O3 – Öğretim Desteği İçeriği

ÖĞRETMEN HEDEFLİ OYUN TASARIMINA DAYALI ÖĞRENME SENARYOLARI





Fikri Çıktı: O3/A1 - Öğretmen hedefli oyun tasarımına dayalı öğrenme senaryoları
Fikri Çıktı No - Başlık: O3 – Öğretim Desteği İçeriği
Fikri Çıktı Lideri: South-West University "Neofit Rilski" (Bulgaristan)
Dâhil Olan Ortaklar: University of Ljubljana (Slovenya), University of Rijeka (Hırvatistan)

Feragatname

Bu proje Avrupa Birliği Erasmus + Programı tarafından finanse edilmiştir.

Bu yayında belirtilen bilgi ve görüşler yazar(lar)a aittir ve Avrupa Birliği'nin resmi görüşünü yansıtmayabilir. Avrupa Birliği kurumları ve kuruluşları veya onların adına hareket eden herhangi bir kişi burada yer alan bilgilerin kullanımından sorumlu tutulamaz.

Tüm hakları saklıdır. Kaynağın belirtilmesi kaydıyla ticari amaçlar dışında çoğaltılmasına izin verilir.

Copyright © Coding4Girls, 2018-2020



Creative Commons Attribution-ShareAlike 4.0 International Public License (<u>CC BY-SA 4.0</u>)



İÇİNDEKİLER

GiRiş	5
ÖĞRENMME SENARYOLARI	6
TEMEL ÖĞRENME SENARYOLARI	8
Öğrenme Senaryosu 1 - Snap! ile tanışın - Arayüz	8
Öğrenme Senaryosu 2 - Karakterinizi hayata döndürme zamanı	12
Öğrenme Senaryosu 3 - Sahnede hareket etmek	16
Öğrenme Senaryosu 4 - Kostüm değiştirme ve döndürmeler	22
Öğrenme Senaryosu 5 - Çiftliğin sesi	29
Öğrenme Senaryosu 6 - Bukalemunun yaz tatili, basit sürüm	37
Öğrenme Senaryosu 7 - Prense ve prensese evcil hayvanlarını bulmada yardım etme 🥠	47
Öğrenme Senaryosu 8 - Tebeşir ile çizim yapmak	54
Öğrenme Senaryosu 9 - Çöpleri toplamak ve parkı temizlemek	65
Öğrenme Senaryosu 10 - Kedileri beslemek	73
Öğrenme Senaryosu 11 - Bir barınaktaki kedi sayısını tahmin etmek	81
İleri Seviye Öğrenme Senaryoları	89
Öğrenme Senaryosu 12 - Sağlıklı yiyecekler yakalamak	89
Öğrenme Senaryosu 13 - Hikâye anlatma	99
Öğrenme Senaryosu 14 - Çizim yapma 12	10
Öğrenme Senaryosu 15 - Fareyi yakalamak 11	18
Öğrenme Senaryosu 16 - Piknik için yiyecek satın almak 12	26
Öğrenme Senaryosu 17 – Operasyonlar/İşlemler 13	35
Öğrenme Senaryosu 18 – Geri dönüşüm 14	42
Öğrenme Senaryosu 19.1 - Piyano çalma 14	48

Co-funded by the Erasmus+ Programme of the European Union





Öğrenme Senaryosu 19.2 - Piyano çalma	152
Öğrenme Senaryosu 20 - Test	162
Referanslar	174





Geçen yüzyılın önde gelen psikologları, oyun oynamayı yaşa veya gelişim aşamasına bakılmaksızın yaşam becerilerinin geliştirilmesi için en önemli etkinliklerden biri olarak tanımladılar. Oyun boyunca çocuk, yeni koşullara hızlı bir şekilde adapte olarak değişiklikleri kolaylıkla ele alır. Oyun oynarken, kelimeden temel kavramları keşfeder ve aralarındaki ilk temel ilişkiler kurulur.

Günümüzde oyunlar, bir çocuğun gelişiminin ilk aşamalarında evde ve anaokulunda daha yaygın olarak kullanılmaktadır. Okulda öğrenme, çoğu zaman pasif öğrencilerle öğretmen merkezli bir modelde geleneksel bilgi aktarımına dayanmaktadır. Öte yandan, geçen yüzyılda geliştirilen öğrenme kuramları öğrenci merkezli, probleme dayalı, daha yüksek taksonomik seviyelerde, motivasyonel ve genellikle BİT tarafından desteklenen yeni öğretim ve öğrenim yaklaşımlarını teşvik ettiler.

CODING4GIRLS Projesi "düşük giriş yüksek tavan" yaklaşımından yola çıkarak daha az bilgi gereksinimi olan ve daha ileri düzeydeki öğrenciler için problem çözme zorluklarını sınırlamayan programlama etkinliklerini içerir. Öğrenciler, eksik kod yapı taşlarını ekleyerek veya kendi çözümlerini oluşturarak kısmen tamamlanmış olan çözümleri bitirmeye teşvik edilecektir. Aktiviteler, tek bir programlama konseptine sahip temel olanlardan çoklu programlama konseptleri ile daha ileri seviyeye kadar sırayla planlanır. Proje ekibi, Snap!'de öğrenme etkinlikleri hazırlarken kızların tercih ettiği oyunların belirlenen özellikleri ve gerçek dünya sorunları ile ilgili etkinliklerden yola çıktı.

Hazırlanan öğrenme senaryoları, düşünme öğrenme yöntemlerini ve önerilen ciddi oyunları derslerine entegre etmelerine yardımcı olacaktır. CODING4GIRLS, aktif ve oyun tabanlı öğrenme tasarımına göre hazırlanmıştır. Ayrıca hem kız hem de erkeklerin programlama becerilerini geliştirmek amacıyla üretilecek her bir öğrenme aktivitesi için bilgiler içerir. Öğrenme senaryolarında aşağıdaki bilgiler mevcuttur:

- İlgili öğrenme faaliyetinin genel eğitim hedefi
- Öğrenme faaliyetinin kapsadığı kavramlar
- Belirli öğrenme hedefleri
- Beklenen öğrenme çıktıları
- CODING4GIRLS oyun tasarımına dayalı öğrenme yaklaşımının adım adım kullanımı
- Geliştirilen bilgiyi değerlendirme yöntemleri
- Sınıf işbirliği bağlamında öğrenciler arasında münazara başlatmak için sorular

Öğrenme faaliyetlerine karşılık gelen 21 öğrenme senaryosu hazırlanmıştır. Öğretmenler senaryoları ve oyunları önerilen sırayla kullanabilir veya tercihlerine ve ihtiyaçlarına göre özgürce seçebilirler. Öğrenme senaryoları, kullanıcı etkileşim süreçleri, geri bildirim oluşturma dâhil önerilen ciddi oyunun genel işlevselliğini ve önerilen ciddi oyunda uygulanacak tüm öğrenme etkinliklerinin açıklamalarını içerir.

Öğrenme senaryoları İngilizcenin yanı sıra proje ortaklarının ulusal dillerinde de (Türkçe, Bulgarca, Hırvatça, Yunanca, İtalyanca, Portekizce ve Slovence) mevcuttur.





ÖĞRENME SENARYOLARI

Hazırlanan öğrenme senaryoları, temel bir programlama konseptinden çoklu programlama konseptiyle daha ileri seviyeye kadar devam eder. Aşağıdaki tablo önerilen faaliyet sırasını göstermektedir.

	TEMEL ÖĞRENME SENARYOLARI	GELİŞTİREN ORTAK
1	Snap! ile tanışın – Ara yüz	
1	Snap!'i yakından tanıyın - Görsel programlama ortamı	UL
	Karakterinizi canlandırmanın vakti geldi	
2	Programlama bloklarını bulmak, bağlamak, hareket ettirmek, karakter	UL
	oluşturmak ve konuşturmak	
3	Sahnede hareket etmek	
5	Blokları anlamlı bir sıraya sokmak	01
4	Kostüm değiştirme ve döndürmeler	UL
5	Çiftliğin sesi	111
5	Ses ekleme, içeri aktarmak, kayıt etmek ve ses çıkarmak	01
	Bukalemunun yaz tatili, basit versiyon	
6	Olaylara aşina olma, renk algılama, Boolean değerleri, iki farklı oyun	UL
	durumunu kontrol etme ve bunlara yanıt verme	
7	Prense ve prensese evcil hayvanlarını bulmada yardım etme	UI
,	Koşulları kullanma ve çizme	01
8	Tebeşir ile çizim yapmak	UI
	Döngü kullanmak, çevirmek, arka planı değiştirmek	01
٩	Çöpleri toplamak ve parkı temizlemek	UI
	Değişkenlere aşina olmak, karakterleri çoğaltmak, kod blokları	
	Kedileri beslemek	
10	Değişkenler (döngünün içinde/dışında), döngüler, rastgele sayılar, dizi	UL
	birleştirme, 4 işlem, girdi kullanma	
	Bir barınaktaki kedi sayısını tahmin etmek	
11	Rastgele değerler, değişken girişi, koşul ifadeleri, karşılaştırma işlemleri,	UL
	sayaç kullanma	CELISTIDEN
ILERIS		ORTAK
12	Sağlıklı yiyecekler yakalamak	UL
	Degişkenler, koşullar, döngü, nokta yönü, rastgele kullanma	
13	Hikâye anlatma	SWU
14	Çizim yapma	UNIRI
15	Fareyi yakalamak	UL
	Döngüler, koşullu ifadeler, değişkenler kullanma	
16	Piknik için yiyecek satın almak	UL
	Değişkenler, koşul ifadeleri, 4 işlemi kullanma	
17	Operasyonlar	SWU
18	Geri dönüşüm	SWU
19.1	Piyano çalma 1	SWU
19.2	Piyano çalma 2	UNIRI





20	Test	SWU
	Basitleştirilmiş PACMAN oyunu	
21	Olay tabanlı nesne hareketi, renk algılama, Boole değerlerini kullanma, iki	UL
	farklı oyun durumunu kontrol etme ve bunlara yanıt verme	





TEMEL ÖĞRENME SENARYOLARI

Öğrenme Senaryosu 1 - Snap! ile tanışın - Arayüz

Öğrenme Senaryosu	Snap! İle tanışın – Ara yüz
Adı	
Geçmiş Programlama	/
Deneyimi	
Öğrenme Çıktıları	Genel Öğrenme Çıktıları:
	 Snap! İle tanışın. Görsel öğrenme ortamı.
	Özel Öğrenme Çıktıları:
	 Öğrenci yeni karakter (Sprite – Kukla) oluşturabilir Öğrenci karakterine kostüm ekleyip değişiklikler yapabilir. Öğrenci karakteri merkeze koyarak düzgün bir şekilde döndürebilir. Öğrenci sahneye yeni bir arka plan ekleyebilir ve düzenleyebilir.
Amaç, Görevler ve	Öğrenci yeni bir karakter (Sprite – Kukla) ekler. Karaktere bir kostüm
Faaliyetlerin Kısa	ekler. Kostümü düzenler ve bunlardan birini siler. Öğrenci sahne için
Tanıtımı	yeni bir arka plan oluşturur, düzenler ve istenmeyenleri siler.
	Amaç: Bir saatin sonunda öğrenciler, bir oyunda kullanmak için en
	sevdikleri karakteri ve gerçek ya da hayali yaşam ortamını
	çizeceklerdir.
	Etkinliği tüm öğrenciler için daha motive edici kılmak amacıyla,
	karakter çizimlerinin bu hedef kitleye uygun olduğu bilimsel
	çalışmalarda tespit edilmiştir.
Faaliyetin Süresi	45 dk.
Öğrenme ve Öğretme	Öğretmen Gösterimi
Strateji ve Metotları	Bireysel Çalışma
Öğretme Formları	Ön Çalışma
	Bireysel Çalışma





