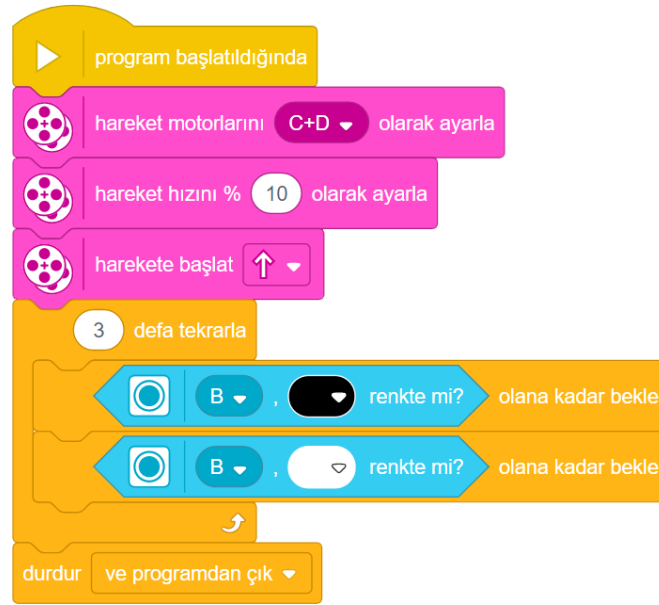
merkezlerinde çocukların bindiği yerdeki izleri takip ederek tur atan oyuncak araçlarda da kullanılırlar).

- Bugün öğrendiğiniz kod blokları hangileridir? Görevleri nelerdir?
- Eliptik yörüngeyi takip eden robot etkinliğinde robotun bir turu daha hızlı tamamlaması için programlarınızda ne tür değişiklikler yaptınız? Bu değişikliklerin etkileri neler oldu?
- Sizce bu hafta en zor görev hangisiydi? Zorluğun üstesinden nasıl geldiniz?

İLAVE ETKİNLİK

Üçüncü Çizgiyi Geçince Duran Robot

Bu etkinlikte öğrencilerden beyaz bir zeminde hareket ederken üç siyah çizgiyi (Ek_7.9_Uc_Siyah_Cizgi.pdf) geçtikten sonra duran robotun program kodlarını oluşturmaları istenir. Siyah çizgiler arasındaki mesafelerin farklı olduğu ve robotun üçüncü siyah çizgiyi geçtikten sonra hemen durması gerektiği öğrencilere bildirilir. Öğrencilerden “Tasarla” bölümünde bir örneği verilen “tanımlama” ve “fikir üretme” sürecini bu problem için de uygulamaları ve not almaları istenir. Örnek bir çözüm Resim 7.9’da verilmiştir.



Resim 7.9 Üçüncü Çizgiyi Geçince Duran Robot Programı Örneği

Not

Derste kullanmak üzere ekte verilen “Ek_7.9_Uc_Siyah_Cizgi.pdf” dosyasının çıktısı alınmalıdır. Dosyada araları ve genişlikleri farklı üç siyah şerit bulunur.

8. Hafta – Sensörlerin Birlikte Kullanımı

Haftanın Amacı:

Bu hafta etkinliklerde öğrencilerin birden fazla sensörü eş zamanlı kullanmalarını sağlayacak etkinliklerin yapılması amaçlanmıştır. Ayrıca bu süreçte öğrencilerin “Ve”, “Veya” ve “Değil” mantık operatörlerini kullanacakları etkinlikler planlanmıştır.

Haftanın Kazanımları:

- Öğrenciler farklı sensörleri bir arada kullanabilirler.
- Öğrenciler farklı sensörlerden alınan verileri kullanarak çözümler üretebilirler.
- Öğrenciler “Ve”, “Veya” ve “Değil” mantık operatörlerini kullanabilirler.
- Öğrenciler mesaj bloklarını kullanarak program akışını düzenleyebilirler.

Kullanılacak Malzemeler:

Robot seti, bilgisayar ve çalışma alanı (mat).

Ekler:

Aşağıda sıralanan örnek programlar .llsp3 (SPIKE uygulaması) dosya formatında rehber öğretmenler ile paylaşılmıştır. Ayrıca konu anlatımında bu programların ekran görüntülerine yer verilmiştir. Bu hafta ayrıca tasarla üret etkinliği için bir video ve çalışma alanı (mat) dosyaları ekte rehber öğretmenlere sunulmuştur. Rehber öğretmen A3 ebatında çıktı alma imkanına sahip ise her grup için bir tane Mat_8_A3.pdf dosyasının, değilse her grup için iki tane Mat_8_A4.pdf dosyasının çıktısını alarak etkinlik öncesinde hazırlamalıdır. Önceki hafta temin edilen çalışma alanları mevcutsa, bu hafta da etkin bir şekilde kullanılabilirler.

Program_8.4_Cizgi_Takip_Engel_Dur.llsp3
Program_8.8_Kirmizi_Renkte_mi.llsp3
Program_8.10_Kirmizi_Renkte_mi_Degil.llsp3
Program_8.11_Veya.llsp3
Program_8.14_Ve.llsp3
Program_8.15_Ve_Eger.llsp3
Program_8.23_Tasarla_Uret_Hedef_Tespit_Kilitlen.llsp3
Video_8.1_Tasarla_Uret_Gorev.mp4
Mat_8_A3.pdf (Mat_7_A3.pdf ile aynı)
Mat_8_A4.pdf (Mat_7_A4.pdf ile aynı)

Sensörlerin Birlikte Kullanımı

Rehber Öğretmen İçeriği – Sensörlerin Birlikte Kullanımı

Bu hafta öğrenciler genel olarak önceki haftalarda öğrendikleri sensörler ile ilgili programlar oluşturma yanı sıra, mantık işlemlerini kullanmayı öğreneceklerdir. Bu hafta etkinliklerde öğrenciler aşağıda açıklanan “Ve”, “Veya” ve “Değil” bloklarını kullanacaklardır.

Operatörler Blok Paleti

Ve



Bu blok, iki Boole bloğunu Ve (And) işlemi ile birleştirir. Her iki koşulunda doğru olması durumunda Ve işleminin sonucu doğru olur. Aksi takdirde sonuç yanlıştır.

Veya



Bu blok, iki Boole bloğunu Veya (Or) işlemi ile birleştirir. Koşullardan herhangi birinin doğru olması durumunda Veya işleminin sonucu doğru olur. Sonuç yalnızca her iki koşulun da yanlış olması durumunda yanlıştır.

Değil



Bu blok içerisinde yer alan Boole koşulunun tersini verir. Koşul yanlış ise sonuç doğru, doğru ise sonuç yanlıştır.

Not

Konu içerisinde sunulan ekran görüntüleri, örnek programların yansıtılarak öğrencilerle paylaşılması ve öğrencilerin bu programları kendi bilgisayarlarında kopyalamaları amacıyla **değildir**. Bu paylaşımlar, rehber öğretmenlerin hazırlık sürecinde kolaylık sağlamak için yapılmaktadır. Gözle etkinliklerinde rehber öğretmen, örnek programı adım adım gerçekleştirerek kullanılan program bloklarını ve programın çalışma mantığını öğrencilere açıklamalıdır. Üret etkinliklerinde ise öğrenciler, verilen probleme aktif olarak çözüm üretmeli ve programlarındaki hataları gidermelidir. Rehber öğretmen, bu süreçte yeterli düzeyde rehberlik sağlamalıdır.

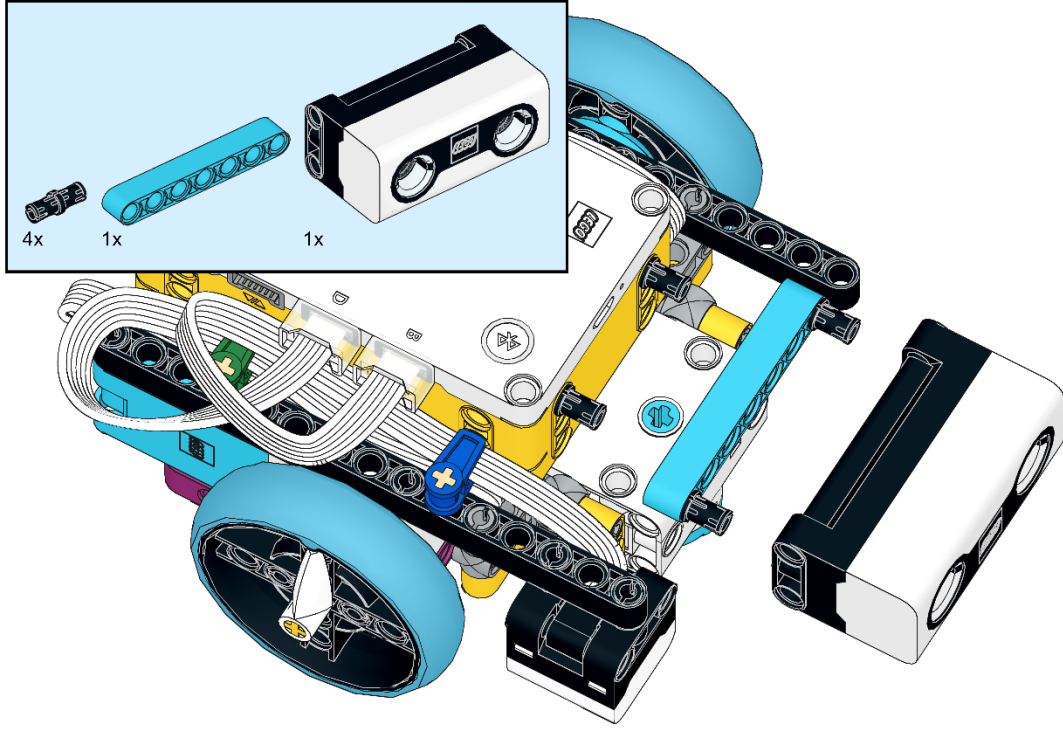
GÖZLE VE UYGULA

Gözle ve Uygula: Sensörlerin Birlikte Kullanımı

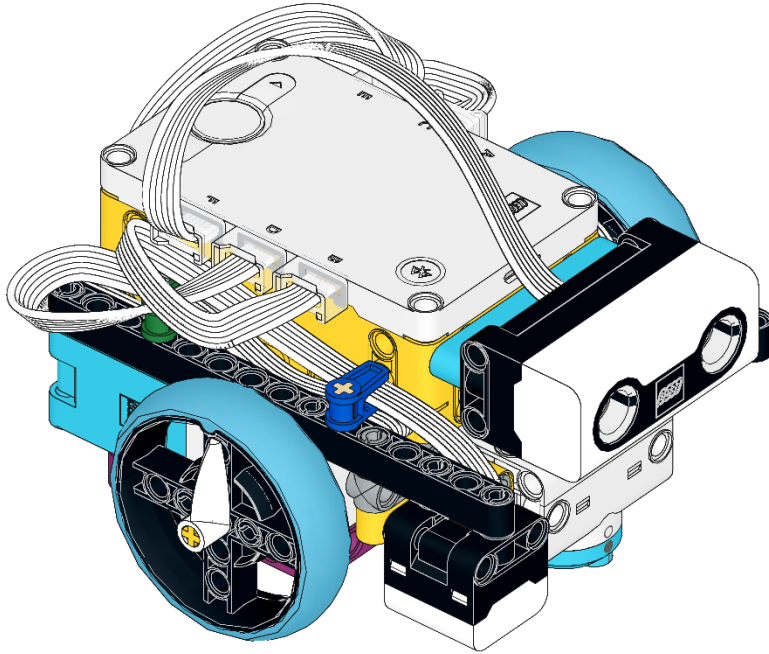
Rehber öğretmen, önceki haftalarda uygulamaları yapılan sensörler ile ilgili etkinlikleri hatırlatır. Her bir sensörün ne amaçla kullanıldığı ve ilgili sensörün özellikleri öğrencilerle birlikte tekrar edilir. Örneğin, kuvvet sensörünün 10 Newton'a kadar kuvveti ölçebildiği ve basıldı, bırakıldı ve sert basıldı durumlarını algılayabildiği, mesafe sensörünün 5 cm ile 200 cm arasındaki nesneleri algılayabildiği özetlenir. Renk ve hareket (gyro) sensörleri için de benzer özetlemeler yapılır.

Öğrencilerle birden fazla sensörün eş zamanlı olarak kullanıp kullanılamayacağı tartışılır. Farklı sensörlerin eş zamanlı kullanılabilecekleri görevlere örnekler vermeleri istenir. Örneğin, ileriye doğru hareket ederken, engele olan uzaklığı 20 cm'nin altında ise yavaşlayan ve engele dokununca duran robot veya siyah renkle belirlenmiş bir alandan dışarı çıkmayan ve alan içerisindeki bir engele de çarpmayan robot veya çizgi takibi gerçekleştirirken bir engel ile karşılaşınca duran robot.

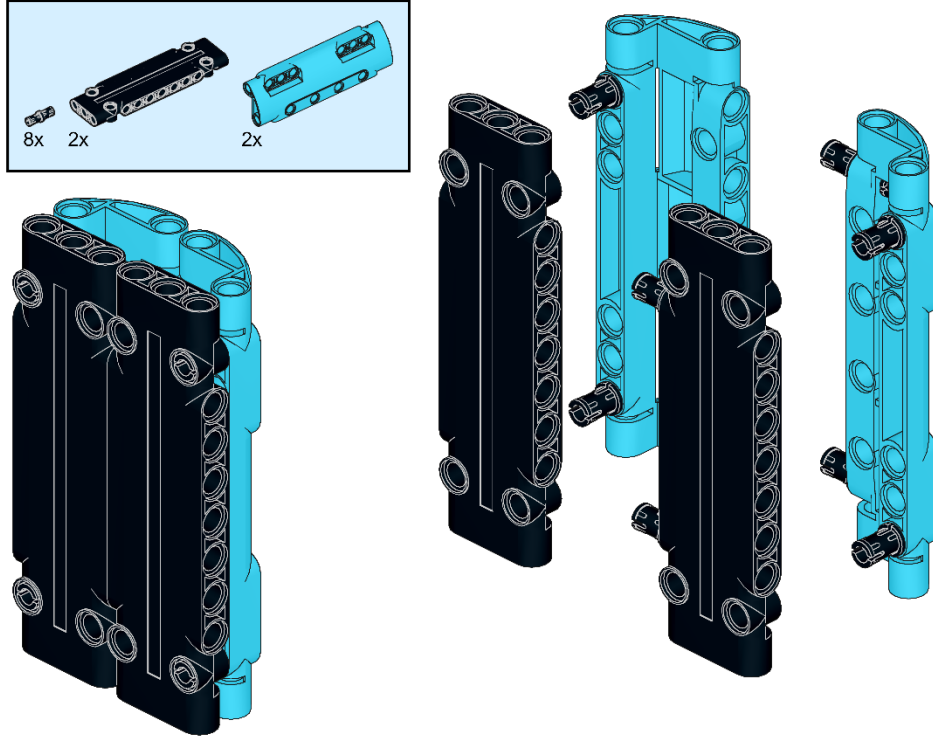
Rehber öğretmen, çizgi takibi gerçekleştirirken bir engel ile karşılaşınca duran robot programını hazırlayacaklarını belirtir. Bu etkinlik için robotun hangi sensörlere ihtiyacı olduğu tartışılır. Öğrencilerden renk sensörü takılı olan Sürüş Modeli tasarımına Resim 8.1 ve Resim 8.2'de gösterildiği gibi mesafe sensörünü kablosunun sıkışmaması için baş aşağı (Lego yazısı ters olacak şekilde) birleştirmeleri istenir. Sonrasında öğrenciler mesafe sensörü kablosunu Hub'ın F portuna bağlarlar. Öğrenciler ayrıca Resim 8.3'de gösterildiği gibi mesafe sensörünün algılayabileceği bir engel hazırlayabilirler ya da program çalışırken ellerini mesafe sensörünün önüne getirerek bir engel oluşturabilirler.



Resim 8.1 Mesafe Sensörünün Takılması Birinci Adım

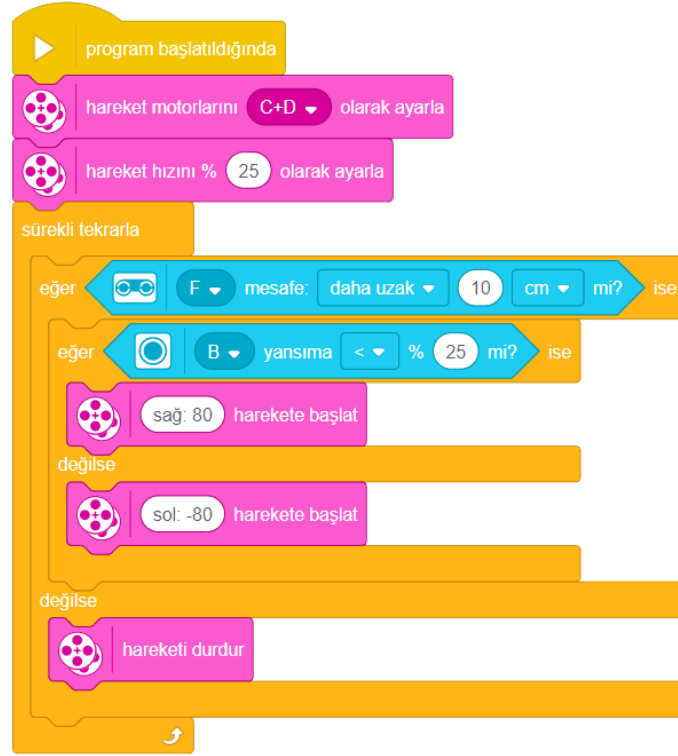


Resim 8.2 Mesafe Sensörünün Takılması İkinci Adım



Resim 8.3 Engelin Hazırlanması

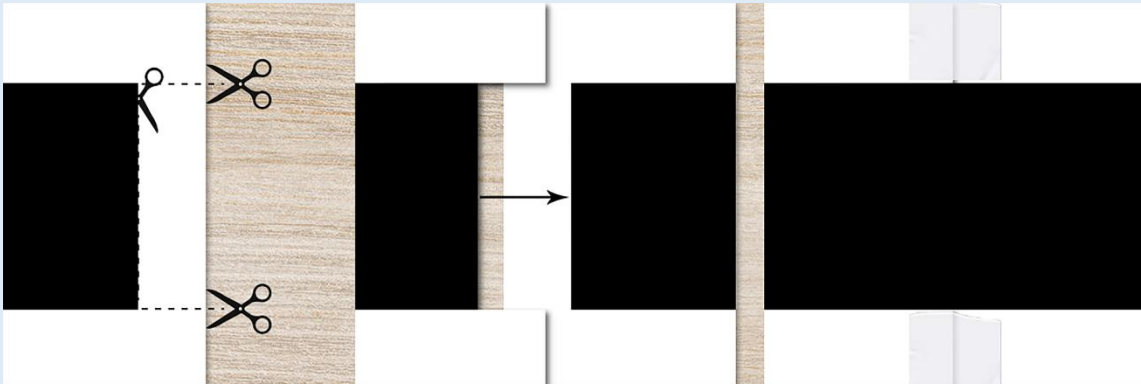
Rehber öğretmen, Resim 8.4'te görüldüğü gibi basit bir renk sensörü ile çizgi takibi programına mesafe sensörü koşulunu da ekleyerek, engel ile karşılaşılması durumunda robotun durmasını sağlayan programı adım adım hazırlar. Sonrasında rehber öğretmen, öğrencilerden benzer bir robot programını hazırlamalarını ve Resim 8.7'de görüldüğü gibi çalışma alanı üzerinde programlarını test etmelerini ister.