Öğretme Özeti	(Motivasyon-Giriş, Uygulama, Düşünme ve Değerlendirme)
	Bir saatin sonunda öğrenciler, bir oyunda kullanmak için en sevdikleri
	karakteri ve gerçek ya da hayali yaşam ortamını çizecekler.
	[Adım 1]
	Öğrencilere Snap!'i bulabilecekleri web sayfasını gösterin!
	(https://snap.berkeley.edu/). Onlara ara yüzün farklı bölümlerini
	gösterin. Bloklu bölüm, senaryoları bir araya getirebilecekleri /
	kostümleri değiştirebilecekleri / sesleri ekleyebilecekleri bölüm,
	üzerinde karakter (Sprite – Kukla) bulunan sahne, karakter listesi.
	[Adım 2]
	Aşağıdaki resimde işaretlenen düğmelerden birini tıklayarak yeni bir
	karakter oluşturabilirsiniz.
	>
	2 Martin
	Yeni bir karakter çizmeyi deneyeceksiniz. Boya fırçasına tıkladığınızda,





karakterinizi Microsoft Paint Uygulamasındakine benzer şekilde çizebileceğiniz bir açılır pencere açılır. Öğrenciler için görev: İlk karakterinizi çiziniz. 10 dakikanız var. Hareketli grafik çizildikten sonra, hareketli grafiğin dönüş merkezinin olmasını istediğiniz yerde olduğundan emin olmalısınız. Ð simgesini kullanabilirsiniz. Bunun icin Öğrenciler için görev: Karakterinizi merkeze koyun. [Adım 3] Karakterinizi düzenlemek için yalnızca hareketli grafiğiniz tıklandığında görünen Kostümler sekmesini seçin. Düzenlemek istediğiniz bir kostüme sağ tıklayarak düzenlemeyi seçin. Ayrıca kostümünüzü aynı menüden kopyalayabilir veya silebilirsiniz. untitled ⇔ Sprite(2) √ draggable Sounds re from a edit rename duplicate delete export [Adım 4] Halihazırda var olan bir kostümü içe aktarmak için, üzerine bir parça kağıt çizilmiş simgeye tıklayın ve Kostümler'i seçin....





Co-funded by the Erasmus+ Programme of the European Union



Yine, bu seçenek yalnızca karakteriniz sahnede tıklandığında gösterilecektir.

Öğrenciler için görev: Bir kostüm seçerek karaktere ekleyin.

[Adım 5]

Artık karakteriniz var ve sahneye biraz arka plan eklemelisiniz. Bunu yapmak için önce sahnedeki karakterin yerine *Sahne Alanı (stage*)'na tıklayın.



Öğrenciler için görev: Kendi arka planınızı oluşturun.

Öğrenciler için görev: Mevcut arka planlar arasında arama yapın ve bunlardan birini içe aktarmaya ekleyin. Böylece iki tane elde edersiniz. Öğrenciler için görev: Arka planınızı düzenlemenin bir yolunu bulun. Arka planlarınızdan birini silmenin bir yolunu bulun. Böylece geriye yalnızca biri kalır.





	Düşünme ve değerlendirme:
	Öğrenciler karakterlerini ve çevrelerini yaşadığı yerde çizmeyi
	başardılar mı? Herhangi bir sorunları oldu mu? Onları nasıl çözdüler?
Öğretmenler İçin Araçlar ve Kaynaklar	https://snap.berkeley.edu/
Öğrenciler İçin Kaynaklar ve Materyaller	Öğrenci için talimatlar (C4G1_InstructionsFor Student.docx)

Öğrenme Senaryosu 2 - Karakterinizi hayata döndürme zamanı

Öğrenme Senaryo	Karakterinizi hayata döndürme zamanı
Başlığı	
Önceki programlama	/
deneyimi	
Öğrenme Çıktıları	Genel öğrenme sonuçları:
	 Öğrenciler programlama bloklarını nerede bulacağını ve bunları bir sıraya nasıl bağlayacaklarını bilir. Öğrenciler bir karakterin (Sprite – Kukla) nasıl taşınacağını bilir. Öğrenciler karakterine bir şey söyletmeyi bilir. Algoritmik düşünme odaklı özel öğrenme çıktıları: Anlamlı bir blok dizisi oluşturma
Amaç, Görev ve	Öğrenciler programlama bloklarının nerede depolandığını ve uygun
Etkinliklerin Kısa	olanları nasıl bulacağını, hangi blok kategorilerinde olduğunu ve
Tanımı	blokların bir sıraya nasıl bağlanacağını öğrenir.
Faaliyet Süresi	45 Dakika
Öğrenme ve Öğretme	Öğretmen sunumu
Stratejisi ve Yöntemleri	Bireysel çalışma
Öğretim Formları	Ön çalışma
	Bireysel çalışma





Öğretim özeti	(Motivasyon-Giriş, Uygulama, Düşünme ve Değerlendirme)
	Bu derste karakterinizi hareket ettirecek ve bir şeyler söyleteceksiniz.
	Onlara bu derste programlayacakları bir programın örneğini
	gösterebilirsiniz.
	[Adım 1]
	Önce kullanabileceğiniz programlama bloklarının nerede olduğuna
	bakalım. Neredeler?
	Sol tarafta, blokların farklı kategorilerini bulabilirsiniz. Hareket,
	Görünüm, Sesler, Kalem, Kontrol, Algılama, İşlemler ve Değişkenler.
	İlk kullanacağız blok. 🚾 🗢 steps
	Öğrenciler için görev: Önce bloğu bulun ve sonra üzerine çift tıklayın.
	Ne oldu?
	[Adım 2]
	Bloğu bir programa bağlamaya başlamak için "move steps"
	bloğunu kod alanı kısmına taşıyın.
	Motion Control Looks Sensing Sound Operators Pen Variables Scripts Costumes Sounds
	turn 👌 15 degrees + move 10 steps
	Kodu çalıştırmak için <i>kod alanı/scripts</i> sekmesinin içindeki bloğu çift
	tıklatabilirsiniz.
	[Adım 3]
	Snap'teki programlar genellikle yeşil bayrağa tıklayarak başlar.
	Öğrenciler için görev: Farklı kategori türlerine tıklayın ve yeşil bayrak
	tıklanırsa programı başlatan bir blok bulmaya çalışın.
	Çözüm:







[Adım 6]

Her iki *söyleme (say)* komutunu *görünümler (Look)* kategorisinde bulabilirsiniz. Temel fark, **say Helol** programı kod devam etmeden önce, ___ saniye beklemesini söylememesi veya herhangi bir zamanda söylemeyi bırakmasıdır.





	[Adım 7]
	Bir önceki derste kullandığınız karakterinizi (Sprite – Kukla) alın.
	Sahnede sürükleyerek sahnenin sol tarafına taşıyın ve bir program
	yazmak için bu bloğu 🚾 steps kulanın. Resimdeki blok
	sahnenin sağ tarafından sol tarafına hareket yapar. Unutmayın! Her
	hareketten sonra karakter bir şey söylemeli ve tek hamleden fazlasını
	yapmalı.
	Deneyin: Programınız her çalıştırıldığında karakter tam olarak aynı
	pozisyonda mı durdu? Karakterinizin her zaman aynı konumdan
	başlayıp sahnede durmasını sağlayacak bir blok bulabilir misiniz?
	Öğretmen İçin İpucu: Eğer karakter sahneden kayboluyorsa, sağ fare
	tuşu ile üzerine tıklayarak göster seçip sahneye geri çağırabilirsiniz.
	Bu blok 📴 to 🗷 🔍 y: 💽 hangi x ve y konumunda olduğunu
	belirlemek için kullanılır. Karakterinizi sahne üzerinde olmasını
	istediğiniz noktaya taşımanızı ve karakterin konumunu x ile y
	konumuna tıklayarak (blokların Hareket kategorisinin alt kısmında) bu
	blok sayesinde gerçekleştirebilirsiniz.
	Düşünme ve Değerlendirme:
	Görevi tamamlamak için karakterinizin kaç kez hareketi tekrarlayıp
	diziyi tekrarlaması gerekti? Sayı sınıftaki herkes için aynı mı? Neden?
Öğretmen için Araçlar	Örnek program:
ve Kaynaklar	https://snap.berkeley.edu/snap/snap.html#present:Username=spelac
	<u>&ProjectName=C4G dog goes home</u>
Öğrenciler için	 Öğrenci için talimatlar (C4G2_InstructionsForStudent.docx)
kaynaklar/materyaller	- Öğrenci kendi karakterini ve arka planını oluşturmak isterse
	aşağıdaki linki kullanabilir:
	https://snap.berkeley.edu/snap/snap.html#present:Username=spelac
	&ProjectName=C4G dog goes home tmp





Öğrenme Senaryosu 3 - Sahnede hareket etmek

Öğrenme Senaryosu	Sahnede hareket etmek
Adı	
Geçmiş Programlama	Öğrenci, programlama bloklarını nerede bulacağını ve bunları bir
Deneyimi	dıziye nasil bağlayacağını bilir.
Öğrenme Çıktıları	Genel Öğrenme Çıktıları:
	 Anlamlı bir blok dizisi oluşturma
	Algoritmik düşünmeye yönelik Özgül Öğrenme Çıktıları:
	 Öğrenci, karakteri sahnede konumlandırır.
	 Öğrenci, karakterin x ve y konumunu değiştirir. Öğrenci y döngüçünü tekror kullanır.
	 Öğrenci, karakteri hareket halindeki hareketinin yönünün ve
	adımlarının karakterin döndüğü yöne bağlı olduğunu öğrenir.
Amaç, Görevler ve	Kısa açıklama: Öğrenci sahnede karakteri x ve y yönünde hareket
Faaliyetlerin Kısa	ettirmeyi öğrenir. Verilen görevleri çözmek için kolay bir program
Tanıtımı	programlar. Karakteri farklı bir yöne nasıl çevireceğini ve bunun
	move_steps (adım hareket et) bloğunu nasıl etkilediğin öğrenir.
	Görevler: Hareketli grafiği x yönünde hareket ettiren bir program
	oluşturun. Hareketli grafiği y yönünde hareket ettiren bir program
	oluşturun. X ve y yönlerinde hareketi birleştiren bir program
	oluşturun.
	Amaçlar: sahnede x ve y yönündeki hareketleri ayırt etmek ve tekrar
	döngüsünü kullanmak.
Faaliyetin Süresi	45 dakika
Öğrenme ve Öğretme	Öğretmen gösterimi (demonstrasyon)
Strateji ve Metotları	Bireysel çalışma
Öğretme Formları	Ön çalışma
	Bireysel Çalışma



Γ



Öğretme Özeti	(Motivasyon-Giriş, Uygulama, Düşünme ve Değerlendirme)
	Farklı hayvanların hedeflerine ulaşmalarına yardım edeceksiniz. Bunu
	yapmak için onlara sahnede nasıl hareket edecekleri konusunda
	talimat vermeniz gerekecek.
	[Görev 1]
	"Topu yakalayın"ı açın ve köpeğin topu yakalaması için kod ekleyin.
	change x by 🔵 ve wait 🔿 secs blokları kullanarak köpeğin topa
	doğru koştuğu bir animasyon yapın.
	Görevin muhtemel çözümü:





Co-funded by the Erasmus+ Programme of the European Union



Gördüğünüz gibi, sola veya sağa hareket ettiğinizde x değişir. X = 0 ise, karakteriniz sahnenin ortasındadır. Ortanın sağında, x değerleri 0'dan büyük sayılardır.

ipucu: Bu aktivite ondalık sayılar konusunu bilen daha büyük yaştaki öğrencilerle yapılırsa bekleme süresi daha kısa olabilir. Örneğin 0.1. koordinat sisteminin ne olduğunu bilirlerse bazı açıklamaların yapılmasına gerek olmayabilir.