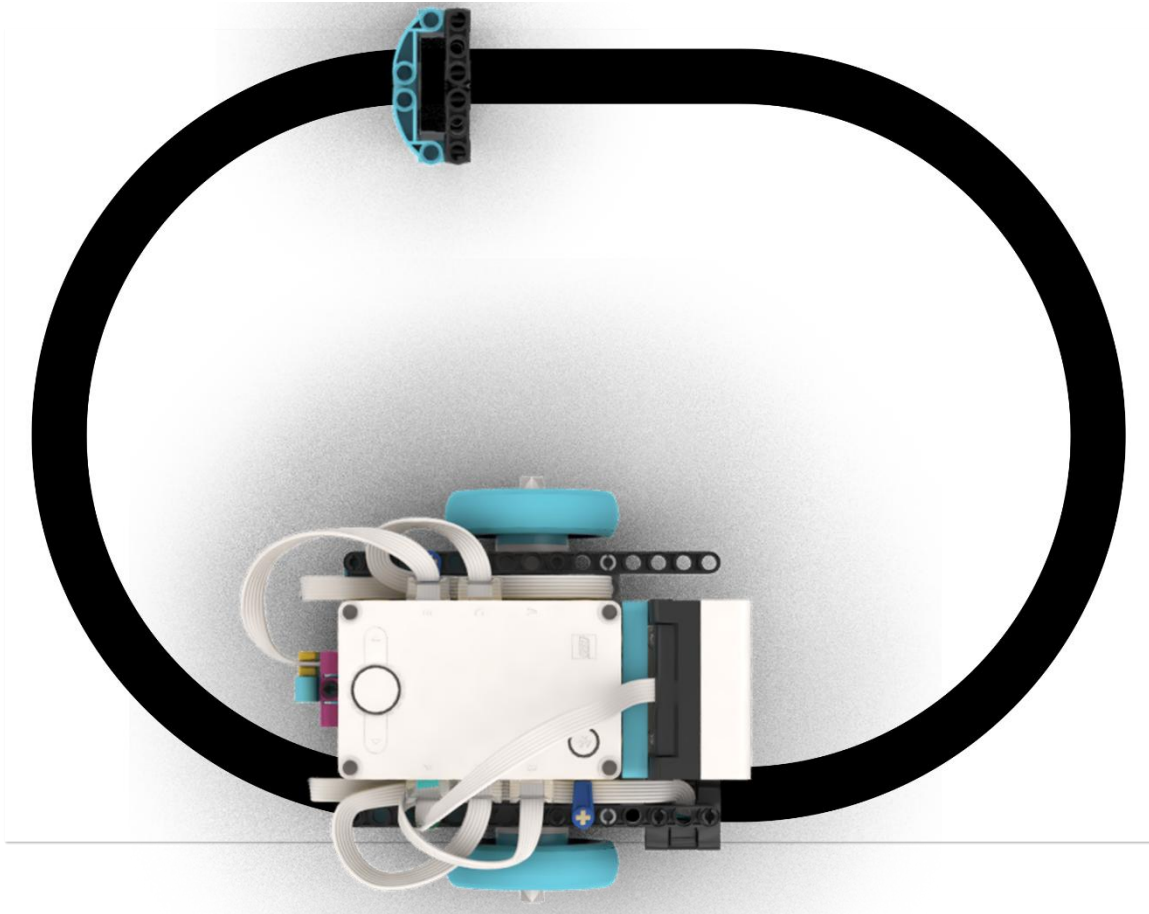
Resim 8.4 Çizgi Takibi ve Engelde Duran Robot Örnek Programı

Not

Çalışma alanını iki adet A4 ebatlarındaki çıktıların birleştirilmesi ile oluşturulduğunda, baskı esnasında oluşan kenar boşlukları birleştirilen kenarlardan biri makas ile kesilerek veya siyah kurşun kalemle boyanarak elips şeklindeki rotanın kesintisiz devam etmesi sağlanabilir. Ayrıca iki A4 kâğıdı yapıştırılarak etkinlik esnasında rotanın bozulması önlenir. Resim 8.5'te çalışma alanının kenarları kesilerek birleştirme, Resim 8.6'da çalışma alanının kurşun kalemle boyanarak birleştirme adımları gösterilmiştir.



Resim 8.5 Çalışma Alanı Kesilerek Birleştirme Adımları



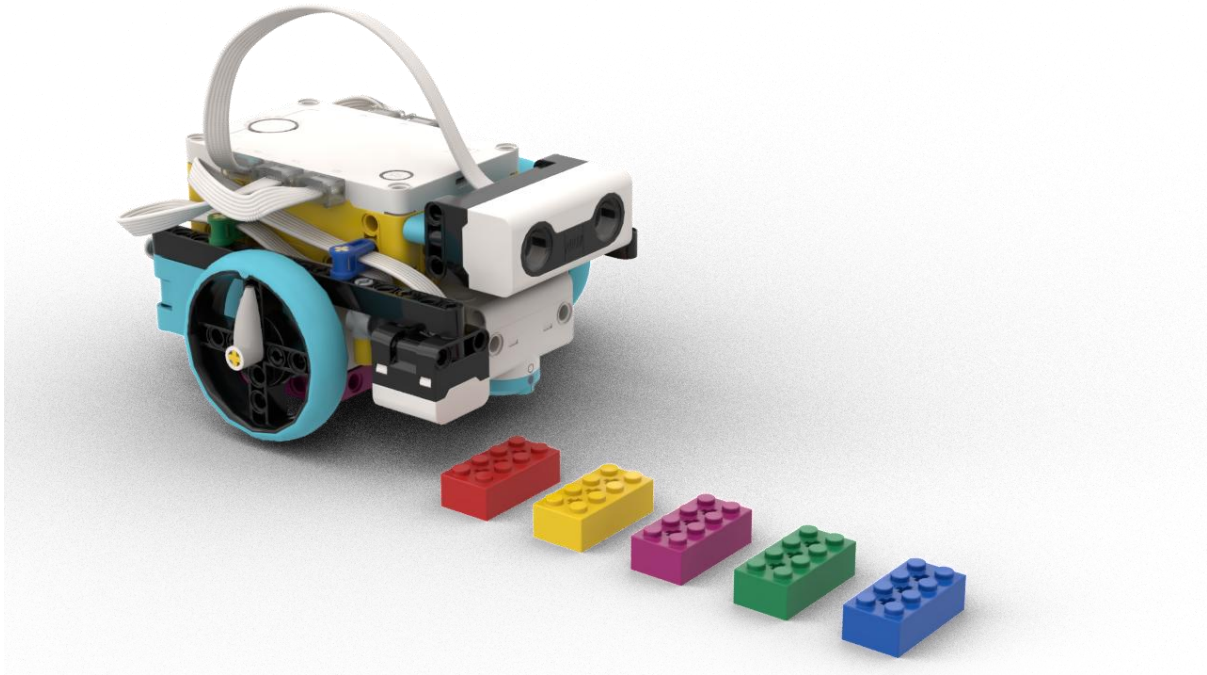
Gözle ve Uygula: Mantık Operatörleri Değil (Not)

Rehber öğretmen öğrencilere “Boole (Boolean) Blokları” hakkında hatırlatma yapar. Bu blokların altıgen şeklinde olduklarını ve yalnızca doğru veya yanlış (1 veya 0) değerlerini alabildiklerini ifade eder. Genellikle “C Blokları” (“eğer”, “olana kadar tekrarla” blokları vb.) içine yerleştirildiklerini belirtir. Örneğin bir “eğer” bloğu için koşulun ne olacağı “Boole Blokları” tarafından belirlenir. Koşulun doğru olması durumunda “eğer” bloğunun içerisinde yer alan bloklar çalıştırılır.

Rehber öğretmen öğrencilere “*değil*” bloğunun görevini anlatmadan Resim 8.8’de verilen programı hazırlayıp, Resim 8.9’da görüldüğü gibi farklı renklerdeki Lego parçaları ile robotun verdiği tepkiyi gözlemlenmelerini ister.

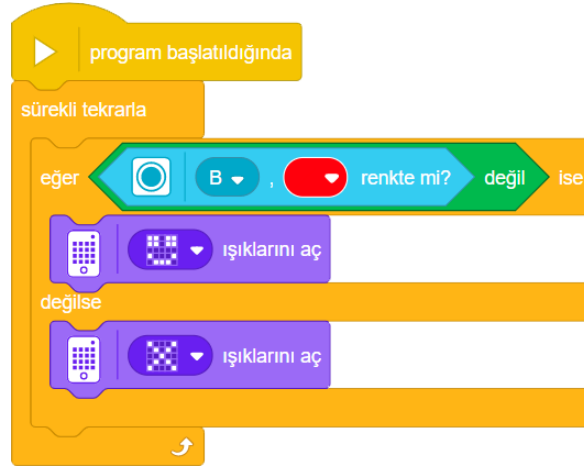


Resim 8.8 Kırmızı Renkte Mi Programı



Resim 8.9 Kırmızı Renkte Mi Uygulama

Sonrasında rehber öğretmen Resim 8.10’da gösterildiği gibi “*şu renkte mi*” bloğunu “*değil*” bloğu içerisine alarak aynı programı çalıştırmalarını ve farklı renklerdeki Lego blokları ile programı tekrar test etmelerini ister. Öğrencilerden iki programın sonucunu karşılaştırarak “*değil*” bloğunun görevini keşfetmeleri istenir. İhtiyaç hissedilmesi durumunda, öğrencilerin çıkarımları sonrasında rehber öğretmen, “*değil*” bloğunu ve işlevini özetler.



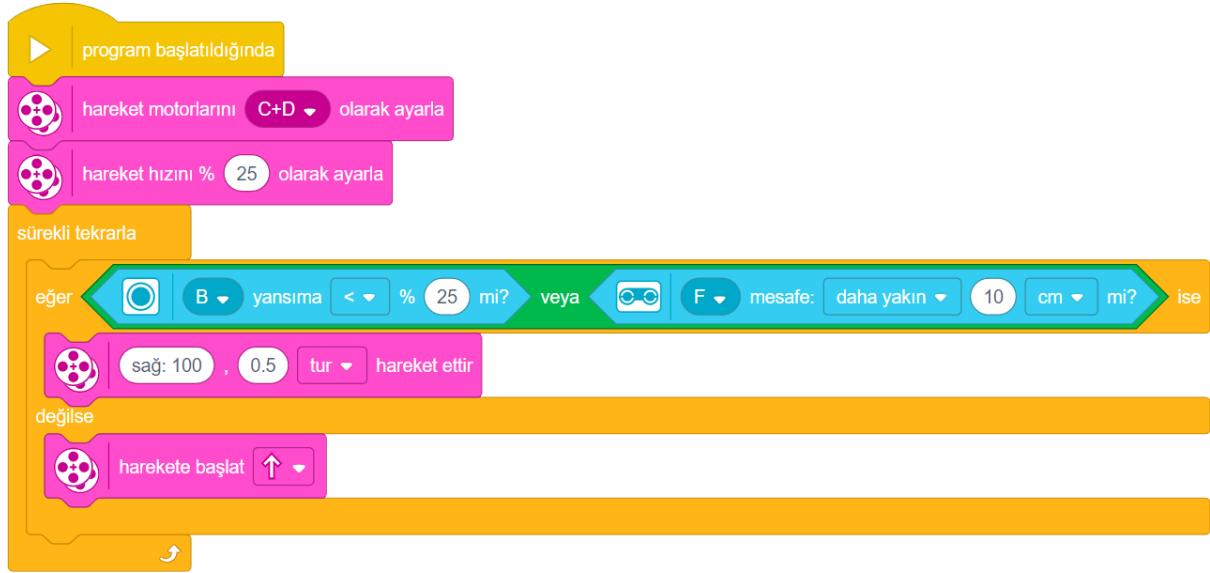
Resim 8.10 Kırmızı Renkte Değil Mi Programı

Gözle ve Uygula: Mantık Operatörleri Veya (Or)

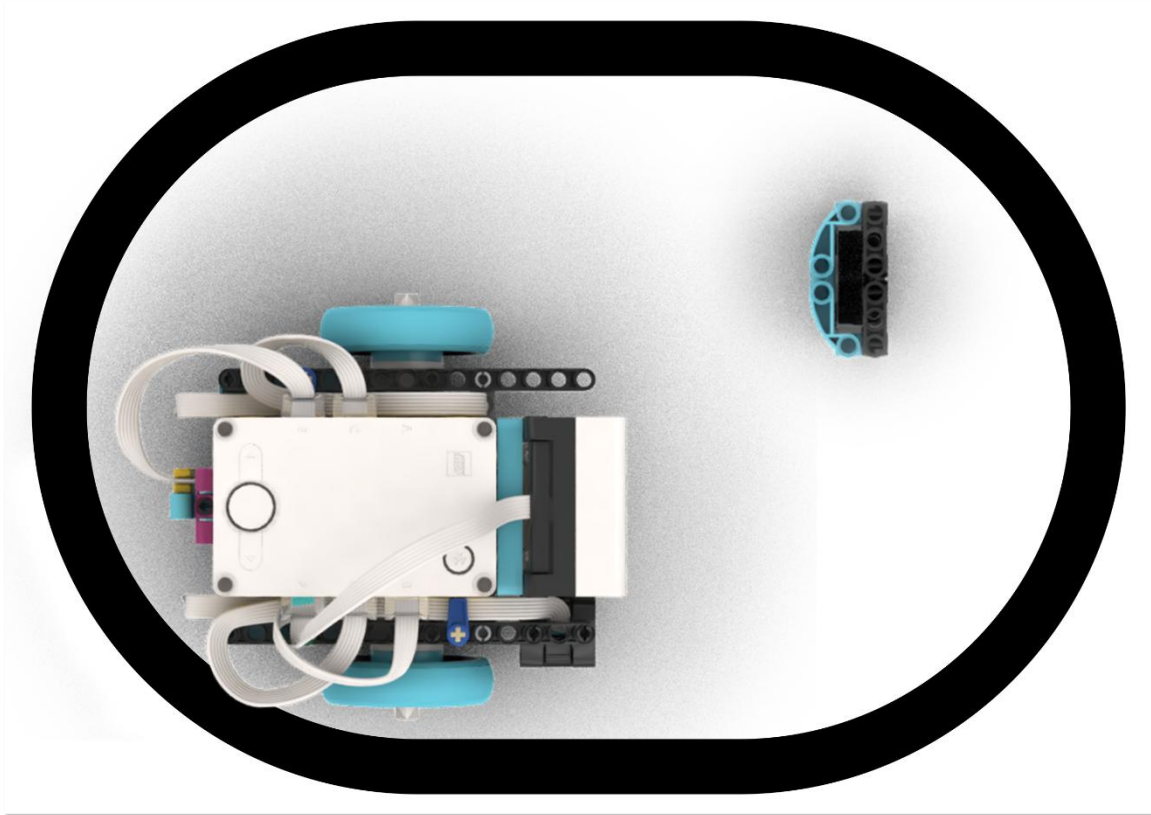
Rehber öğretmen, bir diğer operatör bloğunun “veya” bloğu olduğunu belirtir. “Veya” bloğunun iki farklı “Boole Bloğunu” birleştirdiğini ve herhangi birinin doğru olması durumunda sonucunun doğru, birleştirilen her iki bloğun yanlış olması durumunda sonucunun yanlış olduğunu belirtir. Rehber öğretmen örnek bir cümle ile veya operatörünü somutlaştırır. Örneğin, “Ali **veya** Veli robot yarışmasına katılacak.” cümlesinde, Ali’nin robot yarışmasına katılması durumunda, Veli’nin robot yarışmasına katılması durumunda ve Ali ile Veli’nin her ikisinin de robot yarışmasına katılması durumunda cümle doğrudur. Yalnızca Ali ile Veli’nin her ikisinin de robot yarışmasına katılmaması durumunda cümlenin anlamı yanlış olur.

Rehber öğretmen, öğrencilerden robot tasarımlarında herhangi bir değişiklik yapmadan, robotlarını siyah renkle belirlenmiş bir alandan dışarı çıkmayacak ve alan içerisindeki bir engele de çarpmayacak şekilde programlamalarını ister.

Rehber öğretmen ipucu olarak şu şekilde bir görev tanımlı yapabilir; “Robot siyah çizgiyi algıladığında **veya** engele 5 cm’den daha yakın olduğunda yönünü değiştirecek, aksi halde ileri doğru hareketine devam edecek.” Öğrencilerin Resim 8.11’de verilen programa benzer bir program hazırlamaları beklenmektedir. Sonrasında öğrenciler Resim 8.12’de gösterildiği gibi çalışma alanı üzerinde hazırladıkları programları test ederler.



Resim 8.11 "Veya" Bloğu Örnek Program



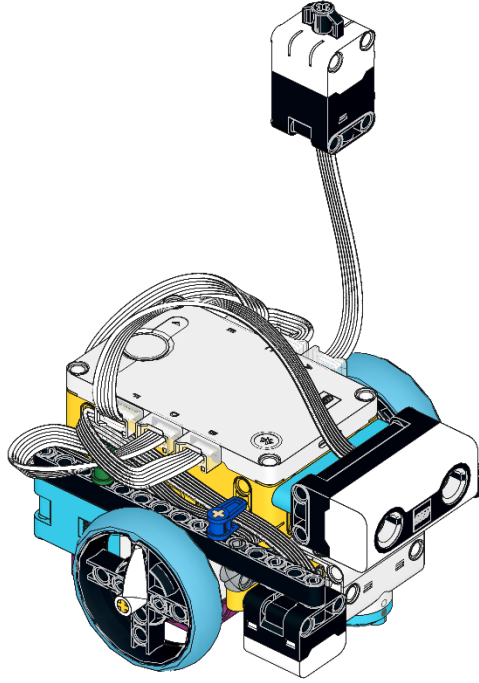
Resim 8.12 Siyah Alandan Çıkmayan ve Engele Çarpmayan Robot Çalışma Alanı

Gözle ve Uygula: Mantık Operatörleri Ve (And)

Rehber öğretmen "veya" bloğuna benzer şekilde "ve" bloğunu öğrencilere anlatır. "Ve" bloğunun iki farklı "Boole Bloğunu" birleştirdiğini ve yalnızca her iki koşulun da doğru olması durumunda

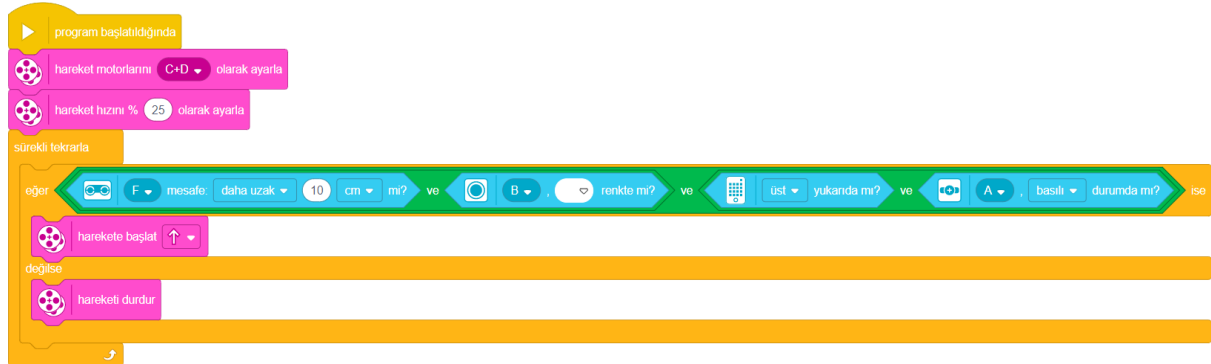
sonucunun doğru, diğer durumlarda sonucunun yanlış olduğunu belirtir. Veya operatöründe olduğu gibi örnek bir cümle ile ve operatörünü somutlaştırır. “Ali **ve** Veli robot yarışmasına katılacak.” cümlesinde, yalnızca Ali’nin robot yarışmasına katılması durumunda, yalnızca Veli’nin robot yarışmasına katılması durumunda ve Ali ile Veli’nin her ikisinin de robot yarışmasına katılmaması durumunda cümle yanlış olacaktır. Yalnızca Ali ile Veli’nin her ikisinin de robot yarışmasına katılması durumunda cümle anlam bakımından doğru bir cümle olur.

Rehber öğretmen, öğrencilerden robotlarına Resim 8.13’te gösterildiği gibi kuvvet sensörünü Hub’ın A portuna takmalarını ister.



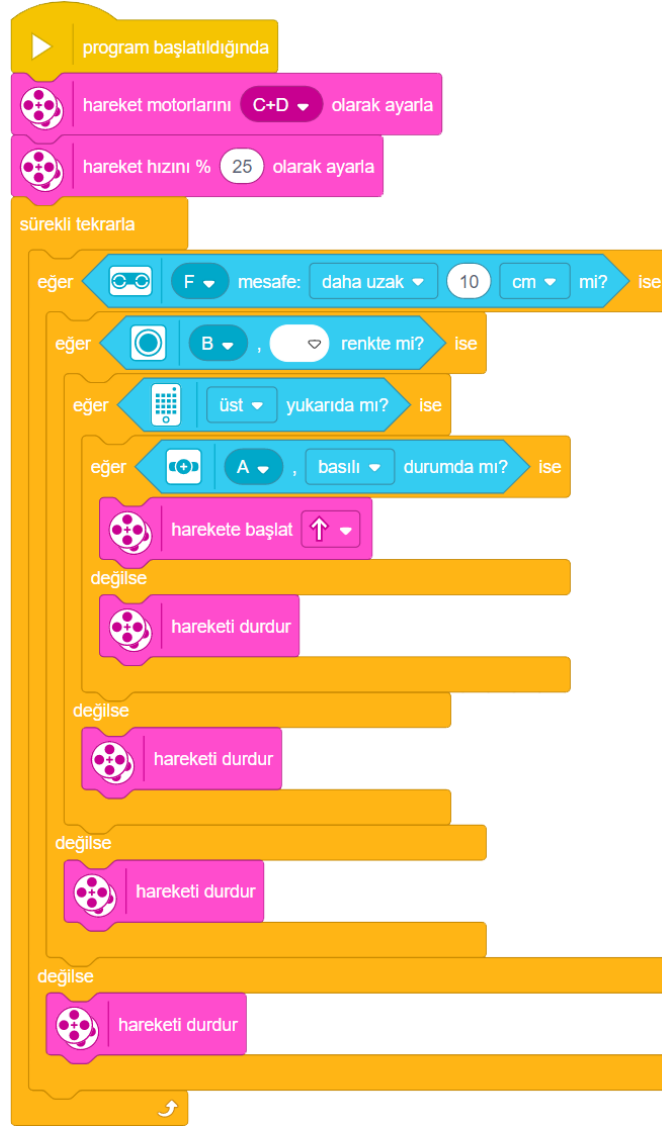
Resim 8.13 Kuvvet Sensörünün Takılması

Rehber öğretmen öğrencilerden robotun yalnızca karşısında 10 cm mesafede içerisinde bir engel olmadığında **ve** robot beyaz renkte bir zemindeyken **ve** robot sağa veya sola devrilmemişse (Hub’ın üst yüzü yukarıda ise) **ve** kuvvet sensörüne basılı ise ileri doğru hareket etmesini sağlayacak programı hazırlamalarını ister. Bu koşullardan herhangi biri gerçekleşmemiş ise robotun hareket etmemesi gerekmektedir. Bu koşullarda hareket eden robotun örnek programı Resim 8.14’te sunulmuştur.



Resim 8.14 “Ve” Bloğu Örnek Program

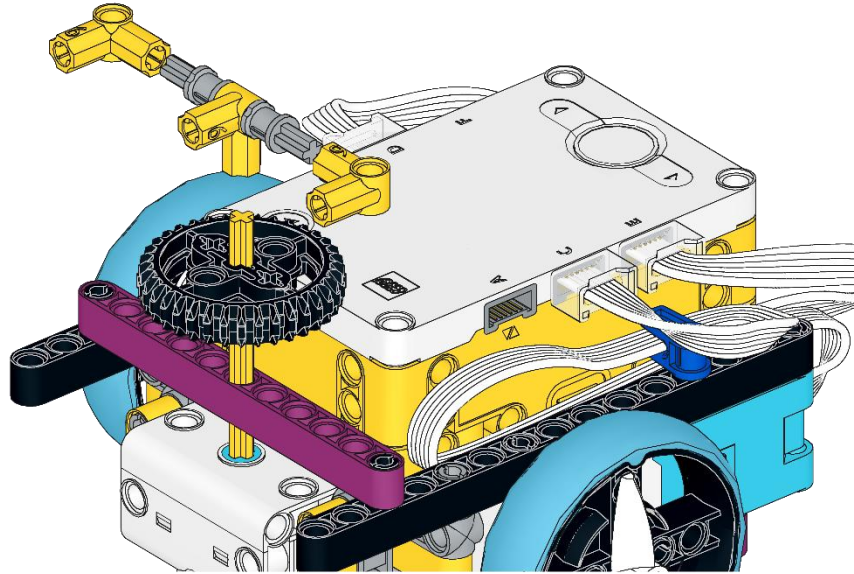
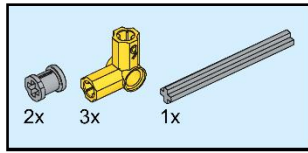
Rehber öğretmen aynı görevin “ve” bloğu kullanılmadan da yapılabileceğini ifade eder. Öğrencilerden “ve” bloğu kullanmadan, bu dört koşulun sağlanması durumunda hareket eden, koşullardan herhangi biri sağlanmadığı zaman hareket etmeyen robot programını hazırlamalarını ister. Öğrencilerin çözüm üretmelerine olanak tanınmalı, gerekli görülmesi durumunda ipucu olarak iç içe “eğer öyle değilse şunu yap” bloklarının kullanılması gerektiği belirtilebilir. Örnek program Resim 8.15’te sunulmuştur. Etkinlik sonrasında rehber öğretmen öğrencilerden bu iki programı kıyaslamalarını ister.



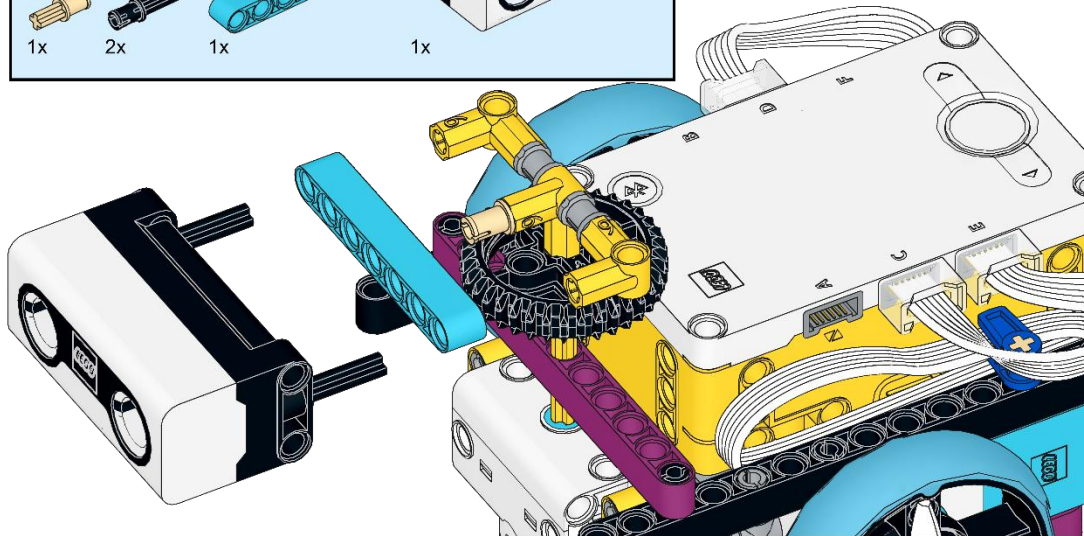
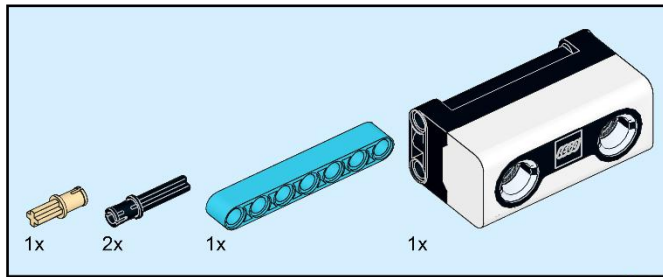
Resim 8.15 “Ve” Bloğu Yerine “Eğer” Bloğu Örnek Program

TASARLA

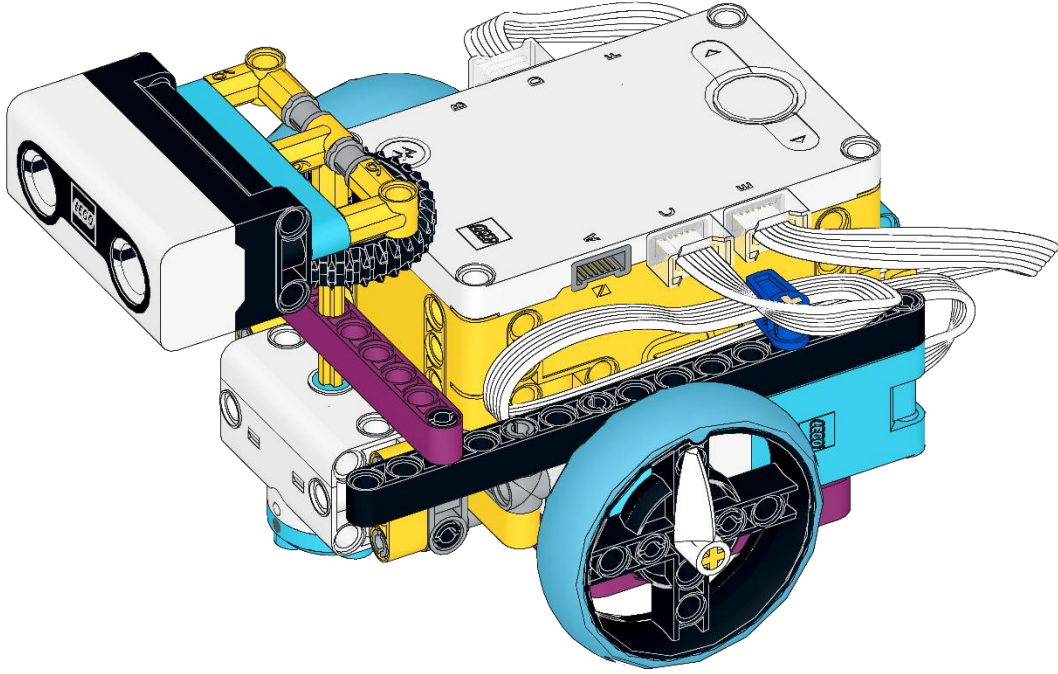
Tasarla etkinliğinde rehber öğretmen öğrencilere 6. haftada yaptıkları hedefe kilitlenen robot etkinliğini hatırlatır. O etkinlikte, robotun hareket sensörünü kullanarak hedef yön kolu ile gösterilen hedefe yöneldiği belirtilir. Rehber öğretmen, bu etkinlikte de robotun hedefe yöneleceğini fakat o etkinlikten farklı olarak, hedefi kendisinin tespit edeceğini ifade eder.



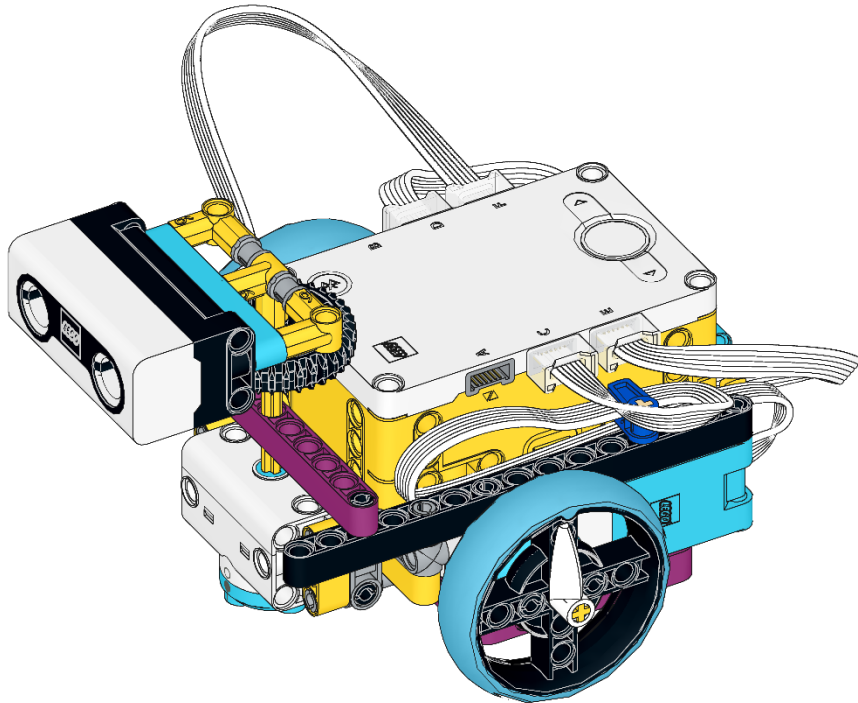
Resim 8.18 Hedef Tespit Robotu Üçüncü Adım



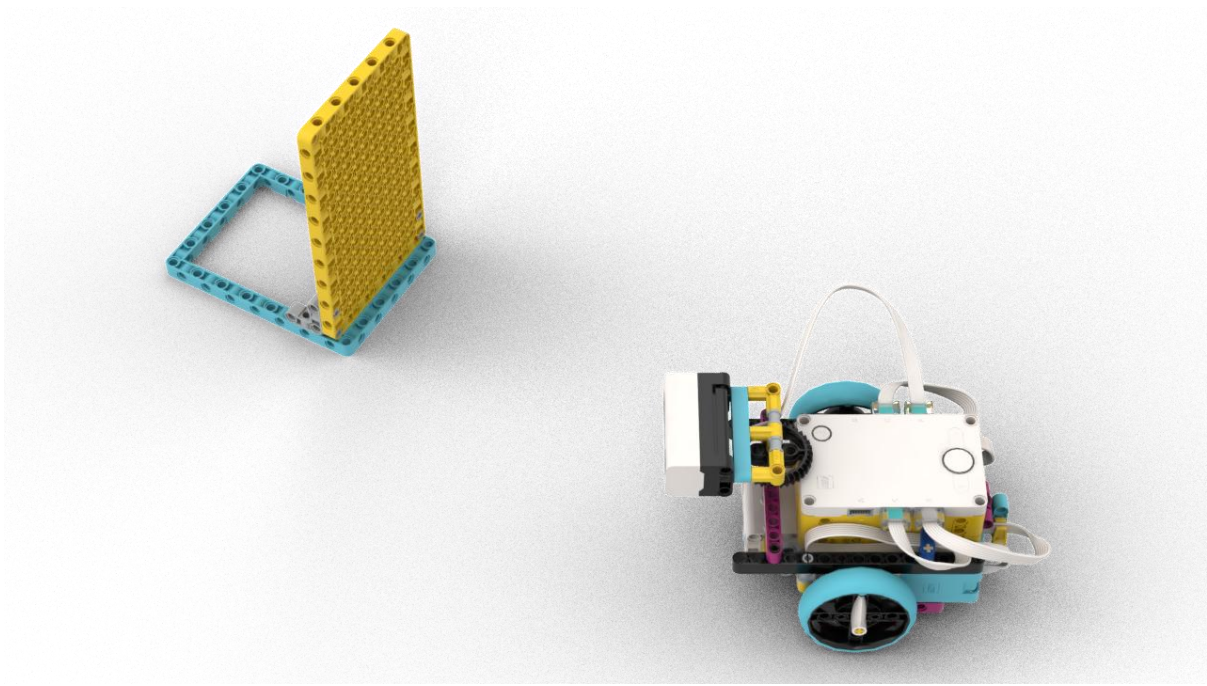
Resim 8.19 Hedef Tespit Robotu Dördüncü Adım



Resim 8.20 Hedef Tespit Robotu Beşinci Adım



Resim 8.21 Hedef Tespit Robotu Altıncı Adım



Resim 8.22 Hedef Tespit Robotu Görev Alanı

Öğrenciler mesafe sensörünü robotlarına taktıktan sonra, rehber öğretmen ayrıca Resim 8.22’de olduğu gibi Lego parçaları ile bir hedef hazırlamalarını ister. Robotun görevi mesafe sensörü ile karşısındaki 180 derecelik alanı 10 derecelik açılar ile dönerek taramaktır. Robot, eğer 50 cm’den daha yakın mesafede herhangi bir hedef tespit edemez ise 1 tur ileri giderek tekrar tarama işlemi yapmalıdır. Robot 50 cm’den daha yakın bir mesafede hedef tespit ettiğinde, robotun hedefe doğru hareket ederek hedefe yaklaşmalı ve hedefe 10 cm uzaklıkta durmalıdır. Rehber öğretmen görevin daha kolay anlaşılması için ekte sunulan **Video_8_Tasarla_Uret.mp4** video dosyasını öğrenciler ile paylaşmalıdır.

Tanımlama: Öğrenciler bilgisayar ve robotları ile çalışmadan önce kendilerine verilen problemi kalem kağıt kullanarak tanımlamalıdır. Robotun istenilen görevleri başarı ile gerçekleştirebilmesi için yapması gerekenler ve alması gereken kararlar sıralanmalıdır. Bu süreçte sonunda öğrenciler aşağıdakine benzer bir görev listesi oluşturulmalıdır.