[Görev 2]

Maymunun ağaca tırmanmasına yardım et ve muzları getirmesi için

ve

maymuna kod ekle.

change y by

wait osecs blokları





kullanarak maymunun palm ağacına tırmanmasını sağla. Bu görevin muhtemel çözümü: when 🚬 clicked go to x: 0 y: -120 wait 🕦 secs change y by (10) vait 🚺 secs change y by 10 wait 🕦 secs change y by (10 vait 1 change y by 📶 wait 🕦 se change y by 1 wait 1 secs change y by (10) wait 🕦 secs change y by 10 wait 1 secs change y by (10) wait 1 change y by (10) wait 🚹 secs change y by 10 wait 🕦 secs change y by 1 Gördüğünüz gibi yukarı veya aşağı hareket ettiğinizde y değişiyor. Y 0 ise karakteriniz sahnenin ortasındadır. Ortadan yüksek olanların tümünde y değerinden büyüktür. Karakterinizin sahnede orta çizginin altında olmasını istiyorsanız sanki dalışa gidiyorsunuz gibi düşünebilirsin. Sayının önüne – (eksi işareti) koyun ve suyun kaç metre altında olduğunuzu yani sahnede çizgiden kaç adım geride olduğunuz belirtin. change y by -10 Eğer ağaçtan inmek istiyorsanız bloğunu kullanabilirsiniz.





İpucu: Bu aktivite ondalık sayılar konusunu bilen daha büyük yaştaki öğrencilerle yapılırsa bekleme süresi daha kısa olabilir. örneğin 0.1. koordinat sisteminin ne olduğunu bilirlerse bazı açıklamaların yapılmasına gerek olmayabilir.

[Adım 3]

Her iki görevde de birbirinin yerine iki blok kullanmak zorundaydınız. Kodu kaç defa tekrarlamak zorunda kaldınız?

Bilgisayara kodunuzu belirli sayıda tekrar etmesini söyleyerek bu kodu yazmayabilirsiniz. Bu *repeat (tekrar)* döngüdür. Aynı eylem veya bir eylem dizisi kendini birden fazla kez tekrarladığında bunu kullanabilirsiniz.

İki görev içinde kodunuzu değiştirin. Yani



döngüsünü kullanın. Tekrarlamak istediğiniz kod bu bloğun içine konulmalı ve boş alana kaç defa tekrarlanması gerektiğini yazmalısınız.

Köpek için kod

when clicked go to x: (150) y: (80) repeat (13) wait (1) secs change x by (20)

Maymun için kod



Görev: Köpeğin topa doğru koşmasını ve geri dönmesini sağlamaya





	calısın.				
	Görov: Maymunun ağaca tırmanmasını vo asağı inmosini sağlamaya				
	Gorev: Maymunun agaca tirmanmasını ve aşagi inmesini sagiamaya				
	çalışın.				
	En çok neyi sevdin? Pozisyonlama için X-Y grid arka planını kullanarak				
	hareketli grafiğin x ve y konumu konusunda yardım alabilirsiniz.				
		Y	(X:0,Y:180)		
		00.			
		3.4			
	(X:-240,Y:0)		(X:0,Y:0)	(X:240,1	(:0)
	200 .21	00	100	200	×
		.200			
			(X:0,Y:-180)		
Öğretmenler İçin	Topu yakala	amanın çözümü			
Araçlar ve Kaynaklar	https://sna	https://snap.berkeley.edu/snap/snap.html#present:Username			
	<u>=spelac⪻</u>	ojectName=C4G	moving x		
	Maymunu a	ağaca tırmandırn	nanın çözümü		
	https://sna	https://snap.berkeley.edu/snap/snap.html#present:Username			
	=spelac⪻	ojectName=C4G	moving y		
Öğrenciler İçin	Topu yakala	3			
Kaynaklar ve	https://sna	https://snap.berkeley.edu/snap/snap.html#present:Username			
Materyaller	<u>=spelac⪻</u>	ojectName=C4G	Catch the ba	<u>II</u>	
	 Maymunu a 	Maymunu ağaca tırmandır.			
	https://sna	https://snap.berkeley.edu/snap/snap.html#present:Username			
	<u>=spelac⪻</u>	=spelac&ProjectName=C4G Help monkey climb the tree			
	 Öğrenci içir 	Öğrenci için talimatlar (C4G2 InstructionsForStudent.docx			





Öğrenme Senaryosu 4 - Kostüm değiştirme ve döndürmeler

Öğrenme Senaryosu	Kostüm değiştirme ve döndürmeler			
Adı				
Geçmiş Programlama	Hareket			
Deneyimi				
Öğrenme Çıktıları	Genel Öğrenme Çıktıları:			
	Anlamlı bir blok dizisi oluşturma			
	Algoritmik düşünceye odaklı özel öğrenme çıktıları:			
	 Öğrenci, animasyon yapmak için karakterin kostümünü değiştirir 			
	 Öğrenciler karakterlerin dönüşünü değiştirir. 			
Amaç, Görevler ve	Kısa açıklama: Öğrenci, bir animasyon yapmak için karakterin			
Faaliyetlerin Kısa	kostümünü nasıl değiştireceğini öğrenir. Ayrıca, karakterin farklı			
Tanıtımı	dönüş türleri arasında nasıl değişiklik yapılacağını da öğrenir.			
	Görevler: Karakterin kostümünü değiştiren bir program oluşturun.			
	Her programda karakter için uygun dönüş türünü ayarlayın			
	Amaçlar: Karakterin kostümünü değiştirmeyi ve uygun dönüş türünün			
	nasıl ayarlanacağını öğrenin.			
Faaliyetin Süresi	45 dakika			
Öğrenme ve Öğretme	Demonstration			
Strateji ve Metotları	Bireysel çalışma			
Öğretme Formları	Ön çalışma			
	Bireysel çalışma			





Öğretme Özeti	(Motivasyon-Giriş, Uygulama, Düşünme ve değerlendirme)							
	Yürüyor,	dans	ediyormuş	gibi	görünmesi	için	bir	karakter
	animasyoi	nunu na	asıl yapacağın	ızı öğr	eneceksiniz.			
	[Adım 1]							
	Yeni bir b	oş proj	e açın. Beya	z bir k	âğıt parçasın	ia ben	zeyen	simgeye
	tıklayın ve	kostün	nler(costum	<i>es)</i> öğ	esini seçin.			
	Önce Bale	rin a'ya	a ardından İç	e Akta	r'a tıklayın. <i>E</i>	Balerin	b, ba	ılerin c ve
	<i>balerin d</i> i	çin de a	ıynısını yapın					
	Karakterir	izin <i>ko</i> s	s <i>tümler</i> sekm	nesind	e artık 4 bale	erin ko	stüm	ünüz var.
	Kostümler	sekme	sinin üzerind	leki m	etni değiştire	erek ka	irakte	ri <i>Balerin</i>
	olarak yer	iden ac	llandırabilirsi	niz:				
	Kostüm Se	ekmesi:						





Co-funded by the Erasmus+ Programme of the European Union







when clicked switch to costume ballerina a repeat 15 next costume wait 1 secs switch to costume ballerina a
[Adım 2] Balerinimiz her zaman aynı pozisyonda olmak istemediği için her kostümü değiştirdiğinde küçük hareketler yapıyor. Bu hareketi onun dansına ekleyin.
Olası çözüm: when clicked switch to costume ballerina a repeat 15 next costume move 10 steps wait 1 secs switch to costume ballerina a
[Adım 3] Yeni bir boş proje açın ve <i>Avery</i> (Snap!'te özel bir karaker/kukla/sprite) yürüyüş kostümlerini içe aktarın. <i>Avery</i> 'nin yürümesi için uygun bir arka plan ekleyin. <i>Avery</i> 'nin sahnenin sol tarafından sahnenin sağ tarafına yürüdüğü bir animasyon oluşturun. <i>Avery</i> 'nin bir şekilde nasıl canlandırılacağını, <i>Adım</i> 'larının gerçek hayatta olduğu gibi bağlantılı görüneceğini anlamaya çalışın Olası çözüm:
when clicked go to x: 220 y: 0 repeat 14 next costume move 30 steps wait 1 secs





[Adım 4]

Şimdiye kadar bir karakterin yalnızca bir yönde hareket ettiği bir program yazdınız. Bu görevde peynire ulaşmak için fareyi döndürmeniz gerekecek. Karakteri döndürmek için aşağıdakilerden birini seçebilirsiniz.

a)	point in direction 90 (90) right (-90) left (0) up (180) down random Buradan hangi yöne doğru dönmesi gerektiğini söyleyebilirsiniz.
b)	Belli bir acıyla saat yönüne turn (15) degrees ve saat
/	yönünün tersine dönmesini söyleyebilirsiniz.
Tam bi	ir daire 360 dereceye sahiptir. Bu nedenle şu an bulunduğunuz
yerden	ı ters yönde dönmek isterseniz 180 derece dönersiniz. Sola
dönme	ek isterseniz saat yönünün tersine 90 derece dönersiniz.
Sağınız	a dönmek isterseniz saat yönünde 90 derece dönüyorsunuz.
Bakınız	2:
<u>https:/</u>	//snap.berkeley.edu/snap/snap.html#present:Username=spelac
<u>&Proje</u>	ectName=C4G Find cheese.
Fareniı	n sadece yeşil alanda yürümesi gerekiyorsa peynire ulaşmak için
takip e	etmesi gereken bir program yazın. Fareyi gideceği yöne çevirin
ve	Adım bloğunu hareket ettirin. Farenin nasıl hareket ettiğini
görme	k için satırlar arasında 1 saniye bekleyin.
Çözüm	







Şimdi 90 derece çevirerek bir program yazmaya çalışın.

Çözüm:



[Adım5]

Gördüğünüz gibi fare peynire ulaşmak için farklı yönlere dönmüştür. Bazen karakterinizin baş aşağı dönmesini istemezsiniz. Ya da sadece sola veya sağa dönerek başının üstünde yürümesini istemezsiniz. Karakterinizin istediğiniz gibi döndüğünden emin olmak için solundaki uygun simgeye tıklamanız gerekir.



Dairesel ok karakterinizi istediğiniz her yöne (farede olduğu gibi) döndürmenize yarar.

<-> şeklindeki ok ise karakterinizin yalnızca sağa ya da yalnızca sola dönmesini sağlar (Köpeğin kafasının üzerinde yürümemesi için





	kullanacağın şey).			
	Son ok (->) hareketli grafiğin her zaman olduğu gibi görüneceği			
	anlamına gelir (bunu maymun için kullanabilirsiniz).			
	Köpek ve maymun için programlarınızı yeniden yazmaya çalışın. Bu			
	sefer önce nesneve gidip dönerek geri dönsünler. Döndürme stilini			
	doğru sekilde değiştirdiğinizden emin olun			
· · · · ·				
Oğretmenler Için	 Balerin programlama çözümü: 			
Araçlar ve Kaynaklar	https://snap.berkeley.edu/snap/snap.html#present:Username=spelac			
	&ProjectName=C4G dancing			
	• Avery:			
	https://snap.berkeley.edu/snap/snap.html#present:Username			
	<pre>=spelac&ProjectName=C4G Avery walking</pre>			
	Peyniri bulmanın çözümü:			
	https://snap.berkeley.edu/snap/snap.html#present:Username			
	=spelac&ProjectName=C4G Find cheese solution			
Öğrenciler İçin	Peyniri bul			
Kaynaklar ve	https://snap.berkeley.edu/snap/snap.html#present:Username			
Materyaller	<pre>=spelac&ProjectName=C4G Find cheese</pre>			
	 Öğrenci için talimatlar (C4G2_InstructionsForStudent.docx) 			





Öğrenme Senaryosu 5 - Çiftliğin sesi

Öğrenme Senaryosu	Çiftliğin sesi			
Adı				
Geçmiş Programlama	Öğrenci, bir arka plan ekleyebilir.			
Deneyimi	 Ogrenci, yeni bir karakter ekleyebilir. Öğrenci, karaktere nasıl bir şey söyletebileceğini bilir. 			
Öğrenme Çıktıları	Genel Öğrenme Çıktıları:			
	 Snap'ın medya kitaplığından ses ekleyin. Diğer ortamlardan ses aktarın. Yeni bir ses kaydedin. Bir tuşa basarak ses çalın. 			
	Algoritmik düşünceye odaklı özel öğrenme çıktıları:			
	 Öğrenci Snap'ın medya kitaplığından ses ekler ve belirli bir tuşa basıldığında sesi çalar, Öğrenci bilgisayardan sesi alır ve belirli bir tuşa basıldığında çalar, Öğrenci yeni bir ses kaydeder ve belirli bir tuşa basıldığında çalar. 			
Amaç, Görevler ve	Kısa açıklama: Oyuncunun belirli tuşlara basarak hayvanların seslerini			
Faaliyetlerin Kısa	öğrendiği basit oyun programı.			
Tanıtımı	Görevler: İlk Adım'da öğrenci sahne arka planını seçmelidir. Daha			
	sonra, öğrenci kadın çiftçiyi talimatları söylemesi için			
	programlamalıdır. 1) Köpeği duymak istiyorsanız, "D"! tuşuna tıklayın.			
	2) İneği duymak istiyorsanız, "C"! tuşuna tıklayın. 3) Koyunu duymak			
	istiyorsanız "S"! tuşuna tıklayın. 4) Domuzu duymak istiyorsanız "P"!			
	tuşuna tıklayın. 5) Atı duymak istiyorsanız "H"! tuşuna tıklayın. Bundan			
	sonra öğrenci, görevi kadın çiftçinin yönlendirdiği şekilde			
	programlamalıdır.			
	Amaç: Öğrencilere yeni bir sesin nasıl ekleneceği ve nasıl kullanılacağı			
	aniatilacaktir. Ayrica ses blogunu block ("play sound			
	[name_of_sound]") ve kontrol bloğunu ("when [the_key] key			
	pressed") nasil kullanacaklarını da öğreneceklerdir.			