- Mesafe sensörü sürekli 180 derecelik bir alanı taramalıdır.
- Tarama esnasında mesafe sensörü 10 derecelik açılarla dönmelidir.
- Herhangi bir hedef tespit edilemediğinde, robot 1 tur ileri hareket etmelidir.
- Mesafe sensörü bir hedef algıladığında, robot yönünü hedefe çevirmelidir.
- Mesafe sensörünün hedefi algıladığı andaki yönü, robota aktarılmalıdır.
- Robotun belirlenen açı değeri kadar dönmesi sağlanmalıdır.
- Robot hedefe doğru hareket etmelidir.
- Robot hedeften 10 cm uzaklıkta durmalıdır.

Fikir üretme: Bu süreçte öğrenciler, tanımlama süreci sonunda ortaya çıkan görevleri robotun gerçekleştirebilmesi için gerekli algoritma ve kullanılacak program bloklarını hakkında fikirler üretirler.

Örneğin:

- Mesafe sensörü sürekli 180 derecelik bir alanı tarayacağından sonsuz döngü kullanılmalıdır.
- Mesafe sensörü büyük motora direk bağlı olduğundan motorun dönme açısı mesafe sensörünün hedefi algıladığı açı olacaktır.
- Büyük motorun dönme açısı 10 derece artarak 0 ile 180 (veya -90 ile 90) aralığında olmalıdır. Bu nedenle “olana kadar bekle” veya “olana kadar tekrarla” blokları kullanılabilir.
- Robotun kendi etrafında, büyük motorun dönme açısı kadar dönmesi, hareket sensörü sapma açısı kullanılarak yapılabilir (6. haftada benzer bir uygulama yapılmıştı).
- Robotun hedefe 10 cm mesafede durması için yine mesafe sensörü kullanılmalıdır.

Öğrenciler bu süreçte farklı çözümler gerçekleştirebilirler. Örneğin mesafe sensörü 180 derecelik bir alanı tararken, bazı gruplar 0'dan onar onar artarak 180'e ulaşınca tekrar 0 dereceye dönüp aynı işlemi tekrar ederken, bazı gruplar 180'den onar onar azalarak 0'a ulaşmayı veya -90 ile +90 aralığında programlarını hazırlamayı seçebilir. Tüm gruplar aynı veya benzer programı hazırlamak zorunlu değildir. Ayrıca, öğrencilerin farklı fikirleri test etmeleri rehber öğretmenler tarafında desteklenmelidir.

ÜRET

Öğrenciler problemi tanımlayıp, çözüm için fikirler ürettikten sonra bilgisayar ve robotlarla çalışarak fikirlerini test etmelidirler. İstenilen görevi gerçekleştiren programı bu sürecin sonunda üretmelidirler. Bu etkinlik için örnek program kodları Resim 8.23'te sunulmuştur.

Rehber Öğretmen İçeriği – Mesaj Blokları

Resim 8.23'te sunulan programda “mesaj yayınla” (haberini sal) ve “mesaj aldığında” (haberini aldığında) blokları kullanılmıştır. Öğrenciler kendi programlarını hazırlarken, aynı programı hazırlamak zorunda olmamakla birlikte, rehber öğretmenin yeni blokların kullanılmasını teşvik edebilir. Tasarla ve Üret etkinliği öncesinde bu blokların öğrencilere anlatması önerilmektedir.

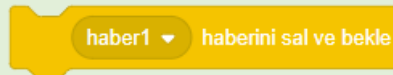
Olay Blok Paleti

Mesaj Yayınla



Bu blok belirli bir haber yayınlar. Haber yayınlandıktan sonra, tüm “mesaj aldığında” şapka blokları çalıştırılır. Program akışı mesaj gönderildikten sonra bir sonraki blok ile devam eder. Böylece program akışında istenilen noktadan itibaren diğer yığıt veya yığıtların çalıştırılması sağlanır.

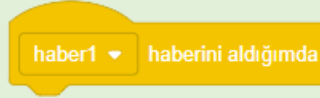
Mesajı Yayınla ve Bekle



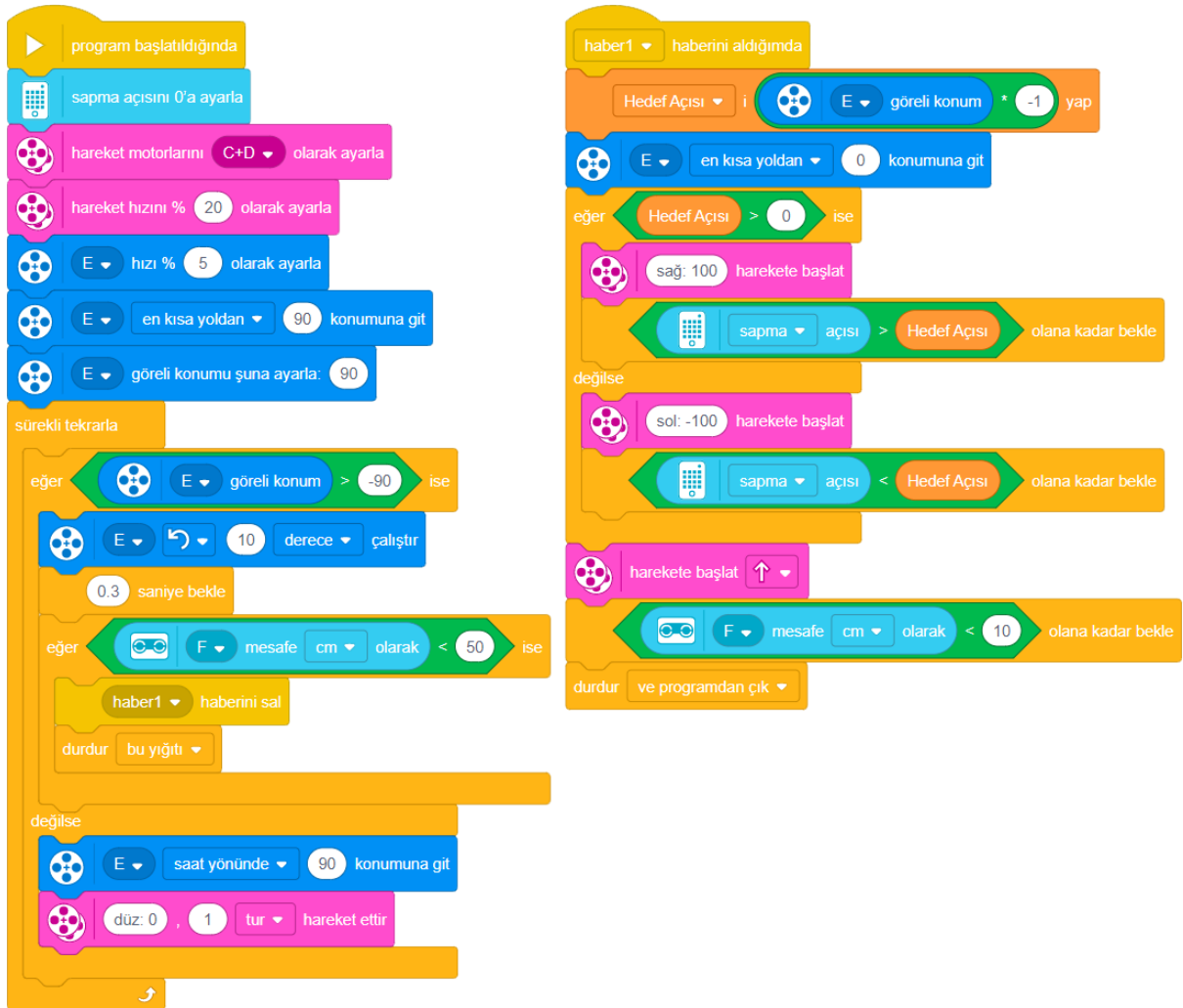
Bu blok belirli bir haber yayınlar. Haber yayınlandıktan sonra, tüm “mesaj aldığında” şapka blokları çalıştırılır. “mesaj yayınla” bloğundan farklı olarak program akışı durdurulur ve tüm

“mesaj aldığımda” şapka bloğu ile başlayan yığıtlar tamamlandıktan sonra, program akışı kaldığı yerden devam eder.

Mesaj Aldığımda



Bu blok “mesaj yayınla” ve “mesajı yayınla ve bekle” blokları tarafından gönderilen haberi aldığında kendisine bağlı bulunan blokları çalıştırır.



Resim 8.23 Hedef Tespit Robotu Örnek Programı

Rehber Öğretmen İçeriği – Hedef Tespit Robotu Etkinliği Açıklaması

Sürüş modeli tasarımında büyük motor (E girişine takılı) ters durduğu için, büyük motorun görelî konumu, robotun sapma açısı ile ters yönlü olup, biri artarken diğeri azalmaktadır. Bu nedenle örnek programda Hedef Açısı değişkeni (robotun sapma açısı miktarı) büyük motorun görelî

konumunun eksi bir (-1) ile çarpımına eşitlenmiştir. Benzer durum 6. haftada gerçekleştirilen tasarla üret etkinliğinde de anlatılmıştı. Rehber öğretmenin bu durumu öğrencilere hatırlatması faydalı olacaktır.

Ayrıca robotun hedefin konumunu daha kusursuz bir şekilde tespit etmesi isteniyorsa, mesafe sensörünün 10'ar derecelik artışları daha küçük derecelere, örneğin 5 veya 3 dereceye ayarlanabilir. Fakat böyle bir durumda robotun hedefi tespit etme süresi artacaktır.

DEĞERLENDİR

Gün sonunda öğrencilerle halka oluşturulur. Sensör isimleri kâğıtlara yazılır ve tombala oyunu misali her grubun bir kâğıt seçmesi istenir:

- Öğrencilerden seçilen sensörlerin özelliklerini anlatmaları ve bu sensörlerin hangi projelerde, nasıl kullanılabileceğini açıklamaları istenir.
- Sonrasında iki farklı sensörü olan gruplar bir araya gelirler. Öğrencilerden kâğıtlarındaki sensörlerin ikili olarak hangi projelerde, nasıl kullanılabileceğini örneklendirmeleri istenir.
- Bütün sensörlerin bir arada kullanılabileceği projeler tartışılır.
- Gün içerisinde öğrenilen kod blokları ve görevleri öğrenciler tarafından özetlenir.
- Bu haftanın en zor görevi ve öğrencilerin bu zorluğun üstesinden nasıl geldikleri konuşulur.

Proje Hazırlıyorum- Hazırlık Aşaması

Öğrencilerin öğrendiklerini grup hâlinde uygulayabilecekleri proje geliştirme sürecinin, eğitimin yaklaşık son 4 haftasını kapsayacak şekilde planlanması beklenmektedir. Projelerin *hazırlık, tanımlama, empati, fikir üretme, geliştirme-test etme ve sunum* gibi etkinliklerle yapılması ve rehber öğretmenler gözetiminde öğrencilere ders dışı etkinliklerle destek olunması önerilmektedir. Bu hafta ile projeye hazırlık aşamasıyla başlanmış olur.

Açıklama: Bu haftadan itibaren gruplar iki hafta boyunca projelerini tanımlama ve tasarlama üzerinde çalışacaklardır. Daha sonra, tanımlayıp tasarladıkları robotu oluşturmaları ve gerekli programları geliştirmeleri beklenir. Öğrenciler öncelikle çevrelerinden veya günlük hayatlarından uygun bir problem belirleyip bu probleme kendi tasarladıkları bir robot ile çözüm üretirler. Öğrencilerin problem üzerinde sistematik bir şekilde çalışmaları için grupların problemi analiz etmeleri ve kendi çözüm önerilerini tasarlamaları, geliştirmeleri ve değerlendirmeleri istenir. Bu süreçte karşılaşılabilecek tasarım ve programlamayla ilgili problemlerde rehber öğretmenler, öğrencilere uygun zaman ve miktarda destek olurlar.

Sonraki Haftaya Hazırlık: Bir sonraki hafta için öğrencilerin çevrelerinden ve günlük hayatlarından robotlar kullanılarak çözülebilecek bir problemi belirlemeleri gerektiği iletilir. Uygun bir problem sorusu belirlemeden önce problemi anlamak gerekir. Belirlenen problem, projenin zorluklarını ve hedeflerini tanımlamaya yardımcı olur. Sonrasında problemle ilgili yoğun bir araştırma ve gözlem yapılması gerekir. Süreç, projenin tamamlanması için gerekli bilgiler sağlar ve ortam hakkında bilgiler sunar. Süreç sonunda çok vakit ve emek harcanarak geliştirilecek robotun, insanların işine yaraması, bir problemini çözmesi ve hayatlarını kolaylaştırması istenir.

9. Hafta – Oransal Programlar

Haftanın Amacı:

Bu haftanın amacı öğrencilere oransal (proportional) algoritmayı tanıtmaktır. Hazırlanacak oransal programlar ile robotun hedeflen durumdan ne kadar uzaklaştığının tespit edilmesi (hatanın bulunması) ve hedeflenen duruma tekrar gelebilmesini sağlayacak tepki hareketi gerçekleştirilecektir. Önceki haftalarda gerçekleştirilen bazı etkinlikler oransal programlar ile yapılarak bu kodlama yaklaşımının verimliliğinin öğrenciler tarafından fark edilmesi amaçlanmıştır.

Haftanın Kazanımları:

- Öğrenciler sensörler aracılığıyla okunan veriler üzerinde işlemler gerçekleştirebilirler.
- Öğrenciler hata kavramını açıklayabilirler.
- Öğrenciler belirlenen bir koşul için hatayı hesaplayabilirler.
- Öğrenciler hesaplanan hata için düzetmeyi belirleyebilirler.
- Öğrenciler oransal algoritmayı kendi cümleleri ile ifade edebilirler.
- Öğrenciler uygun problemler için oransal programlar hazırlayabilirler.

Kullanılacak Malzemeler:

Robot seti, bilgisayar ve çalışma alanı (mat).

Ekler:

Bu hafta aşağıda sıralanan programlar ve önceki hafta kullanılan çalışma alanı (mat) dosyaları ekte rehber öğretmenlere sunulmuştur. Rehber öğretmenin, tüm gruplar için çalışma alanının çıktısını almasına gerek yoktur. Önceki hafta gerçekleştirilen etkinliklerde çalışma alanı zarar gören veya yenilenmesi gereken gruplar için çıktı alınması yeterlidir.

Program_9.1_Arac_Takip.llsp3

Program_9.3_Yonunu_Kaybetmeyen_Robot.llsp3

Program_9.4_Yonunu_Kaybetmeyen_Robot_TepkiGucu.llsp3

Program_9.7_Oransal_Cizgi_Takibi.llsp3

Program_9.11_Oransal_Duvar_Takibi_Donme.llsp3

Mat_8_A3.pdf

Mat_8_A4.pdf

Oransal Programlar

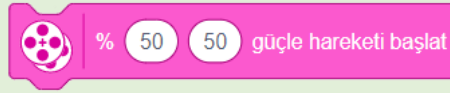
Rehber Öğretmen İçeriği – Oransal Programlar

Bu hafta öğrenciler oransal (proportional) program uygulamaları gerçekleştireceklerdir. Oransal algoritma yaygın olarak yumuşak hareketlerle çizgi takibi gerçekleştirilirken kullanılmaktadır. Genel olarak bir sensör ile algılanan veri ile robotun vereceği tepki belirlenir. Hedeflenen durum ile mevcut durum arasındaki fark (hata) küçük olması durumunda küçük tepkilerle, büyük olması durumunda büyük tepkilerle robotun hedeflenen duruma ulaşması sağlanır.

Bu hafta gerçekleştirilecek etkinliklerde, önceki haftalarda öğrenilen program bloklarına ek olarak “daha fazla hareket” eklentisinde yer alan “belirli güçle harekete başla” bloğu kullanılacaktır.

Daha Fazla Hareket

Belirli Güçle Harekete Başla



Bu blok, her bir motorun gücünün farklı değerlerde ayarlanarak hareketin başlamasını sağlar. Sürüş Modeli için soldaki değer sol motor, sağdaki değer ise sağ motor gücünün belirlenmesi için kullanılır. Bu blok hızı düzenlemez. Özellikle kendi kendine dengede duran (Gyro boy) ve çizgi takibi robotu gibi hızdan ziyade gücün düzenlenmesi gereken durumlarda kullanılır.

Not

Oransal programlama algoritmasının öğrenciler tarafından anlaşılması oldukça önemlidir. Konu anlatımında ekran görüntüleri sunulan ekteki örnek programların öğrencilere yansıtılması ve öğrencilerin bir kopyasını bakarak oluşturmaları bu algoritmanın anlaşılması için yeterli değildir. Rehber öğretmenler gözle etkinliklerinde algoritmanın nasıl olduğunu ve her bir bloğun görevini açıklayarak örnek programları adım adım gerçekleştirmelidirler. Üret etkinliklerinde ise öğrencilerin sunulan probleme kendilerinin çözüm üretmelerine olanak sağlayacak düzeyde rehberlik sağlanmalıdır. Örnek programlar kesinlikle öğrenciler ile **paylaşılmalıdır**.

GÖZLE VE UYGULA

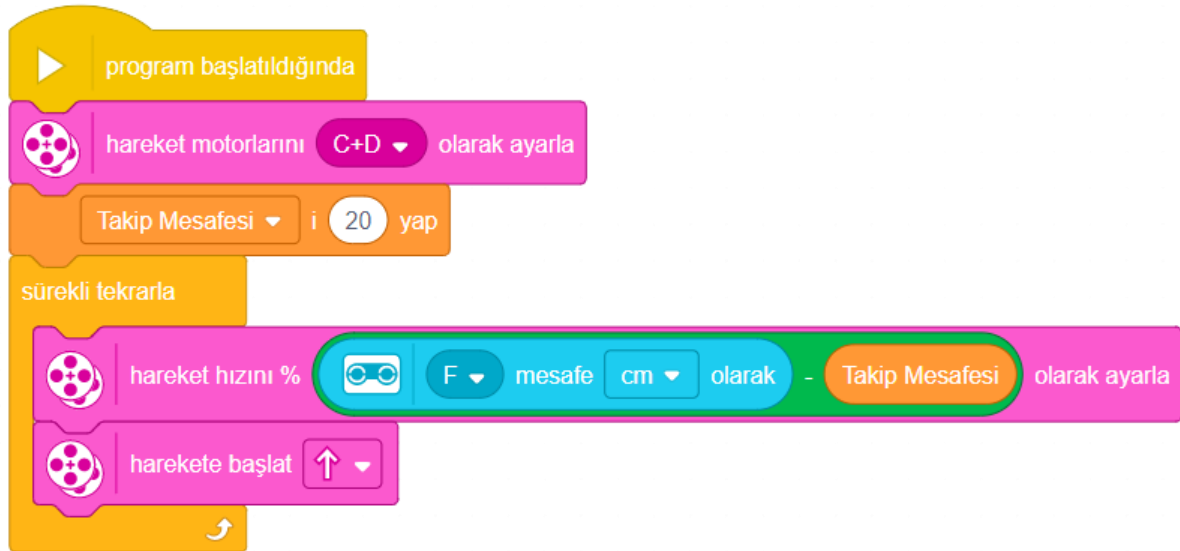
Gözle ve Uygula: Öndeki Aracı Takip Eden Robot

Rehber öğretmen, mesafe sensörünün anlatıldığı beşinci hafta yapılan öndeki aracı takip eden robot etkinliğini hatırlatır. Bu etkinlikte de benzer bir görevi yani robotun karşısında bulunan bir nesneyi belirli bir mesafeden takip edeceğini, hedef nesne robottan uzak mesafede ise robotun hızlı bir şekilde nesneye yaklaşacağını, takip mesafesi azaldıkça hızını azaltacağını ve belirlenen mesafe kadar yaklaştığında robotun duracağını belirtir. Beşinci haftada yapılan etkinlikte, robot ile nesne arasındaki mesafe kademeli olarak artıkça, hareket hızı da artmaktaydı. Örneğin robot ile öndeki nesne arasındaki mesafe 20 cm'den büyükse robotun hareket hızı %40, 40 cm'den

büyükse robotun hareket hızı %60 olarak değiştiriliyordu. Bu etkinlikte robotun hareket hızının dinamik olarak belirleneceği, robotun nesne ile arasındaki mesafeye otomatik olarak tepki verebilmesinin sağlanacağı belirtilir.