Faaliyetin Süresi	45 dakika		
Öğrenme ve Öğretme	Aktif öğrenme		
Strateji ve Metotları	Oyun temelli öğrenme.		
Öğretme Formları	Ön öğrenme		
	Bireysel çalışma		
Öğretme Özeti	Motivasyon-Giriş (Motivasyon-Giriş, Uygulama, Düşünme ve		
	değerlendirme)		
	Oyunu oynayarak öğrencileri motive ederiz (kodu görmezler). Dersin		
	amacı oyunu böyle yapmaktır.		
	[Adum 1]		
	Ik adım oyunun arka nlanını belirlemektir. Arka nlan farklı bayyanlar		
	icermelidir Üc seceneğimiz yar		
	1 Öğrenciler arka nlanı kendileri cizebilirler		
	2. Öğrəncilər çovrimici üçrətsiz rəcim ərəvəbilirlər		
	2. Ogrenciler çevrimiçi ucretsiz resim arayabilirler.		
	3. Öğrenciler için arka plan sağlarız (zamandan tasarruf etmek		
	istiyorsak).		
	Öğrenciler nasıl arka plan ekleyeceklerini önceden öğrendikleri için bu		
	aktiviteyi bireysel yapabilirler.		





[Adım 2]

İkinci adım kadın çiftçiyi eklemektir. İlk Adım'daki gibi seçeneklerimiz aynı.

1. Öğrenciler kadın çiftçiyi kendileri çizebilirler.

2. Öğrenciler internette kadın çiftçinin ücretsiz imajını arayabilirler.

3. Öğrenciler için kadın çiftçi imajını sağlayabiliriz (eğer zamandan kazanmak istiyorsak).

Öğrenciler yeni bir karakteri nasıl ekleyeceklerini önceden öğrendikleri için bu aktiviteyi bireysel yapabilirler.



[Adım 3]

Daha sonra öğrenciler oyuncu için talimatları programlamalıdır. Talimatlar kadın çiftçi tarafından verilmektedir. Öğrenciler bunu *Looks* / say [string] ya da wait[n] (Görünümler Söyle [string] veya [n] kadar bekle) bloklarını kullanarak yapabilirler. Öğrenciler bunu nasıl yapacaklarını önceden öğrendikleri için bu aktiviteyi bireysel yapabilirler.



Uygulama

Şimdi öğrencilere oyuna nasıl ses ekleneceğini göstereceğiz. Üç seçeneğimiz var.

1. Snap'in medya kitaplığından bir sesi içe aktarma.





2. Bilgisayarımızdan bir sesi Snap! içine sürükleyerek içe aktarma.

3. Snap!'te yeni bir ses kaydetme.

Öğrencilere her üç seçeneği de frontal öğretim şeklinde gösteriyoruz. Hepsini tanıttığımızda aşağıdaki görevleri ayrı ayrı programlamaya başlarlar (öğretmen desteği ile).

[Adım 4]

Öğrenciler köpeğin sesini programlamalıdır. Oyuncu "D" tuşuna bastığında köpeğin havlaması gerekir. İlk olarak öğrenciler sesi Snap'in medya kitaplığından arka planın ses sekmesine aktarır.



Akabinde çocuklar köpeğin sesini seçerler (Dog 1 veya Dog 2).



Öğrenciler "D" tuşuna basıldığında çalınacak olan köpeğin sesini programlamak zorundadır. Bunu *Control/when [the_key] key pressed* veya *Sound/play sound [name_of_sound] (Kontrol/ / ...düğmesi basıldığında veya Ses/ sesi çal [sesin adı])*bloğunu kullanarak yaparlar.







[Adım 5]

Öğrenciler hayvan seslerini programlamalıdırlar. İlk olarak bilgisayarlarından ses eklemeleri gerekir. Bunu arka plan sesleri sekmesindeki sesleri sürükleyerek yaparlar.



Sesleri içe aktardıktan sonra yeniden adlandırmak için sesleri sağ tıklayabiliriz. Bizim durumumuzda onlara inek, domuz, at ve koyun şeklinde ad verebiliriz.

Daha sonra öğrencilerin sesi arka planın komut dosyalarına eklemesi gerekir. Bunu *Control/when[the_key] key pressed ve Sound/play sound[name_of_sound] (Kontrol/ / ...düğmesi basıldığında veya Ses/ sesi çal [sesin adı])* bloklarını kullanarak yapabilirler.







programlamaktır. Oyuncu oyuna başladığında kadın çiftçi "çiftliğime hoş geldiniz" demelidir. İlk olarak öğrenciler kadın çiftçinin karşılama selamlamasını kaydetmelidir. Bunu, (kadın çiftçinin) Sesler sekmesinde bulunan ses kaydedici (kırmızı düğme) ile yapabilirler. Sesi kaydettiklerinde (record), bilgisayara kaydetmeleri (save) gerekir.



Ses kaydı tamamlanınca sağ tıklayıp ismi değiştirebilirler. Bizim örneğimizde isim çiftlik olacak.



Şimdi öğrenciler sesi kadın çiftçinin sesinin dosyasına (script) kaydetmeliler. Bunu *sound/play sound[name_of_sound] (ses/ sesi çal [sesin adı]*) bloğu play sound fame ile yapabilirler.





Co-funded by the Erasmus+ Programme of the European Union







Öğretmenler İçin	 Snap! Sitesindeki örnek:
Araçlar ve Kaynaklar	 <u>https://snap.berkeley.edu/project?user=tadeja&project=Farm</u> İmaj için: <u>https://pixabay.com/</u> Ses için: <u>https://www.zapsplat.com/</u> Lajovic, S. (2011). Scratch. <i>Nauči se programirati in postani računalniški maček</i>. Ljubljana: Pasadena. Vorderman, C. (2017). <i>Računalniško programiranje za otroke</i>. Ljubljana: MK.
Öğrenciler İçin Kaynaklar ve Materyaller	 Snap!'ten örnek <u>https://snap.berkeley.edu/project?user=tadeja&project=Soun</u> <u>ds%20of%20the%20farm_0</u> imaj: <u>https://pixabay.com/</u> Ses: <u>https://www.zapsplat.com/</u> Öğrenci için talimatlar (C4G2_InstructionsForStudent.docx)




Öğrenme Senaryosu 6 - Bukalemunun yaz tatili, basit sürüm

Öğrenme Senaryosu	Bukalemunun yaz tatili, basit sürüm
Adı	
Geçmiş Programlama	Ön programlama becerileri gerekmemektedir.
Deneyimi	
Öğrenme Çıktıları	Genel Öğrenme Çıktıları:
	 Olaya dayalı nesne hareketi
	 Tekli veya çoklu renk algılama
	 Mantıksal ifadelerde Boolean değer okumaları
	 Farklı oyun durumlarını tanımlama, ayırt etme, dinamik olarak
	kontrol etme ve bunlara yanıt verme
	Algoritmik düşünceye odaklı özel öğrenme çıktıları:
	 Öğrenci, olayları kullanarak ok tuşlarıyla nesne hareketini
	gerçekleştirir ve kısıtlamaları dikkate alır.
	 Öğrenci, tekli veya çoklu renk algılama okuması için boole değerini
	elde etmek için bir algılama renk bloğu kullanır.
	 Öğrenci, nesne durumunun nesnenin dokunduğu renklerle ifade
	edilebileceğini fark eder.
	 Öğrenci iki temel, beş (tam) farklı durumu ayırt eder ve bunları
	mantıksal ifadelerle nasıl ifade edeceğini bilir.
	 Öğrenci nesnenin konumunun dinamik olarak değiştiğini fark eder.
	Mevcut durumu tekrar tekrar kontrol etmek için sonsuza kadar döngü
	(forever loop) kullanır.
	 Öğrenci, nesnenin mevcut konumuna bağlı olarak farklı yanıtlar
	vermek için if/koşul cümlesini kullanır.
Amaç, Görevler ve	Kısa açıklama: Nesnenin arka plan rengine göre kostümünü
Faaliyetlerin Kısa	değiştireceği basit bir oyun programlayın.





Tanıtımı	Görevler: Öğrenciler bukalemunu programlayarak görünüşünü
	(kostümü) değiştirmeli ve beş farklı durumda nerede olduğunu
	söylemelidir: 1) Denizde yüzerken rengini maviye çevirmeli ve
	"denizde yüzüyorum" 2) Deniz ve kumsal arasındayken cildi yarı mavi-
	yarı kum rengine döner ve "deniz ile sahil arasındayım" der. 3)
	Kumsalda kum rengi alır ve "kumsalda rahatlıyorum" der. 4) Kumsal
	ile orman arasında yarı yeşil-yarı kum rengine döner ve "kumsal ile
	orman arasındayım" der. 5) Ormanda cildi yeşile döner ve "ağacın
	gölgesinde serinliyorum" der.
	Öğrencilere, dinamik olarak değişen oyun durumları arasında ayrım
	yapmak ve doğru yanıtları vermek için renk bloğunu algılama ve
	mantıksal ifadelerde bloğun nasıl kullanılacağı anlatılacaktır.
Faaliyetin Süresi	45 dakika
Öğrenme ve Öğretme	Aktif öğrenme, işbirliğine dayalı öğrenme, problem çözme
Strateji ve Metotları	
Öğretme Formları	Ön öğrenme
	Bireysel Çalışma/İkili Çalışma/Grup Çalışması
Öğretme Özeti	(Motivasyon-Giriş, Uygulama, Düşünme ve değerlendirme)
	Bukalemun bir yaz tatiline çıktı. Denizde yıkanmayı, plajda dinlenmeyi,
	hava çok sıcak olduğunda serinlemek için yakındaki ağaçların
	barınağına gitmeyi seviyor. Bukalemun olduğu için rengini şimdiki arka
	plana göre değiştirir.
	[Temel Versiyon]
	Bu temel versiyonda iki farklı hâl arası geçiş yapacağız.
	[Adım 1]
	Öğrencilerden sahne arka planını, her biri farklı bir yeri temsil eden
	mavi ve kumlu aynı renkte iki parçaya bölünecek şekilde
	düzenlemelerini istiyoruz. Mavi renk deniz için, kum rengi (sandy) ise
	kumsal içindir. Öğrencilere arka planı daha gerçekçi hale getirmek





amacıyla diğer öğeleri dâhil etmeleri için talimat verebiliriz. Örneğin: dalgalar, deniz kabukları, kumdan kaleler, güneş şemsiyeleri, vb. Öğrencilerin daha büyük ve arka plandan tamamen farklı renklerle renklendirilmiş öğeleri seçmemeye dikkat etmeleri gerekir. Bu durumda renk algılama bloğu, karakterin sahnenin hangi bölümünde olduğunu algılayamayacaktır.



[Adım 2]

Öğrencilerin bukalemunu çizip, cildini iki farklı renkle boyamaları gerekmektedir.



[Adım 3]

Öncelikle bukalemunlarını tuşları kullanarak dört yönde hareket ettirmeleri gerekir. Kendi tuş kombinasyonlarını seçebilirler (Ör: Ok tuşları veya WASD-klavye harfleri). Bu noktada, önceki faaliyetlerden bunu nasıl yapacaklarını bildiklerini varsayıyoruz. Öğrencilere hareketi programlarken uygun blok kullanmazsak karakterin sahnenin dışına çıkabileceğini hatırlatmalıyız (sekme kenar bloğundaysa).

Bukalemun hareketini biraz daha gerçekçi kılmak için baktığımız yatay yöne bakacak şekilde sola veya sağa dönmesini istiyoruz (*point in direction / yönüne dön* bloğu kullanarak).







[Adım 4]

Öğrencilere, dokunduğu rengi (renkleri) algılayan karakter kavramını tanıtıyoruz. "touching color?" bloğu ile Boolean değerleri (- belirli bir renge dokunuyorsa Doğru veya Yanlış) biçiminde bilgi alabiliriz. Bu bloktan Boolean değeri aldığımız için, onu gövdesinde listelenen komutları çalıştırıp çalıştırmayacağımıza karar verildiği If/Koşul cümlesinin başında kullanabiliriz.

Daha sonra öğrencilerle bukalemunun sahnedeki farklı pozisyonlarının neler olabileceğini ve bunları "touching color? (renge dokunur)" bloğu kullanarak nasıl ifade edebileceğimizi öğretebiliriz.

İşte iki örnek;

1. Mavi renge dokunuyor -> Touching color [blue]?

2. Kumlu renge dokunuyor -> Touching color [sandy]?

Belli bir renge dokunduğunda görünüşünü değiştirmeli ve ona nerede olduğunu söylemesini sağlamalıyız. Kostümleri arasında geçiş yaparak bir karakterin görünümünü değiştirebiliriz. Bu, olası kostümlerden hangisini görüntülemek istediğimizi seçtiğimiz *Looks/switch to costume[option] (*Görünümler / kostüme geçiş [seçenek]) bloğuyla yapılır. Bukalemun konuşmasını sağlamak için *Looks/say[text] (*Görünüm / söyle [metin]) bloğunu kullanıyoruz.

Çünkü "if - else" koşullu bloğunu kullanabileceğimiz sadece iki olasılık vardır.

Hangi rengi kontrol edeceğimizi seçebiliriz. Ardından diğer renkler "else" durumuna girecektir. Örnek kodda kumlu (sandy) rengi seçtik.





say l'am'sunbathing'on'a beach switch to costume kameleon_oranzen lise say l'am'swimming

switch to costume kameleon_moder

[Adım

5]

Kullandığımız programın tüm süresi boyunca belirli komutları yürütmemiz gereken durumlar için – forever loop (- sonsuza kadar döngü) butonunu kullanırız. Sonsuz döngünün gövdesi altında yazılan her şey defalarca yürütülecektir. Öğrencilerimizle bu oyunu oluşturmak için bizim durumumuzda tam olarak istediğimizin/ihtiyacımızın bu komut olduğunu tartışıyoruz.

[Nihai Kod]



[Tam Versiyon]

[Adım 1]

Öğrencilerden sahne arka planını, her biri farklı bir yeri temsil eden aynı renkte üç parçaya bölünecek şekilde düzenlemelerini istiyoruz. Deniz için mavi renk, kumsal için kum rengi ve orman için yeşil. Arka planı daha gerçekçi hale getirmek için dalgalar, deniz kabukları, kumdan kaleler, güneş şemsiyeleri, ağaçlar vb. gibi başka öğeler ekleyebilirler. Ancak eklenen öğelerin ana karakterin kendisinden daha büyük olmamasına dikkat etmeleri gerekir. Çünkü bu durum





karakteri üç renkten hiçbirine dokunmayacak ve Snap'in algılama özelliği sahnenin hangi bölümünde olduğunu karakterin algılayamayacaktır. [Adım 2] Öğrencilerin bir bukalemun çizmeleri ve derisini sahnedeki konumunu temsil eden beş farklı kombinasyonda boyamaları gerekiyor. [Adım 3] Öncelikle bukalemunlarını tuşları kullanarak dört yönde hareket ettirmeleri gerekir. Kendi tuş kombinasyonlarını seçebilirler (Örnek: Ok tuşları veya WASD). Bu noktada, önceki faaliyetlerden bunu nasıl Öğrencilere, yapacaklarını bildiklerini varsayıyoruz. hareketi programlarken uygun blok kullanmazsak karakterin sahnenin dışına çıkabileceğini hatırlatmalıyız (sekme kenar bloğundaysa). Bukalemun hareketini biraz daha gerçekçi kılmak için, baktığımız yatay yöne bakacak şekilde sola veya sağa dönmesini istiyoruz (point in direction bloğu kullanarak).





[Adım 4]

Öğrencilere, dokunduğu rengi (renkleri) algılayan karakter kavramını tanıtıyoruz. "touching color?" bloğu ile Boolean değerleri (- belirli bir renge dokunuyorsa Doğru veya Yanlış) biçiminde bilgi alabiliriz. Bu bloktan Boolean değeri aldığımız için onu gövdesinde listelenen komutları çalıştırıp çalıştırmayacağımıza karar verildiği If/Koşul cümlesinin başında kullanabiliriz.

Daha sonra öğrencilerle bukalemunun sahnedeki farklı pozisyonlarının neler olabileceğini ve bunları "touching color?" bloğu kullanarak nasıl ifade edebileceğimizi öğretebiliriz.

Hızlıca beş tane olduğunu anlıyoruz.

1. Tamamen mavi kısımda -> Touching color [blue]? ([mavi] renge dokunuyor mu?)

Mavi ve kumlu kısım arasında -> Touching color[blue]? VE Touching color [sand]? ([mavi] renge dokunuyor mu? VE Renkli [kum] dokunmak?)

3. Tamamen kumlu kısımda -> touching color [sand]? (renkli [kuma] dokunuyor mu?)

 Kumlu ve yeşil kısım arasında -> Touching color[sand]? VE Touching color [green]? (Rengine [kum] dokunuyor mu? VE Renkli [yeşil] dokunmak?)

5. Tamamen yeşil kısımda -> Touching color [green]? ([yeşil] renge dokunuyor mu?)

Belli bir renge/renklere dokunduğunda bukalemun görünümünü değiştirmeli ve ayrıca ona nerede olduğunu söyletmeliyiz. Kostümleri arasında geçiş yaparak bir karakterin görünümünü değiştirebiliriz. Bu, olası kostümlerden hangisini görüntülemek istediğimizi seçtiğimiz *Looks/switch to costume[option]* (Görünümler/kostüme geçiş





[seçenek]) bloğuyla yapılır. Bir bukalemun konuşması için Looks/say [text] bloğunu kullanıyoruz.

Öncelikle, bukalemunun sahnenin tamamen aynı renk kısmında olduğu daha basit durumlarla ilgileniyoruz:



Daha sonra VE mantıksal işlemci kullanarak mantıksal bir ifade oluşturuyoruz. Çünkü bukalemunun aynı anda iki renge dokunup dokunmadığını doğrulamak istiyoruz.



Yukarıdaki koşullu cümleleri birleştirip *When Green Flag Clicked* (Yeşil Bayrak Tıklandığında) bloğunun altına koyarsak bu koşulların tam olarak bir kez kontrol edileceğini görürüz. Ana karakterin hareketini kontrol ettiğimiz için bukalemun pozisyonunun oyun sırasında sürekli değişeceğini fark etmelerine yardımcı oluyoruz. Bu nedenle, bu koşulları sürekli olarak yalnızca bir kez değil, kelimenin tam anlamıyla her zaman kontrol etmemiz gerekiyor!

[Adım 5]

Programın tümünün çalışması için belirli komutları yürütmemiz gereken durumlar için – forever loop (- sonsuza kadar döngü)'u kullanırız. Sonsuz döngünün gövdesi altında yazılan her şey defalarca yürütülecektir. Öğrencilerimizle bu oyunu oluşturmak için bizim istediğimiz / ihtiyacımızın bu komut olduğunu tartışabiliriz.





[Nihai kod]		
when clicked forever if touching ? say learnsunbashingron a beach switch to costume kameleon_oranzen *	when left arrow key, pressed point in direction 20 change x by 510 if on edge, bounce	when aget arrow key pressed point in direction 200 change x by 10 if on edge, bounce
if touching ? say learnswimming switch to costume kameleon_moder if touching ?	when down aros - key pressed change y by 50 if on edge, bounce	when up arrow wey pressed change y by 10 if on edge, bounce
say Coolingrofilundenatives switch to costume kameleon_zelen = if touching ? and touching ? say (Ismibetweentheses and the back)		
switch to costume kameleon_oranzen_model -		

[Öğrenciler Kodu Ayarlarlar]

Bu aktiviteyi basitleştirmek için kodun bir kısmını önceden bir şablon dosyasında hazırlayabilir ve öğrencilere bunu tamamlamaları için talimat verebiliriz.

Önerilen öğrenme yolunu izleyen öğrenciler nesneyi anahtarlarla hareket ettirmeyi zaten öğrendi. Böylece hareket kodunu bir şablon dosyasına ekleyebiliriz. Tuş ayarlarını ok tuşlarından özel düzenlemeye (örneğin WASD) değiştirebilirler.



Sonsuz döngü kavramını (forever loop) ve arka plan rengini algılamak için sonsuz döngüyü nasıl kullanılacağını anlamalarına yardımcı olmak ve iki durumu algılamak için kod ekleyebiliriz:





	1) Nesne tamamen tek renktedir.
	2) Nesne aynı anda iki renge dokunmaktadır.
	Her vaka için kodu tamamlamaları talimatını verdik.
	Önerilen kod şablonu:
	when clicked forever if touching ? say lam subbathing on a beach switch to costume kameleon_oranzen if touching ? and touching ? say lam between the sear and the beach switch to costume kameleon_oranzen_moder
Öğretmenler İçin	• Snap!:
Araçlar ve Kaynaklar	Temel:
	https://snap.berkeley.edu/project?user=zapusek&project=cha
	meleon simple
	Tam:
	https://snap.berkeley.edu/project?user=zapusek&project=cha
	<u>meleon</u>
	 Lajovic, S. (2011). Scratch. Nauči se programirati in postani računalniški maček. Ljubljana: Pasadena.
	 Vorderman, C. (2017). Računalniško programiranje za otroke. Ljubljana: MK.
Öğrenciler İçin	 Snap! Şablonu:
Kaynaklar ve	https://snap.berkeley.edu/project?user=zapusek&project=cha
Materyaller	<u>meleon template</u> Spapl/toki.vari.tamamlanmis (Half Bakod) aktivito
	• Shap! teki yan tamamanniş (nan-bakeu) aktivite
	meleon half baked
	Öğrenci için talimatlar (C4G2_InstructionsForStudent.do