Robotun temel görevi, takip edilen nesne ile arasındaki mesafeyi sürekli ölçerek, ölçülen bu değer ile takip mesafesi arasındaki farkı hareket hızı olarak belirlemektir. Örneğin 20 cm olarak belirlenen takip mesafesi için, nesne ile robot arasındaki uzaklık 60 cm olduğunda robot $(60 - 20 = 40)$ %40, 56 cm olduğunda robot %36, 24 cm olduğunda robot %4 ve 20 cm olduğunda robot %0 hızla önündeki nesneyi takip edecektir.

Rehber öğretmen Resim 9.1’de verilen örnek programı adım adım hazırlar ve bu süreçte öğrencilere açıklar. Öğrenciler önceki hafta tasarladıkları mesafe sensörü takılı Sürüş Modeli robotunda (Resim 8.21 veya Resim 8.2) herhangi bir değişiklik yapmadan programı kendileri de hazırlayarak çalıştırır. Takip edilecek nesne olarak ellerini kullanabilirler. Robotun ellerine olan uzaklığa göre nasıl tepki verdiğini gözlemlerler. Ellerini robota çok yaklaştırdıklarında (örneğin 10 cm kadar) robotun neden geriye doğru hareket ettiğini açıklarlar.

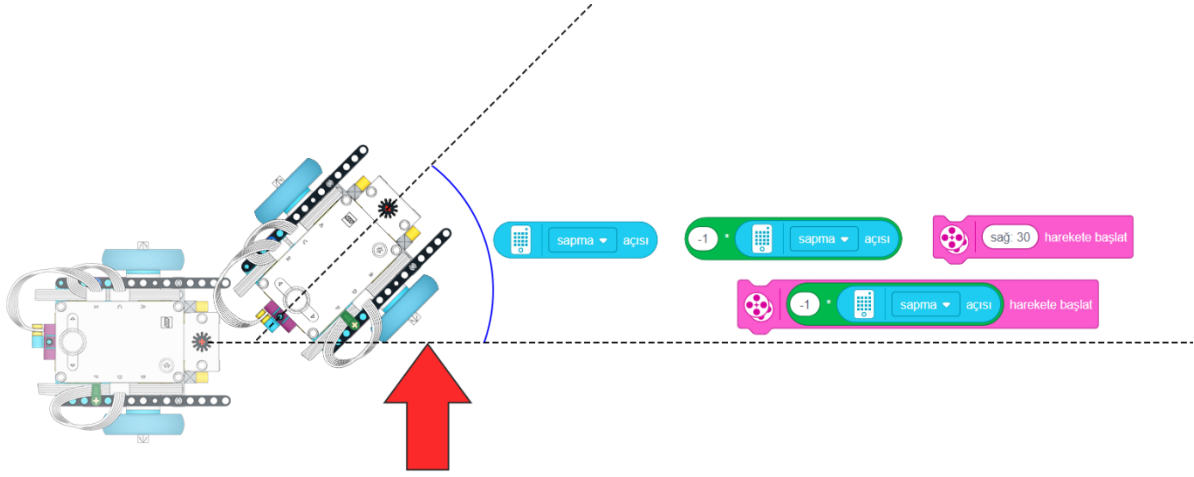


Resim 9.1 Öndeki Aracı Takip Eden Robot Örnek Programı

Gözle ve Uygula: Yönünü Kaybetmeyen Robot

Bu etkinlikte 6. hafta hareket sensörü konusunda gerçekleştirilen yönünü kaybetmeyen robot etkinliği, oransal algoritma yaklaşımı ile tekrar gerçekleştirilecektir. Öğrencilere yönünü kaybetmeyen robot etkinliğinde düz bir şekilde ilerleyen robotun dışarıdan yapılacak bir müdahale sonrasında tekrar başlangıçtaki doğrultusuna dönmesini sağlayan robotu programladıkları hatırlatılır.

Temel olarak yapılması gereken; robotun sürekli olarak başlangıçtaki rotasından ne kadar saptığının tespit edilmesi ve sapma miktarı kadar robotun ters yönde hareketlinin sağlanmasıdır. Resim 9.2’de robotun sapma açısı -1 ile çarpılarak ters yönde gerçekleştirilecek dönme açısı ve bu değer robotun hareket yönü olarak belirlenmesi gösterilmiştir. Robotun ters yönde hareketi sapma açısının zamanla azalmasını sağlayacaktır. Böylece robotun ters yönde hareketi de sapma açısına bağlı olarak zamanla azalacak ve robot bir süre sonra başlangıçtaki yönünü bulacaktır.



Resim 9.2 Yönünü Kaybetmeyen Robot

Rehber öğretmen Resim 9.3'te verilen programı adım adım hazırlayarak öğrencilere anlatır. Öğrenciler benzer bir programı hazırlayarak robotlarında çalıştırırlar.

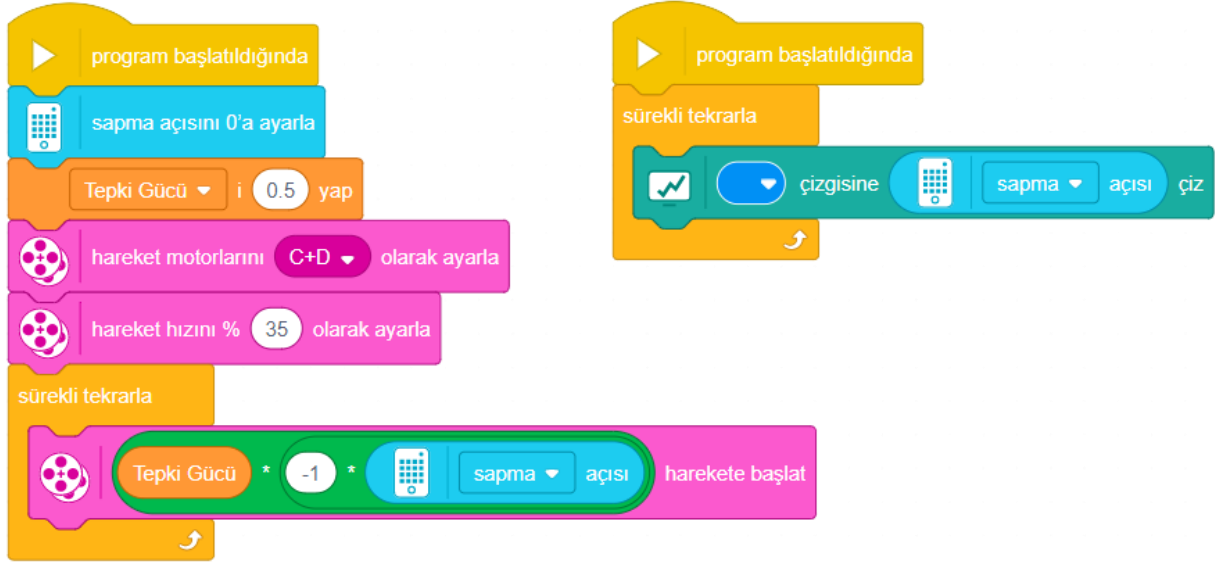


Resim 9.3 Yönünü Kaybetmeyen Robot Örnek Programı

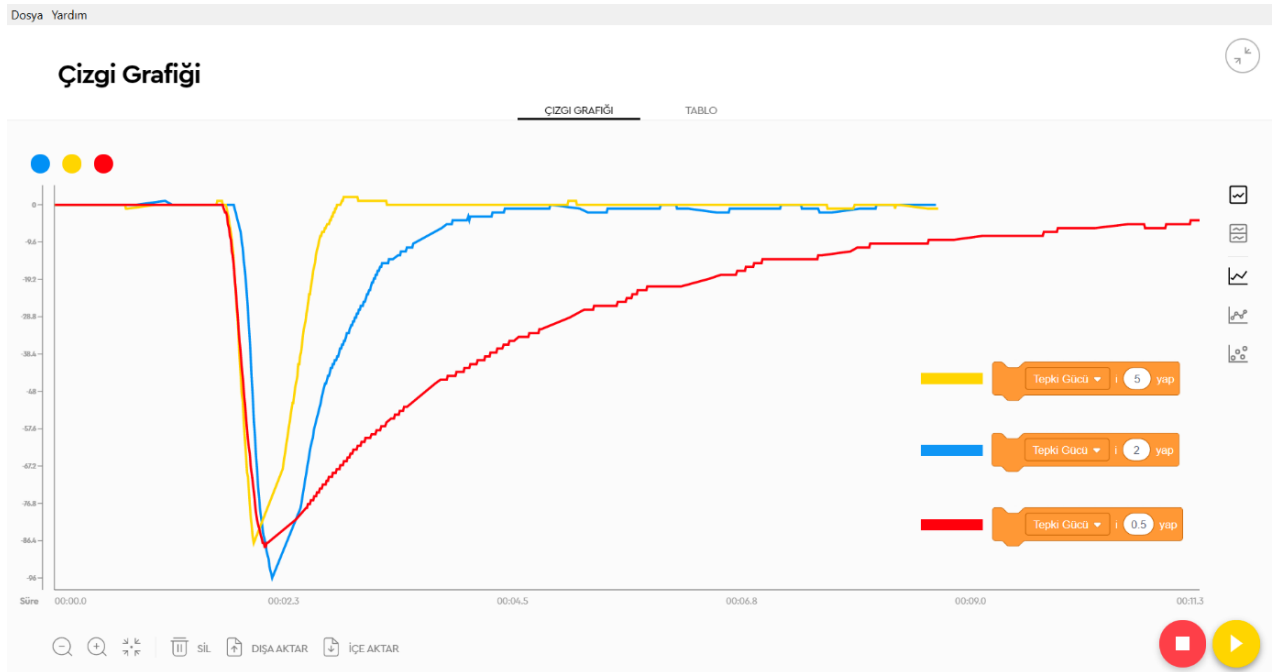
Gözle ve Uygula: Tepki Gücü

Bir önceki etkinlikte hazırlanan program, robotun tepki gücü belirlenerek zenginleştirilebilir. Rehber öğretmen Resim 9.4'te gösterildiği gibi, bir önceki programa "Tepki Gücü" isminde bir değişken ekleyerek, robotun dönme miktarının bu değişken ile çarpıldığı programı hazırlar. Böylece robotun sapma açısına vereceği tepki büyüklüğü bu katsayı ile belirlenir. Ayrıca eş zamanlı olarak robotun sapma açısının zamana bağlı grafiğinin çizilmesi sağlanır.

Rehber öğretmen öğrencilerden "Tepki Gücü" değişkeninin 0.5, 2 ve 5 değerleri için robot düz giderken sağ veya sola 90 derece dönecek şekilde müdahale edildiğinde; robotun başlangıçtaki doğrultusuna dönme tepkisini ve oluşan grafiği incelemelerini ister. "Tepki Gücü" değişkeni robotun tepkisinde ve grafikte ne gibi değişikliklere sebep olduğu öğrenciler ile tartışılır. Şekil 9.5'te "Tepki Gücü" değişkeninin 0.5, 2 ve 5 değerleri için çizilen çizgi grafiği sunulmuştur.



Resim 9.4 Yönünü Kaybetmeyen Robot Tepki Gücü Örnek Programı



Resim 9.5 Yönünü Kaybetmeyen Robot Tepki Gücü Grafiği

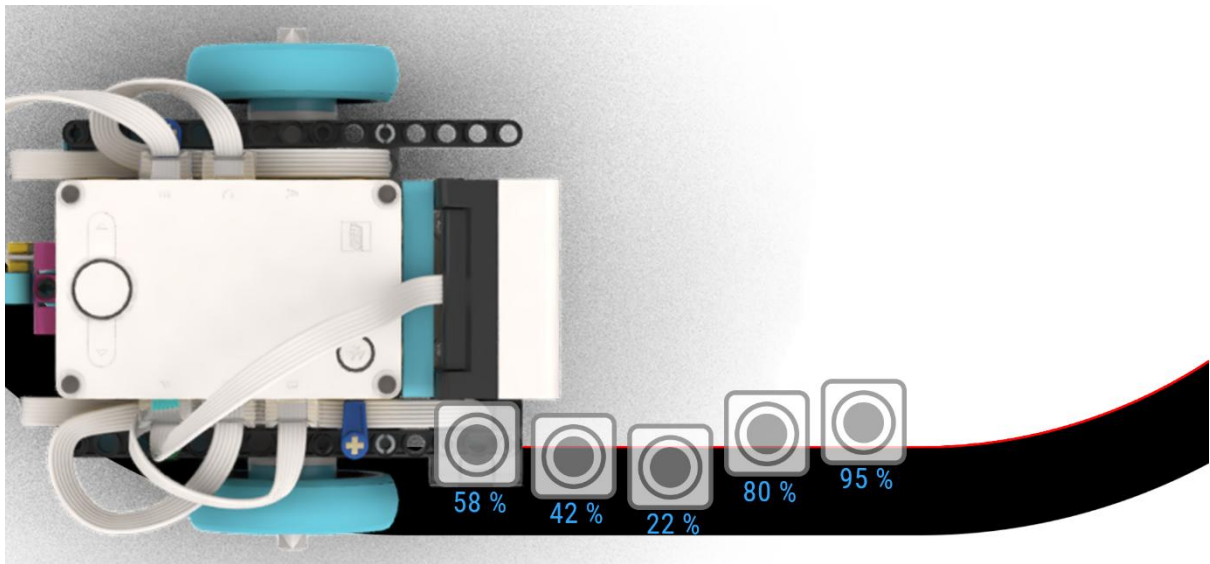
Not

Resim 9.5'te gösterilen grafik üç farklı grafiğin birleştirilmesi ile oluşturulmuştur. Öğrenciler programı her çalıştırdığında yalnızca bir çizgi grafiği elde edeceklerdir. Öğrencilerden üç grafiği bir arada oluşturmaları beklenmemektedir.

Gözle ve Uygula: Oransal Çizgi Takibi

Önceki haftalarda gerçekleştirilen çizgi takibi etkinliğinde, robotun beyaz ve siyah alanları yansıyan ışık miktarı ile fark etmesi sağlanıyor ve beyaz alandan siyah alana, siyah alandan beyaz alana robotun sürekli hareketi ile çizgi takibi gerçekleştiriliyordu. Bu etkinlikte oransal algoritma kullanılarak robotun daha hızlı ve yumuşak dönüşler ile çizgi takibini gerçekleştirmesi sağlanacaktır.

Resim 9.6'da robotun takip etmesini amaçlanan, siyah ve beyaz alan arasındaki sınır çizgisi kırmızı ile gösterilmiştir (çalışma alanında kırmızı çizgi bulunmamaktadır, kırmızı çizgi konunun daha rahat anlaşılması için eklenmiştir). Ayrıca renk sensörünün sınır çizgisi üzerinde, siyah alanda sınır çizgisinden farklı uzaklıklarda ve beyaz alanda sınır çizgisinden farklı uzaklıklarda ölçülen yansıyan ışık miktarları gösterilmiştir.



Resim 9.6 Renk Sensörü Yansıma Miktarı

Hedeflenen hareket robotun siyah ve beyaz alanın kesiştiği sınır çizgisi (kırmızı çizgi) üzerinde hareket etmesidir. Başka bir deyişle robotun yansıyan ışık miktarının %58 olduğu durumda hareket etmesidir. Robotun siyah veya beyaz alanda hareket etmesi, hedeflenen hareketi gerçekleştirmediğini yani **hata** yaptığını gösterir. Robot sınır çizgisinden ne kadar uzaklaşırsa, yansıyan ışık miktarı da o kadar artmakta (beyaz alan için) veya azalmaktadır (siyah alan için). Yani ışık sensörü ile ölçülen yansıyan ışık miktarı, sınır çizgisi üzerindeki yansıma miktarından (%58) uzaklaşmaktadır. Yansıyan ışık miktarı ile sınır çizgisi üzerindeki yansıma miktarının farkı robotun hedeflenen hareketten ne kadar uzaklaştığını yani hata büyüklüğünü verecektir. Robotun yaptığı hata büyüklüğü;

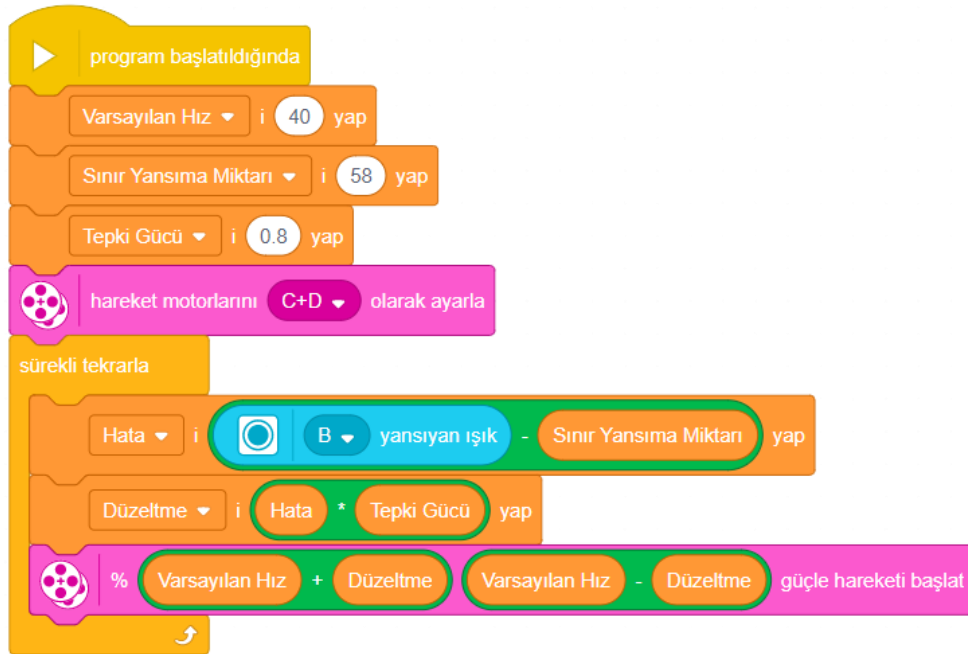
Hata = Yansıyan ışık miktarı – Sınır çizgisi üzerindeki yansıma miktarı formülü ile bulunur.