Öğrenme Senaryosu 7 - Prense ve prensese evcil hayvanlarını bulmada yardım etme

Öğrenme	Prense ve prensese evcil hayvanlarını bulmada yardım etme
Senaryosu Adı	
Geçmiş	Karaktere metin ekleme
Programlama	Events (Olaylar) kullanarak ok tuşlarıyla nesne hareketi
Deneyimi	Nesne için <i>object is touching (nesne değiyor)</i> koşulunu kullanmak
	Event (Olaylar) kullanma
Öğrenme Çıktıları	Genel Öğrenme Çıktıları:
	Object is touching için koşul ifadeleri
	 Koordinatlara geçme
	 Kalem yukarı, kalem aşağı (pen up/pen down)
	Kalem rengi
	Algoritmik düşünceye odaklı özel öğrenme çıktıları:
	 Öğrenci, belirli bir renge dokunursa, nesne durumu için if cümlesini kullanır ve nesneyi geri koyar. Karakter için x ve y koordinatlarını başlatır. Öğrenci, çizgi/yol çizmek için kalem yukarı ve aşağı kalem özelliklerini kullanır. Öğrenci bağlandığı çifte göre kalem rengini değiştirir. Öğrenci, başlangıçta önceki tüm yolları temizlemesi gerektiğini fark eder.
Amaç, Görevler ve	Kısa Açıklama: Kızlar, Prenses'e kedisini; Prens'e köpeğini bulması için
Faaliyetlerin Kısa	yardım etmelidir. Bunu prensese gidip ona bir çizgi çizerek kedisinin yolunu
Tanıtımı	göstererek yapacağız. Benzer şekilde kızlar Prens'e köpeğine giden yolu
	gösterecekler. Ancak kızlar, yollarının kesişmemesi için hayvanlar
	arasındaki çakışmalardan kaçınmalıdır.
	Görevler: İlk Adım'da öğrenciler uygun arka planı (labirent) seçmelidir.
	Labirente beş karakter eklerler. Karakterleri bir kız, bir prenses, bir prens,
	bir kedi ve bir köpek. Daha sonra kız için düğmelerle hareket etmeyi
	(events kullanarak) programlayacaklar. Ancak burada karakterin çimlerin





üzerinde yürümemesine özen gösterilmelidir. Akabinde kalemle çizim ve
events ile kalem rengini değiştirmeyi programlarlar. Ayrıca, yolu
temizleyen ve kıza talimatları veren başlangıç olayını programlamaları
gerekir.
Amaç: Öğrencilere anahtar hareketlerle çizim yaptırılacaktır. Bunun
yanında, karakterin tüm ekranda hareket etmesini önlemek için koşul
ifadelerini nasıl kullanacaklarını öğrenecekler.
30 dakika
Aktif öğrenme, oyun tasarım tabanlı öğrenme, problem çözme
Ön öğrenme
Bireysel çalışma
(Motivasyon-Giriş, Uygulama, Düşünme ve değerlendirme)
Aşağıdakiler öğrencilere başta verilir:
 Arka plan
Kız karakter
 Kız karakter Tek yönde hareket kodu
 Kız karakter Tek yönde hareket kodu Kız, Prensese kedisini bulması ve Prensin köpeğini bulması için
 Kız karakter Tek yönde hareket kodu Kız, Prensese kedisini bulması ve Prensin köpeğini bulması için hayvanlarına giden yolu göstererek (çizerek) yardım etmeye karar verir.







[Adım 1]

Öğrencilerden sahne arka planını bir labirent olarak düzenlemelerini istiyoruz. "if touching color" (renge dokunuluyorsa) uygulaması için ya arka planın (çimen) tek renkli olması ya da bizim durumumuzda olduğu gibi yolun tek renkli bir çerçeveye sahip olması gerekir. Uygun arka planı bulmada bu "sorunları" önlemek için onlara bu arka planı veriyoruz.

[Adım 2]

Öğrenciler aktivitenin başında zaten bir kız karakter oluşturmuşlardı. Başka dört tane karakter daha bulup labirente koymaları gerekiyor. Tüm hareketli resimler için boyutu uygun (labirentteki yolların genişliğinden daha küçük olan) şekilde ayarlamaları gerekir. Her hareketli grafik için aşağıdaki kodu kullanabilirler.



Kız karakter için önerilen büyüklük 8%'dir. Diğer karakterler daha büyük olabilir.

[Adım 3] Bundan sonra, kızın hareketini tuşları kullanarak dört yönde yapmaları





gerekmektedir. Önceki faaliyetlerden bunu nasıl yapacaklarını zaten bildiklerini varsayıyoruz. Onlara bir yön için kod veriyoruz. Bu da onların başka bir üç yön yapmasına yardımcı oluyor.



[Adım 4]

Bir sonraki adımda, kızın çayır boyunca hareket etmesini önlemek zorundalar. Bunu, kahverengi renge dokunuyorlarsa koşullu bir blok ekleyerek yapabilirler. Kız kahverengine (yolun sonuna) dokunuyorsa 10 adım geri gider. O iki adımı görmüyoruz ve sanki kız aynı pozisyonda kalıyor. Bu, sağa hareket etmek için bir koddur. Yani x'i -10 ile 10 adım geri, değiştirmek anlamına gelir.



Bu kodu önceki kodun altına eklerler. Ör: Sağ ok için.



Diğer üç yön için de benzer şeylerin yapılması gerekiyor.

[Adım 5]

Daha sonra çizimi programlarlar. Bunu, when key pressed (tuşa











<pre>when in clicked set size to in in per up clear gd to x: 190 y: 133 say heip-the-Prince-so and the-Prince-to-Ind-their animals, for if secs say Snow-them-theright-way-by-drawing-the path, for if secs when up acrow = key pressed per up when d = key pressed per down when per key pressed set pen color to when per key pressed set pen color to </pre>
Örneğin Prenses
Öğrenciler isteklerine göre ek görevler ekleyebilir veya aşağıdaki görevleri takip edebilirler.
 Prens ve prenses için başlangıç koordinatlarını ayarlayın ve hareketleri için bir kod yazın. Onlar için uygun boyutu ayarlayın. Hayvanlarına bir yol çizmeliler.
 Kız için başka bir karakter (hayvan) ekleyin.
 Her hareketli grafik farklı bir renkle çizilmelidir.
• Başıangıç talımatlarını ayarlayın.
 Bır hareketli grafiği taşımak ve bir hareketli grafiği tıklatarak çizim yapmak için talimatlar ekleyin. Örneğin, prenses divor ki: "W. S. A ve D
tuşlarına basarak beni hareket ettiriyorsun. Tuşa basarak yolu çiziyorum 3.





	Tuşa basarak çizmeyi bırakıyorum 4. Kedimi bulmama yardım et!"
Öğretmenler İçin	Snap! te tüm aktivite mevcut:
Araçlar ve	https://snap.berkeley.edu/project?user=mateja&project=Helping% 20Prince%20and%20Princess%20to%20find%20their%20animals
каупактаг	 Shap! të çozumleri llë ek gorevier <u>https://snap.berkeley.edu/project?user=mateja&project=Helping%20Princ</u> <u>e%20and%20Princess%20to%20find%20their%20animals%20%2B</u> <u>%20Add.%20Task</u> Lajovic, S. (2011). Scratch. <i>Nauči se programirati in postani</i> računalniški maček. Ljubljana: Pasadena. Vorderman, C. (2017). <i>Računalniško programiranje za otroke</i>. Ljubljana: MK.
Öğrenciler İçin	 Snap!'te yarı tamamlanmış (Half-Baked) aktivite.
Kaynaklar ve	https://snap.berkeley.edu/project?user=mateja&project=Helping%20Princ
Materyaller	 <u>e%20and%20Princess%20to%20find%20their%20animals%20-</u> <u>%20Part</u> Öğrenci için talimatlar (C4G2_InstructionsForStudent.docx)





Öğrenme Senaryosu 8 - Tebeşir ile çizim yapmak

Öğrenme Senaryosu	Tebeşir ile çizim yapmak
Adı	
Geçmiş	Karaktere metin ekleme
Programlama	Kalem ile çizim (kalem yukarı, kalem aşağı, renk belirleme)
Deneyimi	Adımlarla Hareket Etmek
	Döngüler(Loops) kullanma
	Olayları (Events) kullanma
Öğrenme Çıktıları	Genel Öğrenme Çıktıları:
	 Tekrarlama döngüsü
	 90 derece döndürme Bir vöne dönme
	 Arka planı değiştirme
	Algoritmik düsünceve odaklı özel öğrenme cıktıları:
	• Öğrenci, aynı bloklar 2/4 kez tekrarladığında döngü tekrarını (loop repeat) kullanır.
	• Öğrenci, farklı şekiller (kare, dikdörtgen, "T" harfi) çizerken 90 derece döndürmeyi kullanır.
	 Öğrenci, 90 yönündeki anlam noktasını anlar.
	 Öğrenci, bir tuşa basıldığında bir olay kombinasyonuyla arka planı nasıl değiştireceğini bilir.
Amaç, Görevler ve	Kısa açıklama: Oyuncu üç farklı arka plan elde etmek ve noktaları üç
Faaliyetlerin Kısa	farklı şekle bağlamak zorundadır. Şekilleri oluşturmak için - bir kare, bir
Tanıtımı	dikdörtgen ve bir "T" harfidir.
	Görevler: Öğrenciler "tahtaS" (boardS) arka planını seçerler ve bir kare
	çizmeye başlarlar. Başlangıç konumları "A" noktasıdır. Kare çizerken
	belirli adımları 4 kez tekrar ederler. Yani aynı kodu 4 kez yazmak yerine 4
	kez döngü tekrarı kullanabilirler. Sonra yine bir döngü tekrarı kullanarak
	bir dikdörtgen çizerler. Bu sefer 2 kez tekrarlanır. Son görevlerinde
	adımların sayısını bulmaları gereken noktaları "T" harfi şeklinde
	birleştirmeleri gerekiyor. Mümkün olduğunda döngü tekrarını
	kullanabilirler.





	Amaç: Bu etkinlikte öğrenciler bir kodla farklı şekillerin nasıl çizileceğini
	öğreneceklerdir. Ayrıca kodu kısaltmak ve arka planı değiştirmek için
	döngü tekrarını kullanmayı öğreneceklerdir.
Faaliyetin Süresi	60 dakika
Öğrenme ve	Aktif öğrenme, oyun temelli öğrenme, problem çözme
Öğretme Strateji ve	
Metotları	
Öğretme Formları	Ön öğrenme
	Bireysel Çalışma/ Grup Çalışması
Öğretme Özeti	(Motivasyon-Giriş, Uygulama, Düşünme ve Değerlendirme)
	Aşağıdakiler öğrencilere başlangıçta verilir.
	 Birleştirmeleri gereken tüm noktaların bulunduğu üç arka plan
	• Tebeşir karakteri (Sprite)
	Tebeşir bir kare, bir dikdörtgen çizmek ve "T" harfi şeklindeki noktaları
	birleştirmek istemektedir. Ancak nasıl hareket edeceğini ve nasıl
	döneceğini bilemez. Bir kod yazın ve tebeşire nasıl yapılacağını gösterin!
	[Adım 1]
	DRAW A SQUARE
	D , C
	PA D
	Öğrenciler yukarıdaki arka planla başlar. Kare çizmek için bir kod
	yazarlar. "A" noktasından başlayarak X Adım kadar "B" noktasına hareket
	ettirirler. Sola 90 derece dönerler. X Adım kadar "C" noktasına hareket





ettirirler. 90 derece sola dönerler. X Adım kadar noktaya "D", sola 90 derece dönün. "A" noktasına X adım getirin (ve sola 90 derece dönün).



90 derece döndürme'yi (*turn 90 degrees*) kullanmak en kolay yoldur. Çünkü her zaman 90 derece döndürmeyi kullanabiliriz (sadece sola veya sağa dönüp dönmemize bağlıdır). 0, 90, 180, -90 yönünde nokta kullanmak ise bir diğer seçenektir. Ancak bu biraz daha karmaşıktır. Çünkü 4 olasılığı ayırmak zorundayız ve bir döngü tekrarı kullanamıyoruz. Sadece çizimi/tüm adımları görmek için *1 sn. Bekle* (*Wait 1 secs*) bloğu eklenmiştir. Bu blok olmadan tüm kod bir saniyede gerçekleşir. Öğrencilerin bunu anlayabilmeleri için bu blok olmadan denemelidir.

Öğrenciye, mümkünse kodu nasıl kısaltacaklarını soruyoruz. Tekrarlardan başka bir kısım var mı? Cevap Evet. Aynı kodu 4 kez yazmak yerine, programlamada döngü tekrarı kullanıyoruz.



Gerçekten ne çizdiğimizi görmek istiyorsak *tekrar (repeat)* döngüsünden önce bir *kalem aşağı (pen down*) bloğu koymalıyız.





Tebeşir dönüş gerçekleştirirken etrafında dönmemesini istiyorsak yön bloğunda *döndürme* (*don't rotate*) seçeneğine tıklarız.



[Adım 2]

Kodu etkinleştirmek için öğrenciler *olay (event)* bloğunu kullanır. Örneğin *S tuşuna basıldığında (when S key is pressed)*. Ayrıca kalem rengini de ayarlayabilirler ve önceki aktivitelerden bildikleri gibi aşağıdaki blokları takip edebilirler. *Kalem yukarı (pen up)* (önceki oyundan kalması durumunda), *temizle (clear)* (önceki oyundan çizimi temizler) ve *x, y'ye git (go to x, y)* (tebeşir her zaman bu koordinatlarda başlar).

Bazen oyun sırasında programı durdurduğumuz ve ardından bir karakteri "garip bir yöne" döndürüldüğümüz olur. Bu bir oyuna yeniden başlarken sorun olabilir. Eğer bir karakter yanlış döndürülürse örneğin ilk adımda sağa değil aşağıya gidecektir. Bu sorunu önlemek için *90 yönünde dön (point in direction 90)* bloğu ekliyoruz.

> when s key pressed set pen color to clear pen up point in direction 90 go to x: 60 y: 105 pen down repeat 4 move 150 steps wait 1 secs wait 1 secs

[Adım 3]

Bir kare çizdikten sonra bir dikdörtgen çizmek istiyoruz. Bu, arka planı değiştirmemiz gerektiği anlamına gelir. Bunu iki Adımda yapacağız: a) Arka plana tıklıyoruz (ekranın sağ tarafında adlı pano).







Arka planlara tıkladığınızda, bu etkinlik için önceden hazırlanmış olan üç gerekli arka planı (tahta Kare, tahta Dikdörtgen, tahtaT (*boardSquare, boardRectangle, boardT*)) görebiliriz.



Bir kod yazmak için öğrencilerin Komut Dosyalarına tıklamaları gerekir. Değişen arka planı programlamak için R tuşuna basıldığında bir event (olay) bloğu seçerler ve ardından *BoardRectangle (dikdörtgen pano) kostümüne geçin* tuşuna basarlar (*switch to costume boardRectangle*).











"R" tuşuna bastıktan sonra arka plan buna dönüşür. Öncekine benzer şekilde noktaları birleştirmeleri ve bir dikdörtgen çizmeleri gerekir. Öğrenciler önceki kod bloklarını kopyalayabilir ve programın bir dikdörtgen çizmesi için bunları düzeltebilirler.

Döngü tekrarını (*repeat*) değiştirirler. Şimdi bu döngü 2 kez tekrarlanacak.



[Adım 5]

Bir dikdörtgen çizdikten sonra öğrenciler noktaları "T" harfi şeklinde birleştirecekler. Bu, arka planı değiştirmeleri gerektiği anlamına geliyor. bu yüzden bu adımda aslında [Adım 3]'ü tekrar ediyorlar. Sadece ("T") harfi ve kostümü (boardT) değiştiriyorlar.

a) Arka planı değiştirmek için bir kod yazdıkları arka plana (ekranın sağ tarafındaki *board* adlı panoya) tıklarlar. Bunu *T tuşuna basıldığında* (*when T key pressed*) ve ardından *kostüm T panosuna geç* (*switch to costume boardT*) kullanarak yapacaklar.



 b) Tebeşire ve [Adım 4] ün altındaki koda blok eklemek için tekrar tıklarlar. Burada oyuncuya arka planı değiştirmek için ne yapması gerektiğini söylerler. Yani "T" tuşuna basın.







[Adım 6]



"T" tuşuna bastıktan sonra arka plan buna dönüşür. Öncekine benzer şekilde noktaları birleştirmeleri ve bir "T" harfi çizmeleri gerekir. Öğrenciler önceki kod bloklarını kopyalayabilir ve düzeltebilirler.

Öğrenciler, öncekiyle aynı olmayan başlangıç koordinatlarını değiştirmek zorunda kalacaklar. Önceki faaliyetten doğru koordinatları nasıl belirleyeceklerini zaten biliyorlar.

Daha sonra "T" harfini çizmek için bir kod yazarlar. Adımların sayısını bulmaları gerekir. Olası bir çözüm şudur:

















	 Yeni bir arka plan ekleyin ve bazı noktalar çizin. Noktaları birleştiren bir kod yazın. Bir arka plan çizebilir veya size verilenlerden birini kullanabilirsiniz.
Öğretmenler İçin Araçlar ve Kaynaklar	 Snap!'te yer alan tüm aktivite <u>https://snap.berkeley.edu/project?user=mateja&project=Drawin</u> <u>g%20with%20a%20chalk</u> Lajovic, S. (2011). Scratch. Nauči se programirati in postani računalniški maček. Ljubljana: Pasadena. Vorderman, C. (2017). Računalniško programiranje za otroke. Ljubljana: MK.
Öğrenciler İçin Kaynaklar ve Materyaller	 Snap!'teki Yarı Tamamlanmış Aktivite <u>https://snap.berkeley.edu/project?user=mateja&project=Drawin</u> <u>g%20with%20a%20chalk%20-%20Part</u> Öğrenci için talimatlar (C4G2_InstructionsForStudent.do





Öğrenme Senaryosu 9 - Çöpleri toplamak ve parkı temizlemek

Öğrenme	Çöpleri toplamak ve parkı temizlemek
Senaryosu Adı	
Geçmiş	Başlangıç koordinatlarını ayarlama
Programlama	Karakterin boyutu ayarlama
Deneyimi	Metin ekleme
	Ok tuşları kullanarak nesne hareket ettirme
	Nesnenin durumu için nesne dokunuyor (object is touching) şart ifadesini
	kullanma
Öğrenme Çıktıları	Genel Öğrenme Çıktıları:
	Değişkenler
	Karakter göstermek ve gizlemek
	Karakter kopyalamak
	Kod blokları kopyalamak
	• Şart ifadeleri
	Algoritmik düşünceye odaklı özel öğrenme çıktıları.
	 Öğrenci toplanmış atık sayısını hesaplamak için değişken kullanır.
	 Öğrenci oyunun başlangıçta karakteri gösterip bir olaydan sonra onu gizler.
	 Öğrenci karakteri kopyalar (1 şişeden mesela 4 şişemiz olacak).
	 Öğrenci kod blokları kopyalayabilir (şişe karakterinden kâğıt karakterine).
	 Bulunan atığın sayısını ve karakterin durumunu kontrol etmek için öğrenci şart ifadeleri kullanabilir.
Amaç, Görevler ve	Kısa açıklama: Bahçe çöple doludur. Kız (karakter) onu temizlemeye karar
Faaliyetlerin Kısa	verir. Kız dağınık çöpleri toplayıp çöp kutusuna atacak.
Tanıtımı	Görevler: Oğrenciler kızın başlangıç koordinatlarını ayarlayacaklar. Oyun tüm çöp toplanınca sona erecektir. Bunu sağlamak için öğrenciler değişkenin yardımıyla çöp sayısını hesaplayacaklar (1 toplanmış atık = 1 puan). Kız çöpe dokununca onu toplayacak ve topladığı çöp yok olarak kıza bir puan kazandıracak.
	Amaç: Öğrencilerin değişkenleri kullanmaları, kod veya blok hatta tam bir
	karakter kopyalayabilmelerini sağlamak.
Faaliyetin Süresi	45 dk.





Öğrenme ve	Aktif Öğrenme, Oyun Temelli Öğrenme, Problem Çözme
Öğretme Strateji ve	
Metotları	
Öğretme Formları	Ön Öğrenme, Bireysel Çalışma
Öğretme Özeti	(Motivasyon-Giriş, uygulama, düşünme ve değerlendirme)
	Öğrencilere başlangıçta aşağıdakiler verilir.
	• Arka plan
	 Kız karakteri (hareket koduyla), şişe karakteri, kâğıt karakteri ve çöp kutusu karakteri
	Kız bahçede oynamak istemektedir. Oraya giderken bahçenin çöp ile dolu olduğunu fark eder. Bahçeyi temizleyecek, çöpleri toplayacak ve sonra da bahçede artık dinlenebilecektir.
	[Adım 1]
	Arka plan verilir. Ayrıca kız karakterini tuşlarla hareket ettirebilmek için
	kod verilir. Bu kod kahverengi çizgiye dokunma koşulunu da içerir.











Motion	Control	Variable name
Looks	Sensing	
Sound	Operators	points
Pen	Variables	• for all sprites for this sprite only
Make a variable	<u>م</u>	
Delete a variab	le +	OK Cancel

Turuncu renkli bir Değişkenler (*Variables*) bloğunu seçiyoruz. Ardından Değişken oluştur (*Make a variable*) düğmesine tıklıyoruz. Bir Değişken adı (*Variable name*) yazıyor ve Tamam'a tıklıyoruz. Ardından bir blok noktası belirir.



Eğer kutucuk tıklanarak seçildi ise değişken ve üzerindeki değer ekranda görünür.



Oyunun başında alınan çöp olmadığından değişkenin değeri 0 olmalıdır. [Adım 1] kodunun altına __yı 0 ayarla(set __ to 0)' a blok seti ekler. Açılır menüye tıklayarak uygun değişkeni, yani puanları seçerler.

set points to D

[Adım 3]

Öğrenciler bir şişe için bir kod yazarlar. Buradaki fikir, karakterin kıza dokunduğunda ortadan kaybolmasıdır (bu, gizlemek anlamına gelir). Yani kod, karakter kıza dokunduğunda başlayacak. Öyleyse hangi durumda çöpü alacağını düşünmeliyiz. Çöp kutusu alındığında gizlendiğini söylersek, onu ancak = gösteriliyorsa yani hala oradaysa alabiliriz. Karakter (şişe) hala oradaysa, onu alıp "değişken kutusuna koyarız". Değişken noktalarında önceden 0 elemanımız vardı. Şimdi 1'e sahibiz. Çöpü aldığımızda değişken sayısını (puanları) 1 değiştirdiğimizi, yani 1 arttığını görebiliriz. Çöp toplandığında çöpü gizleriz.







Şimdi kodumuzun doğru olup olmadığını test edebiliriz.

Yeşil bayrağa tıklayıp şişeyi alıyoruz. Şişe kaybolmalı ve puan sayısı 1 olmalıdır. Sonra oyunu tekrar oynamak istiyoruz ve yeşil bayrağa tekrar tıklıyoruz. Ne oluyor? Şişe şimdi nerede?

Şişe gizlendi. Daha önce saklamıştık. Yani oyunun başında şişenin gösterilmesini programlamalıyız. Bunu *göster (show)* bloğunu seçerek yapıyoruz.



[Adım 4]

Artık öğrenciler oyunlarında daha fazla şişe olmasını istiyor. Böylece karakterlerini kolayca kopyalayabilirler. Karaktere sağ tıklayıp *kopyala (duplicate*) seçerler.



Şimdi fare ile yeni şişeye tıklarlar ve onu labirentin içinde bir yere sürüklerler.

Bu adımı tekrarlayıp şişeyi tekrar kopyalayabilirler.

[Adım 5]

Şimdi öğrenciler kâğıt karakter için aynı koda sahip olmalılar. Kod bloğuna sağ tıklayarak şişenin kodunu kopyalayabilirler.







Kağıt karakter üzerine fareyle tıklayarak onu kâğıt karakterin içine bırakın.



Yeşil bayrak tıklandığında-göster (when green flag clicked – show) kod bloğunu kopyalamak için bu adımı tekrarlarlar.

Ayrıca labirentte daha fazla kâğıt çöpü olması için [Adım 4] 'ü tekrar edebilir ve tüm kâğıt karakterleri kopyalayabilirler.