Robotun ters yönde hareket etmesi, tekrar sınır çizgisine dönmesini, yani hatayı sıfırlamasını sağlar. Hata ne kadar büyük ise yani sınır çizgisinden ne kadar çok uzaklaşmış ise o kadar güçlü bir şekilde sınır çizgisine yaklaşılmalıdır. Robotun bu hareketi **düzeltilme** hareketi olarak adlandırılabilir. Düzeltme hareketinin yumuşak veya sert olması bir önceki etkinlikte incelenen tepki gücü çarpanı ile belirlenebilir. Böylece düzeltme miktarı;

Düzeltilme = Hata * Tepki Gücü formülü ile düzenlenir.

Son olarak belirlenen düzeltme miktarı ile robotun yönünü değiştirilmelidir. Resim 9.6'da gösterildiği gibi çalışma alanının iç kısmında ve saat ibresinin tersi yönde çizgi takibi gerçekleştirecek robot için, düzeltme miktarının sol motor gücüne eklenmesi, sağ motor gücünden çıkartılması ile istenen çizgi takibi gerçekleştirilebilir. Örnek program Resim 9.7'de sunulmuştur.

Rehber öğretmen özellikle hata, düzeltme değişkenlerinin nasıl hesaplandığını ve *“belirli güçle harekete başla”* bloğunun nasıl çalıştığını anlatarak, benzer bir programı adım adım oluşturur. Öğrenciler, bilgisayar ve robotları ile çalışarak farklı şekilde (örneğin çalışma alanının dış çizgisini saat ibresi yönünde) çizgi takibi yapan robot uygulamaları gerçekleştirirler.



Resim 9.7 Oransal Çizgi Takibi Örnek Programı

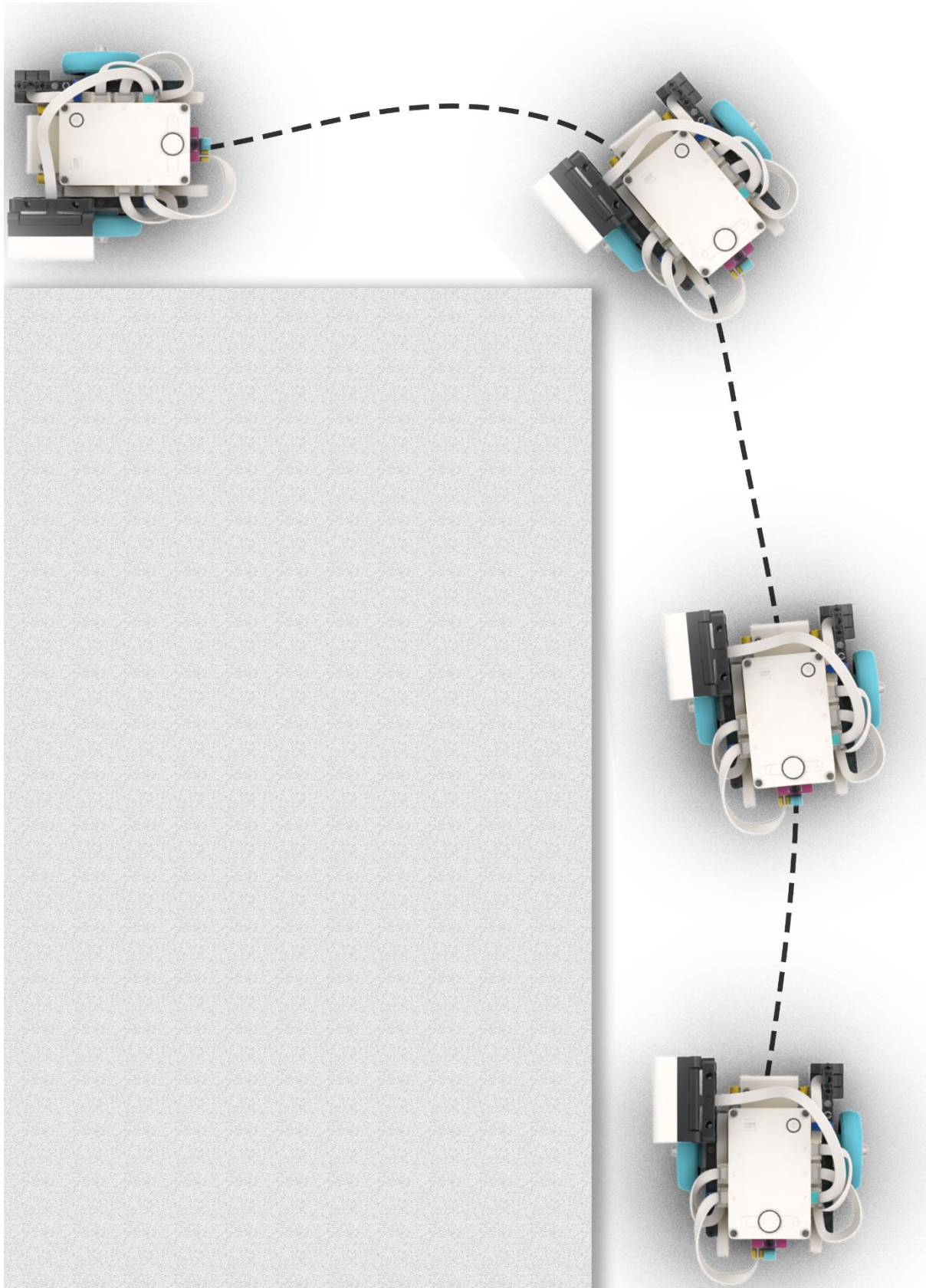
Not

Resim 9.7'de sunulan program her ortamda sorunsuz olarak çalışmayabilir. Ortama bağlı olarak *Sınır Yansıma Miktarı*, *Varsayılan Hız* ve *Tepki Gücü* değişkenlerinin değiştirilmesi gerekebilir. *Sınır çizgisi üzerindeki yansıma miktarı*, etkinliğin gerçekleştirildiği ortamda ölçülüp, bulunan değer değişkene atanmalıdır.

TASARLA

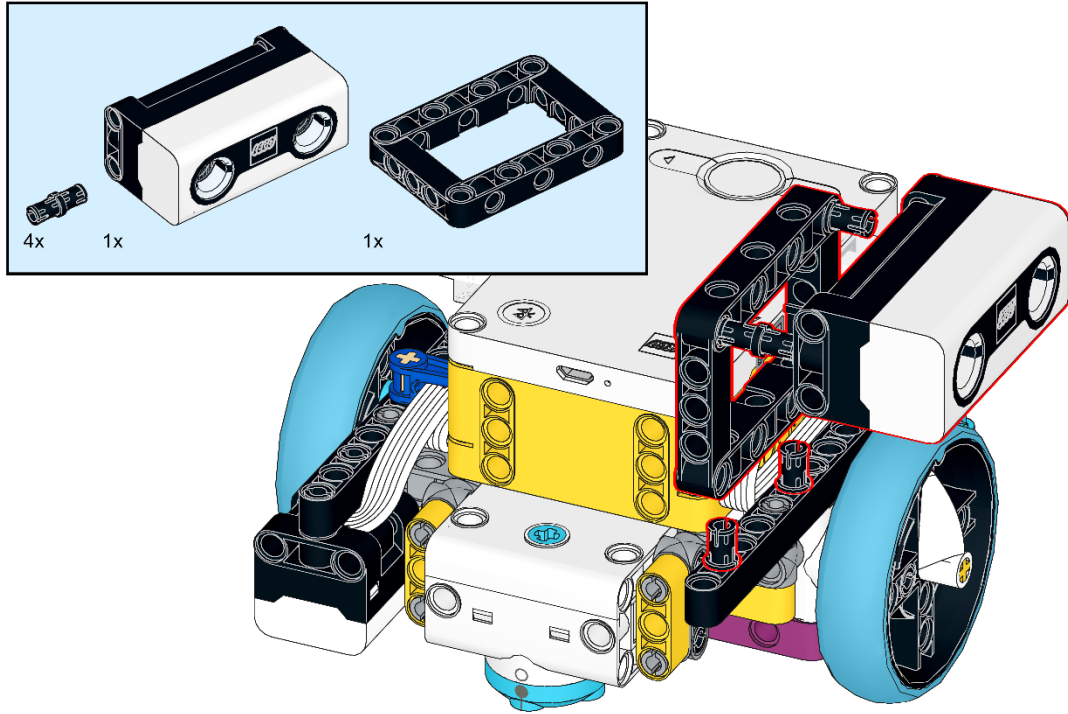
Tasarla etkinliği için rehber öğretmen oransal duvar takibi gerçekleştiren robot tasarımını ve programını gerçekleştireceklerini belirtir. Genel olarak robotun duvarı belirli bir uzaklıktan takip edeceği ve duvar bitiminde duvara doğru 90 derecelik dönüşü başarılı bir şekilde gerçekleştirmesi gerekmektedir. Hazırlanacak programda herhangi bir koşul bloğu kullanılmamalı ve oransal

algoritma kullanılmalıdır. Resim 9.8’de robotun gerçekleştirmesi beklenen hareket kesikli çizgilerle gösterilmiştir.

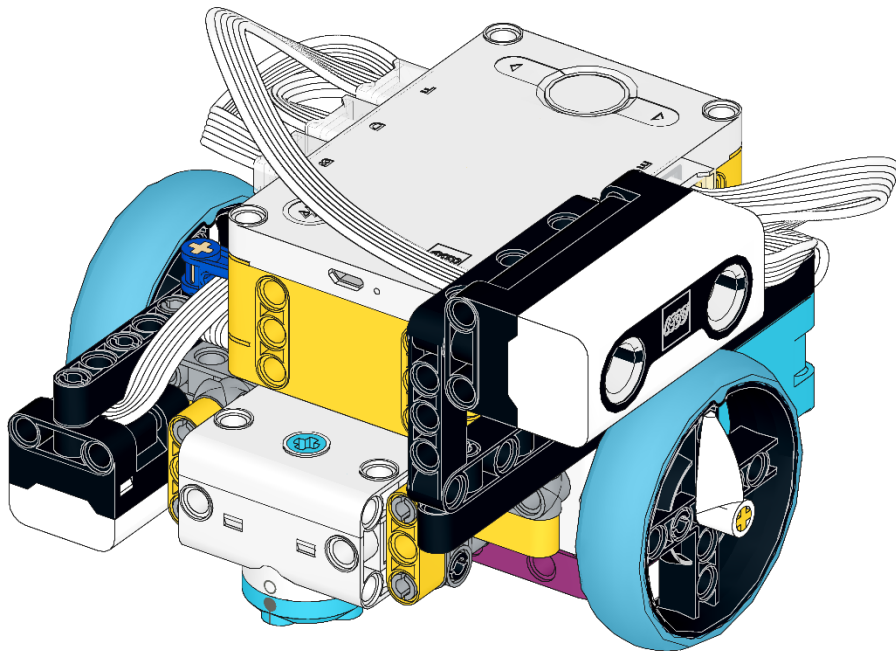


Resim 9.8 Oransal Duvar Takip Robotu

Öğrenciler robot tasarımlarında istedikleri değişiklikleri yapabilirler. Mesafe sensörünü istedikleri şekilde konumlandırabilirler. Örneğin, önceki hafta yaptıkları hedef tespit robotunda mesafe sensörünün yönünü değiştirmekle yetinebilirler. Resim 9.9 ve 9.10'da bu etkinlik için mesafe sensörü birleştirme önerisi sunulmuştur.



Resim 9.9 Mesafe Sensörünün Takılması Birinci Adım



Resim 9.10 Mesafe Sensörünün Takılması İkinci Adım

Robotun görevi sadece duvarı takip etmektir. Duvarın bitiminde ise yine duvarı takip etmeye devam etmelidir. Görev basit gibi görünse de robotun oransal takip programı ile köşeyi dönebilmesi hassas bir ayarlama gerektirmektedir. Robot duvarın köşesine geldiği anda, mesafe sensörü ile ölçülen uzaklık bir anda arttığından robot kendi etrafında sürekli dönebilmektedir.

Bu problemin çözümünde öğrenciler, farklı çözüm yollarını test edebilirler. Robot tasarımında istedikleri değişiklikleri yapabilirler. Örneğin mesafe sensörünün duvara paralel değil de, 30 derecelik bir açı ile konumlandırılması daha rahat bir dönüş sağlasa da duvar takibi esnasında zaman zaman mesafe sensörün ölçüm yapamaması ile sonuçlanabilmektedir. Duvara olan uzaklığın azaltılması köşedeki dönmeyi kolaylaştırırsa da duvar takibi esnasında duvara çarpmalarla sonuçlanabilmektedir. Benzer şekilde tepki gücünün yüksek olması dönme esnasında problemlere, düşük olması duvar takibi sürecinde problemlere neden olmaktadır. Bu nedenle öğrenciler robot tasarımında ve programlarında (oransal algoritma kullanma ve koşul kullanmama şartıyla) özgürdürler.

Tanımlama: Bu süreçte öğrenciler problemi tanımlamalı ve olası alt problemlere karar vermelidir. Bu etkinlikte önceki haftalarda olduğu gibi, robotun bir dizi görevi sırası ile gerçekleştirmesinden ziyade, robotun sürekli olarak duvarı takip etmesi ve takip programı parametrelerinin (ve tasarımının) robotun 90 derecelik bir dönüşü de başarılı bir şekilde gerçekleştirmesini sağlayabilmelidir. Yine de öğrenciler robotun gerçekleştirmesi istenen görevleri kalem kâğıt kullanılarak yazmalıdır.

Örneğin:

- Robot sürekli olarak duvara olan uzaklığını ölçmelidir.
- Duvara istenilenden fazla yaklaştığında veya uzaklaştığında ters yönde hareket etmelidir.
- Robot köşeyi dönerken bir yay çizmelidir (keskin bir dönüş yapmamalıdır).

Fikir üretme: Öğrenciler gün içinde öğrendikleri oransal algoritma yaklaşımını, sunulan göreve nasıl uygulayabileceklerine ve hangi program bloklarını kullanacaklarına yönelik fikirler üretirler.

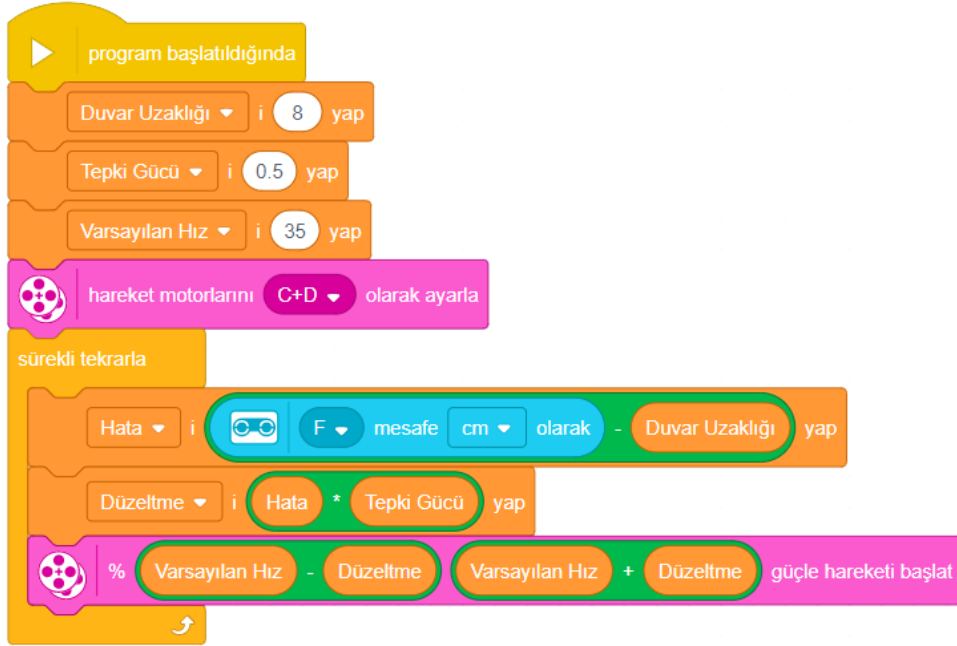
Örneğin:

- Mesafe sensörünün yönü ileriye doğru olmamalıdır.
- Mesafe sensörü sürekli olarak duvara olan uzaklığı ölçeceğinden sonsuz döngü kullanılmalıdır.
- Hata (istenilen rotadan sapma) hesaplanmalıdır.
- Hataya bağlı olarak düzeltme hesaplanmalıdır.
- Tepki gücü, takip mesafesi, robotun hızı değişkenleri robotun çalışmasına (ortama) bağlı olarak düzenlenmelidir.

ÜRET

Öğrenciler tanımlama ve fikir üretme süreci sonrasında sonra bilgisayar ve robotlarla çalışarak fikirlerini somutlaştırmalı ve test etmelidirler. İstenilen görevi gerçekleştiren robot tasarımını ve programını bu sürecin sonunda üretmelidirler. Robotun sorunsuz bir şekilde köşeyi dönmesini sağlayabilmek zorlayıcı bir görevdir. Rehber öğretmen bu süreçte öğrencilerin başarısız

denemelerden dolayı hayal kırıklığı yaşamalarına engel olmalıdır. Doğru sonucu direk söylemeden onları doğru sonuca ulaştıracak sorular ile yönlendirmeli ve rehberlik sağlamalıdır. Bu etkinlik için örnek program Resim 9.11’de sunulmuştur. Örnek programın her koşulda doğru şekilde çalışmayabileceği unutulmamalıdır. Farklı zemin ve ortamlar için program başlatıldığında değişkenlere atanan değerlerin örnekteki değerlerden farklı olması gerekebilir.



Resim 9.11 Oransal Duvar Takip Robotu Örnek Programı

DEĞERLENDİR

Gün sonunda rehber öğretmen öğrencilere aşağıdaki soruları yönelterek bu hafta öğrenilenlerin özetlenmesini sağlar.

- Oransal (proportional) programların mantığını özetler misiniz?
- Oransal programlara örnekler veriniz. Bu hafta yapılan etkinlikler dışında oransal program örnekleri verebilir misiniz?
- Bu hafta yapılan etkinlikleri, önceki haftalarda yaptığınız etkinliklerle kıyaslayınız?
- Oransal algoritmanın artıları ve eksileri nelerdir?
- Sizce bu haftanın en zor görevi hangisiydi? Zorluğun üstesinden nasıl geldiniz?

Proje Hazırlıyorum- Tanımlama ve Empati Aşaması

Öğrenciler çevrelerinden veya günlük hayatlarından uygun bir problem belirleyip bu probleme kendi tasarladıkları bir robot ile çözüm üreteceklerdir. Geçen hafta proje grupları oluşturulmuş ve proje konusunun araştırılması istenmiştir.

Bu hafta gruplardan belirledikleri robot projesi için empati çalışması yapmaları ve problemlerini tanımlamaları beklenir. Açıklamalarının aşağıdaki bilgileri içermesi gerektiği belirtilir. Empati çalışması sonucunu paylaşmaya başlamadan önce gruplara hazırlanma, gözden geçirme ve tekrar etme için süre verilir.

- Problem cümlesi: Robotun ne yapması isteniyor?

- Robotun kullanılması planlanan ortamdaki ihtiyaçlar nelerdir?
- Robotun kullanılması planlanan ortamdaki beklentiler nelerdir?

Açıklanan projelerle ilgili olarak diğer öğrencilerin ve rehber öğretmenlerin fikirleri alınır.

Rehber Öğretmene Not: Öğrenci tarafından belirlenen robot görevlerinin gerçekleştirilebilir olması önemlidir. Atölyelerde bulunan imkânlarla bu görevin başarılabilir olması gerekir. Bu safhada gerçekleştirilemeyecek bir görevin seçilmesi veya öğrencilerin bu tür bir projeye yönlendirilmesi, proje sürecinde vakit kaybetmelerine ve motivasyonlarının olumsuz etkilenmesine neden olabilir. Bu nedenle rehber öğretmenin proje önerilerini çok iyi incelemesi/irdelemesi ve proje önerilerinin gerçekleştirilebilir/yapılabilir olduğundan emin olması gerekir.

Sonraki Haftaya Hazırlık: Öğrencilerden, sonraki hafta için belirledikleri projeler için "tanımlama" adımlarını gerçekleştirmeleri istenir. Tanımlama adımı, "empati" adımı, toplanan bilgiler bir bütün hâline getirilir. Süreçte ortaya çıkabilecek problemleri çözebilmek için nelerin yapılması gerektiği ile ilgili fikirler ortaya konur. Çözümün başarılı olabilmesi için ihtiyaçların iyi tanımlanmış olması ve bu ihtiyaçların giderilmesi için uygun çözüm önerilerinin sunulması önemlidir.

10. Hafta- Bloklarım

Haftanın Amacı:

Bu haftanın amacı öğrencilere bloklarım menüsünü tanıtmak ve oluşturdukları blokları programlarında kullanmalarını sağlamaktır. Öğrencilerin blokları kullanmaları, programlama sürecinde kolaylık, sadelik, hata tespiti kolaylaştırma ve işleyişi takip işleminde kolaylık sağlamaktadır.

Haftanın Kazanımları:

- Öğrenciler blok kavramını açıklayabilir.
- Öğrenciler blok oluşturup programların kullanabilir.
- Öğrenciler bloklara veri gönderme işlemini tanımlayıp bu blokları programlarında kullanabilir.
- Öğrencilere bloklara birden fazla veri gönderme işlemini tanımlayıp bu blokları programlarında kullanabilir.
- Öğrenciler robota sayı girişi için gerekli programı oluşturabilirler.

Kullanılacak Malzemeler:

Robot seti, bilgisayar.

Ekler:

Bu hafta aşağıda sıralanan programlar ekte rehber öğretmenlere sunulmuştur.

Program_10.1_Saga_Donen_Robot.llsp3

Program_10.2_Sapma_Acisi_ile_Saga_Donen_Robot.llsp3

Program_10.4_Bloklarla_Deger_Gonderme.llsp3

Program_10.5_Bloklarla_Deger_Gonderme.llsp3

Program_10.6_Hizlanmaya_Devam_Eden_Robot.llsp3

Program_10.7_Cokgen_Hareket_Blogu.llsp3

Bloklarım

Rehber Öğretmen İçeriği – Renk Sensörü

“Bloklarım” diğer sözcük bloklarından oluşturulan yeni kelime bloğudur. Bir dizi kelime bloğunu bir arada tutmak için kullanılır. Bir başka ifade ile “Bloklarım”, oluşturduğunuz ve tek bir blokta gruplandırılabilen bir veya daha fazla bloğun birleşimidir. Size özel ve programınız içinde tanımladığınız bloklardır. Oluşturduğunuz kodların düzenlenmesini, okunmasını, değiştirilmesini kolaylaştırır. Oluşturulan yeni bloklar program akışında istenildiği kadar sıklıkta eklenerek kullanılabilir. Kendi oluşturduğunuz blok tanımını değiştirdiğinizde, değişiklik yaptığınız bloğu kullandığınız tüm bölümler etkilenir.

Not: “Bloklarım” paletini girdiler ve çıktılarla yapmak onları çok daha kullanışlı hale getirebilir. Ancak oluşturduğunuz blokları çok karmaşık hale getirmemeye dikkat etmelisiniz. Blokları diğer programlama dillerinde kullandığınız fonksiyonlar olarak görebilirsiniz.

Bloklar Blok Paleti

Size özel ve programınız içinde bloklar oluşturup kullanmaya olanak sağlar.

Bir Blok Oluştur

Kendi bloklarınızı oluşturmaya ve özelliklerinizi belirlemeye olanak sağlar.

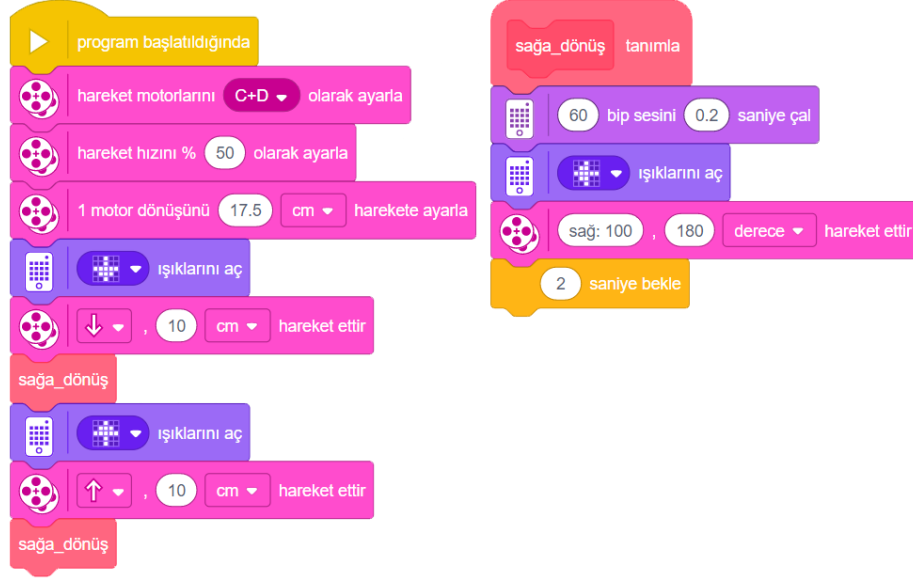
GÖZLE VE UYGULA

Göze: Bloklarım Paleti

Rehber öğretmen öğrencilere diğer sözcük bloklarını kullanarak kendilerine özgü bloklar oluşturabileceklerini ve bu blokları program akışında istedikleri kadar kullanabileceklerini anlatır. Oluşturulan blokların program kodlarının düzenlenmesini, okunmasını ve değiştirilmesini kolaylaştırdığını ifade eder. Bu blokları birçok sözcük bloğunu bir arada tutmak için kullanabileceklerinden, oluşturdukları programları sadece kendi programları içinde kullanabileceklerinden, başka programlarda kullanabilmek için bu blokları kopyalayıp diğer programlara yapıştırmaları gerektiğinden ve oluşturulan yeni blokları program akışlarında istedikleri kadar tekrar ekleyerek kullanılacaklarından bahseder. Ayrıca, oluşturdukları blok

tanımlarını değıştirildiğinde, değışiklik yaptıkları blokları kullanan tüm bölümlerin otomatik olarak etkilendiğini vurgular.

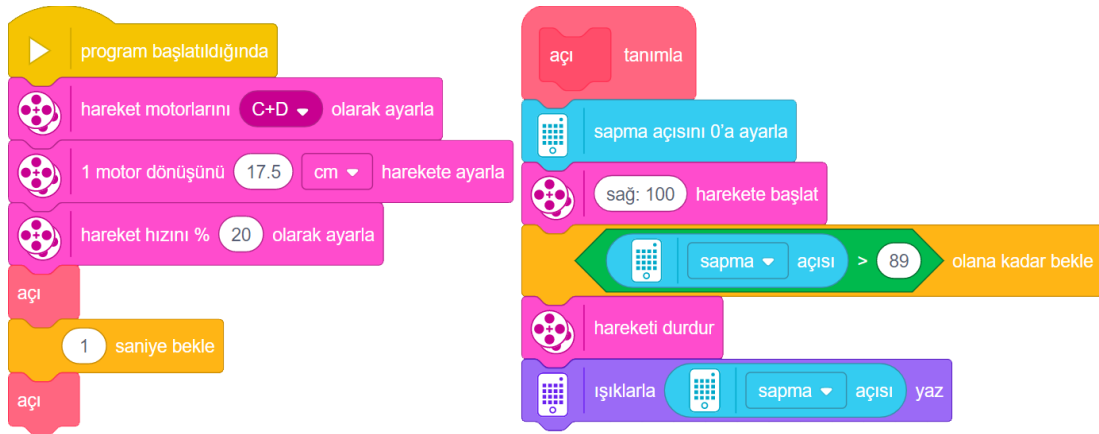
Öğrencilere kendi sözcük bloklarını oluşturabilmek için blok paletindeki “Blokларım” sekmesini kullanacaklarını gösterir. Öğrencilerle birlikte sağa dönüş için oluşturulan “sağa_dönüş” adlı bloğun iki defa kullanımını içeren program akışını yapar. “sağa_dönüş” adlı bloğun önce bip sesini 0.2 saniye çaldığını, ekranda sağ ok yönünü gösterdiğini, 10 cm sağa dönüş yaptığını ve 2 saniye bekleme yaptığını anlatır. Program çalıştırıldığında oluşturulan “sağa_dönüş” bloğunun 2 defa çalıştığını öğrencilere gösterir. Örnek çözüm Resim 10.1’deki gibi olabilir.



Resim 10.1 Blokларım Paleti Kullanarak Sağa Dönen Robot

Uygula: Sapma Açısı ile Sağa Dönen Robot

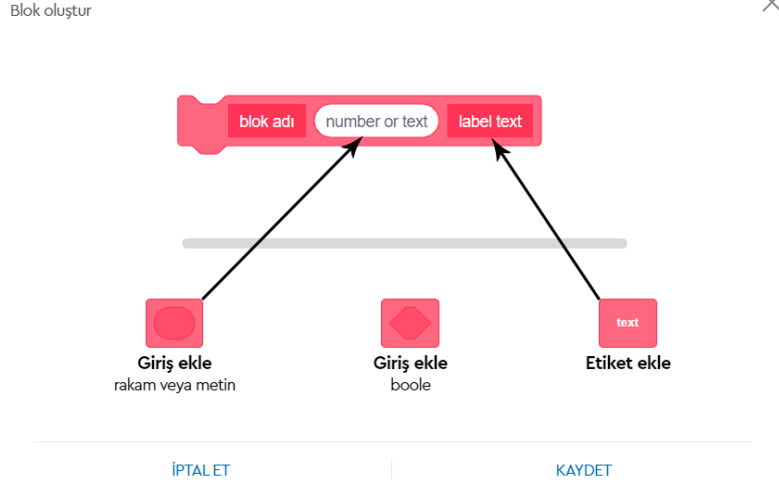
Rehber öğretmen öğrencilerden “Hub yunuslama yuvarlanma sapma açısı” bloğunu kullanarak ve kendi bloklarını oluşturarak 90 derece sağa dönen robotun program bloklarını oluşturmalarını ister. “Blokларım” paletini kullanarak oluşturdukları bloğun önce sapma açısını 0 dereceye ayarlamaları gerektiği, sonra sağa dönüş hareketine başlayarak sapma açısı 90 derece olana kadar dönmeye devam etmeleri gerektiği ve dönüş tamamlandıktan sonra sapma açısını ekrana yazdırmaları gerektiği söylenebilir. Örnek program Resim 10.2’de sunulmuştur.



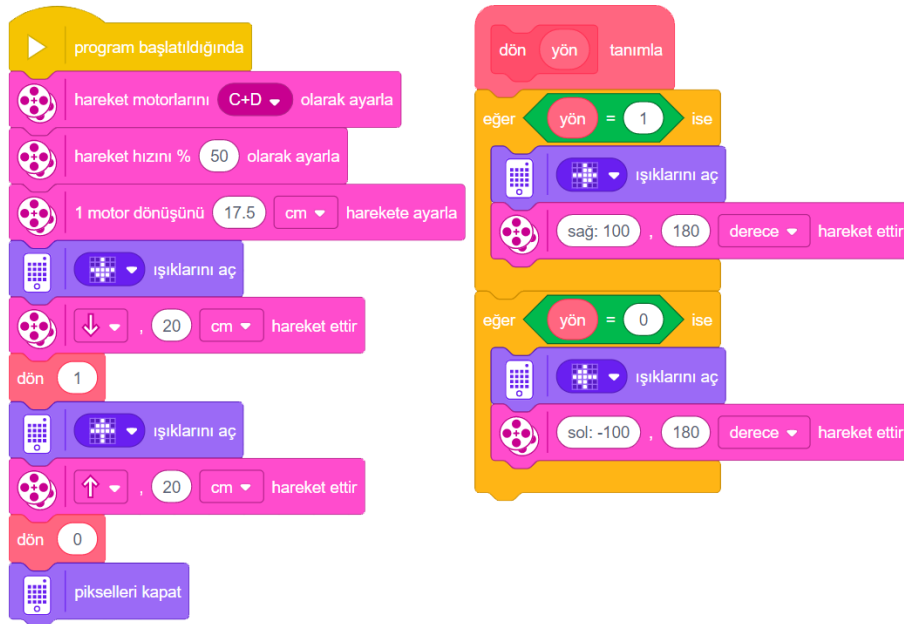
Resim 10.2 Sapma Açısı ile Sağa Dönen Robot

Gözele: Bloklarla Değer Göndermek

Rehber öğretmen öğrencilere oluşturduğumuz bloklara değerler de gönderilebildiğini ve gönderilen bu değerlere göre işlem yapan blokların oluşturulabileceğinden bahseder. “bir blok oluştur” bloğu tıklanarak açılan pencerede oluşturulacak bloğa, giriş, boole ve etiket bölümlerinin (Resim 10.3) eklenebileceğinden bahseder. Rehber öğretmen öğrencilere 1 girilince sağa 0 girilince sola dönen “dön” adında bir blok oluşturacakları söyler. “Blok oluştur” ekranında blok adı “dön” olarak belirlendikten sonra “giriş ekle” tuşuna tıklanarak “yön” adında bir giriş eklenmesi gerektiğini gösterir. Örnek program Resim 10.4’te sunulmuştur.



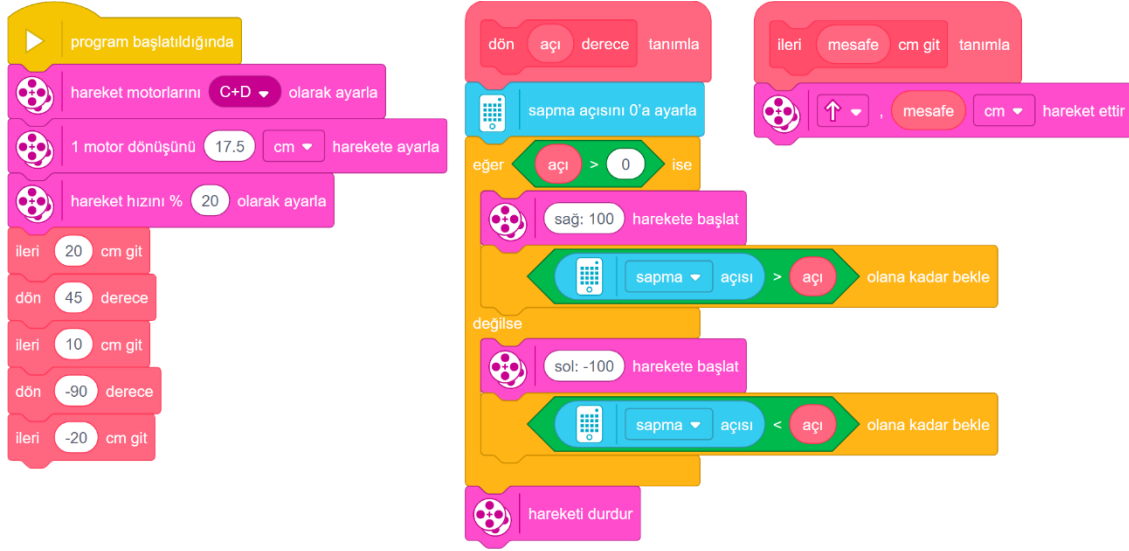
Resim 10.3 Blok Oluşturma Ekranı



Resim 10.4 Bloklarla Değer Gönderme

Uygula: Bloğa Gönderilen Değerler ile İlerleme ve Dönmenin Belirlenmesi

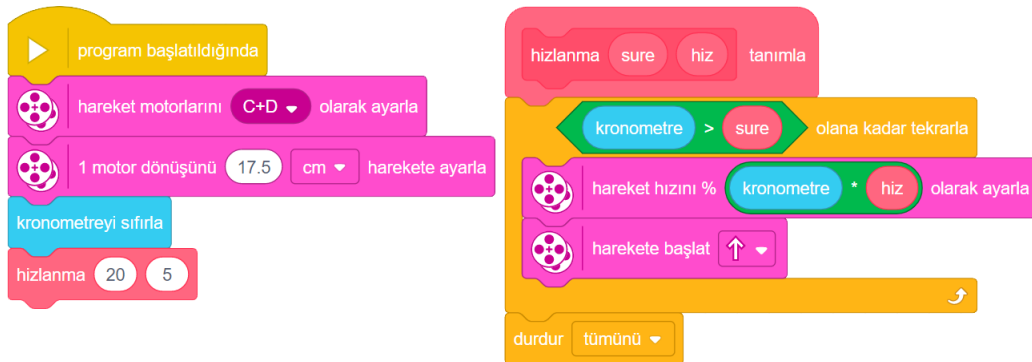
Rehber öğretmen, öğrencilerin iki farklı blok tanımlayacaklarını ve bu bloklara değerler gönderileceğini belirtir. Bloklardan birine gönderilen açı değeri kadar robotun dönmesi, diğerine ise girilen değer miktarı kadar cm ileri gitmesi gerektiği açıklanır. Öğrencilere, her dönüş öncesinde sapma açısının sıfırlanmasının kolaylık sağlayacağı hatırlatılır. Öncelikle, robotun girilen cm değeri kadar ileri gitmesini sağlayan bloğun oluşturulması, ardından sadece pozitif açılar için dönmenin gerçekleştirilmesi ve son olarak da negatif açı değerleri için dönmenin yapılması tavsiye olunur. Örnek program, Resim 10.5'te sunulmuştur.



Resim 10.5 Bloklarla Değer Gönderme

Uygula: Hızlanmaya Devam Eden Robot

Rehber öğretmen öğrencilere blok oluşturulurken birden fazla değer ekle bölümünün tanımlanabileceğinden bahseder. Öğrencilere zaman ve hızlanma (ivmelenme) değerleri gönderebilecekleri bir blok oluşturarak robotun belirtilen süre boyunca belirtilen miktarda hızlanarak (ivmelenerek) ilerlemesi için gerekli robot program bloklarını oluşturmalarını ister. Örneğin zaman değeri 10, ivmelenme değeri 5 olarak belirlendiğinde robot 0 hızla başlamalı ve her saniye geçtiğinde hızını 5 artırarak 20 saniye sonunda %100 hıza ulaşmalıdır. Örnek program Resim 10.6'da sunulmuştur.