[Adım 6]

Öğrencilerin yapması gereken son şey, çöp tenekesi için bir kod yazmaktır. Çöp tenekesi karakteri zaten verilmiştir. Labirentin içinde her yere taşıyabilirler.

Ayrıca bu kod kız ona dokunduğunda aktif hale gelecektir.

Çöp kutusu, tüm çöplerin alınıp alınmadığını kontrol etmek zorunda kalacak. Değişken noktalar sayesinde bunu yapmak kolay olacaktır. Diyelim ki oyunda 8 çöp var. Bu yüzden öğrencilerin puan sayısının 8'e eşit olup olmadığını kontrol etmeleri gerekiyor. Eğer öyleyse, bu tüm çöplerin toplandığı, aksi takdirde olmadığı anlamına gelir. Bunu programlamak için if/eğer ifadesini kullanacak ve oyuncuya tüm çöpleri alıp almadığını söylemek için bazı metinler ekleyecekler.





2 secs	<pre>sets in the set of the set o</pre>	
ht arrow key pressed direction 90 steps ling 2 10 steps	[Nihai Kod] Kz	when touching Girl ? If points = 0 say Congratulations!-You picked up all the trash! for (2) secs efse say Come-back when you pick up all the trash. for (2) secs
ht arrow key pressed direction 90 steps hing ? 10 steps	<pre>Kiz</pre>	[Nihai Kod]
ht arrow = key pressed direction 90) steps ling ? =10 steps	<pre>when elicked go to x: '189' y: '155 say Pickup all the trash and put it in the trash can. for '2' secs set points = to "</pre>	Кız
		<pre>when included go to x: 189 y: 155 say Pickupalithe trash and put lein the trash can. for 2 secs set points = to</pre>
		when touching GM ? If shown? change points by 1 hide
	when clicked show when touching Girl ? If shown? change points = by 1 hide	Çöp Kutusu





	 (volunti (ouching (gir ?)) (a) Congratulations! You picked up all the trash for ?? secs (a) Congratulations! You picked up all the trash for ?? secs (Ek Görevler) Öğrenciler isteklerine göre ek görevler ekleyebilir veya aşağıdaki görevleri takip edebilirler. Başka bir atık türü ekleyin (örn. Biyolojik Atık). Çöp kutusu, örn. "X şişe, Y kâğıt ve Z karpuz aldınız". Bir oyuncu tüm çöpleri alırsa çöp kutusu şunu söyler: "Tebrikler! Tüm çöpleri aldını" Bir oyuncu tüm çöpleri toplamazsa, çöp kutusu ona hangi çöpün alınmadığını söyleyebilir. Örn. "Tüm şişeleri almadın. Bütün karpuzları toplamadın ve "Tüm çöpleri aldığınızda geri gelin".
Öğretmenler İçin	 Snap!'teki tüm aktivite
Araçlar ve	https://snap.berkeley.edu/project?user=mateja&project=Picking%
Kaynaklar	20up%20trash%20and%20cleaning%20the%20park
2 1 1 1	 Snap!'teki ek aktivite (çözümlü) <u>https://snap.berkeley.edu/project?user=mateja&project=Picking%</u> <u>20up%20trash%20and%20cleaning%20the%20park%20%2B%20Ad</u> <u>d.%20Task</u> Lajovic, S. (2011). Scratch. Nauči se programirati in postani računalniški maček. Ljubljana: Pasadena. Vorderman, C. (2017). Računalniško programiranje za otroke. Ljubljana: MK.
Oğrenciler İçin	 Snap!'teki yarı tamamlanmış aktivite
Kaynaklar ve	https://snap.berkeley.edu/project?user=mateja&project=Picking%
Materyaller	20up%20trash%20and%20cleaning%20the%20park%20-%20Part
	 Öğrenci için talimatlar (C4G2_InstructionsForStudent.docx)




Öğrenme Senaryosu 10 - Kedileri Beslemek

Öğrenme Senaryosu	Kedileri Beslemek
Adı	
Geçmiş Programlama	 Koşullu ifadeler (if, if-başka blokları)
Deneyimi	 Metni yazdırma (söyle bloğu)
Öğrenme Çıktıları	Genel Öğrenme Çıktıları:
	 değişken değerini ayarlama ve artırma döngünün içine / dışına değişken değer atama döngü rastgele sayılar dizi birleştirme operatörler: mantıksal, aritmetik, giriş
	Algoritmik düşünceye odaklı özel öğrenme çıktıları:
	 Öğrenci, n kez tekrarlama döngüsünü kullanma durumunu tanır.
	• Öğrenci, döngünün her yinelemesinde ve döngüden önce bir kez
	değer atama arasında ayrım yapar.
	 Öğrenci, bir oyuncudan numara almak için giriş bloğunu kullanır.
	 Öğrenci doğru cevabı üretmek için aritmetik operatörleri nasıl
	kullanacağını bilir.
	• Öğrenci, oyuncu girdisinin doğruluğunu kontrol etmek için if - else
	cümlesini kullanır ve uygun bir yanıt verir.
	 Öğrenci doğru cevapları saymak için bir değişkeni nasıl kullanacağını
	bilir.
Amaç, Görevler ve	Kısa Açıklama: Oyuncunun on adet çarpma işlemi yapması ve doğru
Faaliyetlerin Kısa	cevapları sayması gereken bir oyun programlayın.
Tanıtımı	<i>Görev:</i> Barınak bekçisi Martha'nın oyuncudan belirli bir odada
	besleyebileceği kedi sayısını birkaç defa soracağı etkinliği
	programlayın. Sayı, kâselerin sayısına ve boyutuna bağlıdır. Her oda
	için bu iki numaranın rastgele atanması gerekir. Ayrıca doğru cevapları
	sayacak bir sayacımız da olmalı. İlk sığınak bekçisi, oyuncuya görevi





	açıklamak zorundadır. Ardından oyun başlar. Kedi sayısını 10 defa
	sorduğunda oyun biter. Girilen numaranın doğru olup olmadığı her
	seferinde yanıt vermelidir. Aktiviteden sonra oyuncunun ne kadar
	başarılı olduğunu özetlemesi gerekir. Oyuncunun kaç kez doğru cevap
	verdiğini ve kaç kez yanıldığını söyler.
	Öğrencilere, bir döngü içinde çok değişkenli rastgele değer ataması
	kavramı ve bir döngü dışında yaptığımızdan nasıl farklı olduğu
	anlatılacaktır. Ayrıca doğru oyuncu girdilerinin nasıl alınacağını, test
	edileceğini ve sayılacağını da öğrenecekler.
Faaliyetin Süresi	45 Dakika
Öğrenme ve Öğretme	Aktif öğrenme, işbirlikçi öğrenme, problem çözme
Strateji ve Metotları	
Öğretme Formları	Ön öğrenme
	Bireysel çalışma/İkili çalışma/grup çalışması
Öğretme Özeti	(Motivasyon-Giriş, Uygulama, Düşünme ve Değerlendirme)
	Barınak bekçisi on farklı odada kedilerini beslemeye çalışıyor. Her
	odada rastgele sayıda (2'den 10'a kadar) farklı boyutlarda (1'den 5'e
	kadar) kâse bulunur, ancak her odada tüm kâseler aynı boyuttadır.
	Kâsenin boyutu, ondan kaç kedinin yiyebileceğini söyler. Örneğin kâse
	boyutu 3 ise, bu 3 kedinin ondan yiyebileceği anlamına gelir. Her
	odada besleyebileceği kedi sayısını bulmaya yardım edin.
	[Adım 1]
	İlk önce öğrencilere oyun için ilginç bir arka plan tasarlamaları
	talimatını veriyoruz. Zaman kazanmak istiyorsak, onlara biz verebiliriz.







[Adım 2]

Varsayılan kaplumbağa karakteri için kedi barınağı koruyucusunu temsil edecek yeni bir kostüm seçmemiz gerekiyor.





Gerekli değerleri saklamak için üç değişkene ihtiyacımız var.

1) Doğru cevapların sayısını saklamak için

2) Her odadaki kâse sayısı için rastgele değer atamak için (2-10)

3) Kase kapasitesine (1-5) rastgele değer atamak için.

Doğru cevap sayacı 0 olarak ayarlanmalıdır. Diğer ikisinin döngüden önce ayarlanması gerekmez. Çünkü döngünün her yinelemesinde bunlara yeni rastgele değerler atayacağız. Odaları da saymak istiyoruz. Ancak saymak için özel bir değişkene ihtiyacımız yok. *For döngüsü (for loop)* ile aynı değişkeni kullanacağız. Numarası 1 değerine başlatılacak ve ardından 10 değerine ulaşılana kadar her yineleme için 1 artırılacaktır. Bu, oda sayımını kopyalar.







[Adım 4]

Daha sonra oyuncu için talimatları programlamalıyız. Bunu Looks / say [string (Görünümler/söyle ..[string]) ve wait [n] seconds ([n] saniye bekle) blokları kullanarak yaparız.



[Adım 5]

Öğrencilerle her odada olacak ve dolayısıyla aynı olacak eylemlerin neler olduğunu tartışıyoruz. Bunlar, döngünün her yinelemesinde yürütülecek döngü bloğunun içine yerleştirilmesi gereken komutlardır. Öncelikle, o odadaki kâse sayısı ve kâse boyutu (1-5) için rastgele bir değer (1-10) atamamız gerekecek. Sonra bir oyuncuya o odada kaç tane kedi besleyebileceğimizi sormamız gerekecek. Cevabın doğruluk için test edilmesi gerekecek. Uygun bir yanıt vermemiz ve doğru olup olmadığını hatırlamamız gerekecek (doğru cevap sayacı). Her yinelemenin sonunda oda numarasını da 1 artırmamız gerekecek.

[Adım 6]

Kâselerin sayısı ve büyüklükleri için rastgele değerler atamak için, Değişkenler / set [seçenekler] (Variables/set [options]) değeri ile Operatörler / rastgele[n] ile [m] seç (Operators/pick random [n] to [m]) kullanacağız.

set number_of_bows to pick random (2) to (1) set bowl_size to pick random (1) to (5)





[Adım 7]

Oyuncuya Algılama / [dizi] (Sensing/ask [string]) ve bekleme bloğu (wait block) içinde besleyebileceğimiz kedi sayısını sormak istiyoruz. Aksi takdirde belirli bir saniye boyunca görüntülenecek ve ardından yeni bir metin satırıyla güncellenecektir. Bu şekilde oyuncular mevcut odada kaç kâse / beden olduğunu kolayca unutabilirler. Bir metin kombinasyonundan ve değişkenlere referanslardan oluşacak bir dizge oluşturmak için Operatörler/birleştirme [dize1] [dize2] (Operators/join [string1][string2]) bloğu kullanıyoruz. Tüm cümleye uyması için bu bloğu genişletmemiz gerekecek.



[Adım 8]

Bu uzun dizgiyi *Sense/Ask [string]* içine koymalı ve oyuncudan cevabı alabilmek için blok beklemeliyiz.



[Adım 9]

Oyuncu cevapladığında doğruluğunu kontrol etmeliyiz. Yalnızca iki olası durum vardır, oyuncu doğru veya yanlış olabilir, bu yüzden If-Else bloğunu kullanacağız. Doğru cevap, kâse sayısı ile kâse büyüklüğünü çarpmanın değeridir. Oyuncunun cevabının bu sayıya eşit olup olmadığını kontrol etmeliyiz. Cevap doğruysa, doğru cevap sayacını 1 artırıp yanıt veriyoruz. Değilse, sadece yanıt veriyoruz. Doğru cevap sayacından hesaplayabildiğimiz için yanlış cevapları saymak zorunda değiliz.





f answer) = number_of_bowls × bowl_size
change correct_answers by 1
say Greatl'Your'answeris'correct! for (2) secs
else
say Thisrisrhottherightnumberoficats. for 2 secs
say join The correct answer is: number_of_bowls × bowl_size +cats. •
for 2 secs
say Try to guess the right number in the next room! for 2 