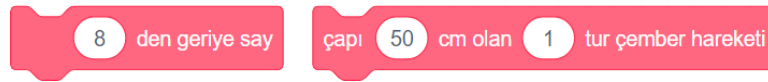
Resim 10.6 Hızlanmaya Devam Eden Robot

Not

Rehber öğretmen öğrencilere robota gönderilen hız değeri ne kadar büyük olursa olsun robotun hızının en fazla 100 olacağını hatırlatır.

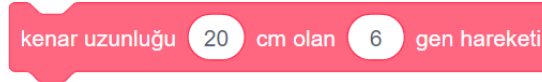
TASARLA VE ÜRET

Rehber öğretmen, öğrencilere tasarım etkinliği için SPIKE Prime Uygulamasında bulunmayan ve program yapmayı kolaylaştıracak kod blokları olup olmadığını sorar. Örneğin, bir bloğa girilen saniye değerinden geriye doğru sayan ve hub ekranında geri sayımı gösteren bir blok veya sürüş modeli robotu için çap uzunluğu verilen çember şeklinde istenilen tur sayısı kadar hareket etmeyi sağlayan blok gibi örnekler verilebilir. Bu örnek bloklar Resim 10.7’de sunulmuştur.



Resim 10.7 Örnek Bloklar

Rehber öğretmen, öğrencilerden gelen cevapları değerlendirir. Uygun bir blok önerisi olması durumunda, bu öneriyi dikkate alır; eğer öneri yoksa, bir kenar uzunluğu ve çokgenin köşe sayısının girilebildiği bir blok ile robotun istenilen çokgen şeklinde hareket etmesini sağlayan bir blok hazırlamalarını ister. Örnek blok Resim 10.8’de sunulmuştur. Rehber öğretmen kenar uzunluğu ve köşe sayısının sabit olmadığını vurgular, değerler değiştirildiğinde robotun çokgeni istenilen şekilde çizmesi gerektiğini belirtir. Ayrıca çokgen olmayan değerlerin girilmesi durumunda, örneğin 2 girilmesi durumunda robotun çalışmaması gerektiği vurgulanır.



Resim 10.8 Örnek Çokgen Çizdiren Blok

Süreç boyunca, öğrencilerin çokgenin her bir köşesinde robotun kaç derece dönmesi gerektiğini hesaplamaları gerekecektir. Eğer öğrenciler yeterli düzeyde ön bilgiye sahip değilse, rehber öğretmenin onlara çokgenin iç ve dış açılarını ve bu açıların nasıl hesaplandığını anlatması beklenmektedir. Öğrencilerle tartışarak, robotun köşelerde iç açı kadar mı yoksa dış açı kadar mı dönmesi gerektiğine karar verilir.

Not

Robotun köşelerde çokgenin dış açısı kadar dönmesi gerekmektedir. Bir çokgenin dış açısı,

$$\text{Dış açı} = 360^\circ / n$$

formülü kullanılarak hesaplanır. Burada, n çokgenin kenar sayısını temsil eder. Örneğin, bir üçgenin (n=3) dış açısı = $360^\circ / 3 = 120^\circ$, bir beşgenin (n=5) dış açısı = $360^\circ / 5 = 72^\circ$ dir.

Tanımlama: Bu süreçte öğrenciler, robotun yerine getirmesi gereken görevleri net bir şekilde tanımlamalı ve bu görevleri gerçekleştirmek için karşılaşılabilecekleri olası alt problemleri belirlemelidir. Öğrenciler, robotun gerçekleştirmesi istenen görevleri kalem ve kağıt kullanarak

yazılı hale getirmeli, bu görevlerin her birini detaylandırarak hangi adımları izlemeleri gerektiğini düşünmelidir. Böylece, robotun işlevselliğini artırmak ve görevleri daha etkili bir şekilde yerine getirmek için gerekli planlamayı yapmış olurlar.

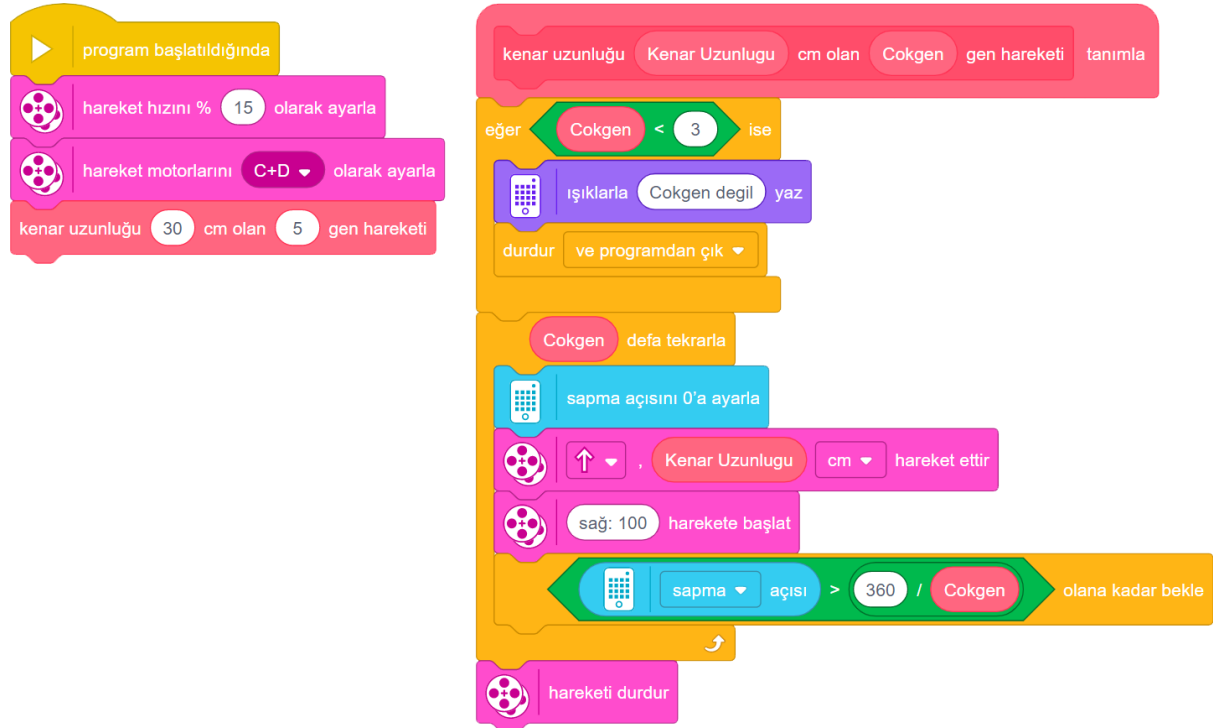
Örneğin:

- Yeni bir blok tanımlanacaktır.
- Bloğun iki tane sayısal değer girişi olmalıdır.
- Robotun her bir köşede döneceği açı hesaplanmalıdır.
- Çokgenin çizimi için hareket, kenar sayısı kadar tekrar edilmelidir.
- Girilen değer çokgen değilse (örneğin 2 veya negatif değer girilmişse) hata mesajı verilmeli.

Fikir üretme: Öğrenciler, gün boyunca edindikleri bilgileri kullanarak kendi bloklarını oluşturma sürecine aktif olarak katılırlar ve hangi program bloklarını kullanacaklarına dair yaratıcı fikirler üretirler. Geliştirdikleri bu fikirler üzerinde düşünerek, sonuca ulaşabileceklerini düşündükleri noktaya kadar muhtemel çözüm önerilerini tartışırlar. Bu tartışmalar sırasında, her bir önerinin avantajlarını ve dezavantajlarını değerlendirirler. Nihayetinde, en etkili ve uygulanabilir çözümleri belirlemek amacıyla bu fikirleri maddeler halinde listeleyerek, sistematik bir yaklaşım geliştirmiş olurlar.

ÜRET

Probleme yönelik tanımlamaları ve çözüm önerilerini geliştirdikten sonra, öğrenciler bilgisayar ve robot ile çalışarak istenilen program kodunu oluşturma sürecine geçerler. Öğrencilerin, aşağıdaki Resim 10.9'da görülen programa benzer bir çözüm üretmeleri teşvik edilir.



Resim 10.9 Çokgen Çizdiren Blok Örnek Programı

Not

Öğrencilerin programı planlanandan daha erken tamamlamaları durumunda, onlardan bloğa dönüş yönü eklemeleri istenebilir. Örneğin, çokgenin saat yönünde veya ters yönde çizilmesi için yeni bir giriş eklenebilir ve programda gerekli düzenlemeler yapılabilir. Ayrıca, kenar uzunluğunun negatif bir değer olarak girilmesi durumunda robotun hareketinin nasıl etkileneceği tartışılarak bu senaryo üzerinde düzenlemeler yapılabilir.

Rehber öğretmenler Tasarla Üret bölümlerinde verilen kod örneklerini ve ekran görüntülerini öğrenciler ile **paylaşmamalıdır**. Tasarla Üret bölümündeki paylaşımlar rehber öğretmenlerin etkinliklere hazırlanmasında kolaylık sağlamak içindir. Öğrenciler Tasarla Üret bölümlerinde sorulan problemlere kendileri çözüm üretmelidir. Rehber öğretmenler öğrencilere problemleri çözebilmeleri için yeterince zaman tanımalı ve direk çözümü vermeden rehberlik sağlamalıdır.

DEĞERLENDİR

Günün sonunda rehber öğretmen öğrencilere aşağıdaki soruları yönelterek bu hafta içeriğinin birlikte özetlenmesini sağlamalıdır.

- Kendi bloklarınızı oluşturmak size ne gibi faydalar sağlayabilir?
- Gün içerisinde en çok zorlandığınız görev hangisiydi? Neden?
- Programlama arayüzünde olması gereken başka blok önerileriniz nelerdir?

Proje Hazırlıyorum- Fikir Üretme Aşaması

Proje hazırlama süreçlerinde öğrenciler çevrelerinden veya günlük hayatlarından uygun bir problem belirleyip bu probleme kendi tasarladıkları bir robot ile çözüm üretirler. Çözümün başarılı olabilmesi için ihtiyaçların iyi tanımlanmış olması ve bu ihtiyaçların giderilmesi için uygun çözüm önerilerinin sunulması önemlidir. Bu hafta, empati ve tanımlama adımıyla toplanan bilgiler bir bütün hâline getirilir ve çözüm önerisi taslak olarak ortaya konur. Farklı çözümlerin önerilmesi gereklidir. Bunlar çok detaylı ve tamamlanmış fikirler olmak zorunda değildir fakat öneriler gerçekçi ve üretilmeye uygun olmalıdır.

Gruplardan belirledikleri robot projesi için yaptıkları fikir üretme çalışmasının sonucunu sınıfta açıklamaları istenir. Açıklamaların aşağıdaki bilgileri içermesi gerektiği belirtilir.

- Problem cümlesi: Robotun ne yapması isteniyor?
- Robotun kullanılması planlanan ortamdaki ihtiyaçlar nelerdir?
- Robotun kullanılması planlanan ortamdaki beklentiler nelerdir?
- Hedeflerin gerçekleştirilmesi için yapılması gerekenler nelerdir?
- Robotun belirlenen hedefleri gerçekleştirebilmesi için tasarımı nasıl olmalıdır?
- Robotun belirlenen hedefleri gerçekleştirebilmesi için hangi işlemleri yapabilmesi gerekir?
- Problem için üretilen alternatif tasarım ve programlama çözümleri nelerdir?
- Tasarım ve programlama çözümünün seçim süreci nasıl gerçekleştirilmiştir?

- Seçilen programlama çözümünün şematik veya maddeler hâlinde gösterimi (algoritması) nasıldır?

Açıklanan projelerle ilgili olarak diğer öğrencilerin ve rehber öğretmenlerin fikirleri alınır.

Sonraki Haftaya Hazırlık: Gruplara, üretilen fikirlere bağlı olarak robotun tasarlanıp programlanması aşamasına geçileceği söylenir.

11. Hafta - Proje Hazırlıyoruz

Haftanın Amacı:

Dersin sonunda, öğrencilerin şimdiye kadar öğrendiklerini kullanarak seçecekleri bir gerçek hayat problemine çözümler üretilmeleri amaçlanmaktadır. Başka bir ifadeyle öğrenciler ders sonunda grup halinde farklı projeler üretecektir.

Haftanın Kazanımları:

- Proje ekibini oluşturur.
- Gerçek hayat problemlerini belirler.
- Yenilikçi fikirler üretir.
- Seçilen probleme tasarlanan robotlar ile çözüm üretir.
- Bir probleme yönelik çözüm önerilerini prototipe dönüştürür.

Öğrencilerin geçen haftalarda grup halinde başladıkları projelerini bu haftada devam etmesi beklenmektedir. Projelerin hazırlık, empati, tanımlama, fikir üretme aşamalarının üzerinden gidilerek *geliştirme* ve *test etme* sürecine başlanacaktır. Proje geliştirme aşamaları dinamik bir süreçler olmasından dolayı tekrarlanabilir ve fikirler, çözümler revize edilebilir. Bu süreçlerde rehber öğretmenler gözetiminde öğrencilere ders dışı etkinliklerle de destek olunması önerilmektedir.

GELİŞTİRME – TEST ETME

Etkinliğin Amacı: Öğrenciler tanımladıkları problemi çözmek için kendi tasarladıkları bir robotu geliştirir ve test ederler.

Prototip Geliştirme: Grupların belirledikleri robot projesi için yaptıkları fikir üretme çalışması sonucunda üretilmesine karar verdikleri robotu oluşturup programlamaya başlamaları gerekir.

Graplardan, fikir üretme aşamasında belirledikleri çözümleri etkinlik sonuna kadar hızlıca uygulamaları, robot tasarımı ve programı geliştirmeleri istenir. Çözüm için geliştirilen robotlarda ve programlarda bazı aksaklıkların/problemlerin olması olağandır. Planlanan robotun ilk denemede oluşturulması ve geliştirilen programın sorunsuz çalışması çok karşılaşılan bir durum değildir. Bu tür sorunların ortaya çıkmasının olağan olduğu rehber öğretmen tarafından öğrencilere söylenmeli ve/veya hissettirilmelidir. Ortaya çıkan sorunların çözümüne yönelik bir plan hazırlanıp uygulanmalıdır. Çözüm için gruplara aşağıdakilere benzer bir süreç önerilebilir:

- Problemler maddeler hâlinde listelenmeli (beklenen robot davranışı ile gerçekleşen arasındaki farklar),
- Problemler arasındaki ilişkiler belirlenmeli,
- Öncelik sırası belirlenmeli,
- Öncelik sırasına göre çözümler geliştirilmeli (Çözüm için hipotezler oluşturulmalı),
- Geliştirilen çözümler (hipotezler) denenmeli ve
- Çözümlerin her koşulda istenen şekilde çalışması için gerekli düzenlemeler yapılmalıdır.

Bütün grupların, hazırladıkları prototipleri (tam olarak çalışmasa bile) rehber öğretmene göstermesi sağlanır. Böylece rehber öğretmenin, geliştirilen ürünü test etmesi/görmesi, çeşitli çözüm önerileri getirmesi ve/veya gözden kaçan noktaları ortaya çıkarması sağlanabilir.

Test Etme: Ortaya çıkan son ürünü hedef kitle ile/hedef ortamda değerlendirmek geliştirilen çözümlerin kusursuz bir şekilde çalışması için önemlidir. Hedef kitleye sınıfta ulaşmak her zaman mümkün olmayabilir. Bu nedenle grupların, ortaya çıkardıkları ürünleri diğer öğrencilerin kendi başlarına deneyimlemelerine izin vermesi, gözden kaçan unsurların ortaya çıkarılmasına yardımcı olabilir. Projelerin test aşaması dersin son haftasına kadar devam eder.

12. Hafta - Proje Hazırlıyoruz

Haftanın Amacı:

Bu hafta, öğrenciler hazırladıkları projeleri tamamlayıp, sunum yapmaları beklenmektedir. Geçen hafta başlanan test aşamalarına bu hafta da devam ederek eksikliklerin giderilmesi beklenmektedir.

Haftanın Kazanımları:

- Bir probleme yönelik çözüm önerilerini prototipe dönüştürür.
- Prototipe çalışmasını test eder.
- Projenin aşamalarını sunar.

Öğrencilerin geçen haftalarda grup halinde başladıkları projelerini bu hafta tamamlaması beklenmektedir. Projelerin tamamlanarak sunum yapılması önerilmektedir.

SUNUM

Projenin son aşamasında gruplara projelerini sunmaları konusunda yardımcı olunmalıdır. Gruplar projelerini bir senaryo çerçevesinde tiyatral olarak sunabilir. Projeler değerlendirilirken süreç bazlı değerlendirme tercih edilmelidir. Özellikle dersin son 4 haftası boyunca proje geliştirme aşamaları dikkate alınabilir. Ayrıca proje değerlendirme için aşağıda verilen örnekteki gibi dereceli puanlama anahtarları (rubrik) de kullanılabilir.

Proje Dereceli Puanlama Anahtarı

Grup İsmi:

		Mükemmel	İyi	Orta	Zayıf	Kötü					
Mekanik Tasarım	Dayanıklılık: Robot yapısal bütünlüğü koruyacak sağlamlıkta tasarlanmıştır.	10	9	8	7	6	5	4	3	2	1
	Mekanik Verimlik: Robotun tasarımı basit ve kolaylıkla düzenlenebilir.	10	9	8	7	6	5	4	3	2	1
	Mekanizasyon: Robot amaçlanan görevler için uygun hız, güç ve doğrulukla hareket etmek üzere tasarlanmıştır.	10	9	8	7	6	5	4	3	2	1
Programlama	Programlama Kalitesi: Program hedefe uygun olarak tutarlı sonuçlar üretmiştir.	10	9	8	7	6	5	4	3	2	1
	Programlama Verimliliği: Program sistem kaynaklarını en az kullanarak en hızlı sonucu üretmiştir.	10	9	8	7	6	5	4	3	2	1
	Otomasyon/Navigasyon: Program dışardan müdahaleye gereksinim duymadan otonom şekilde görevini yerine getirmiştir.	10	9	8	7	6	5	4	3	2	1
Sunum	Yenilikçi Sunum: Yaratıcı sunum teknik ve yöntemleri kullanılmıştır.	10	9	8	7	6	5	4	3	2	1
	Etkin Sunum: Proje hedef kitlenin en hızlı ve kolay anlayacağı şekilde sunulmuştur.	10	9	8	7	6	5	4	3	2	1
	Sunum İçeriği: Proje geliştirme adımları detaylı bir şekilde anlatılmıştır.	10	9	8	7	6	5	4	3	2	1
Yenilikçi Yaklaşım	Tasarım: Proje özgün mekanik tasarım özellikleri içermektedir.	10	9	8	7	6	5	4	3	2	1
	Görev Stratejisi: Proje yenilikçi programlama yöntemleri içermektedir.	10	9	8	7	6	5	4	3	2	1
	Yenilikçi: Proje gerçek hayat problem(ler)ine farklı çözüm(ler) getirmektedir.	10	9	8	7	6	5	4	3	2	1

Toplam: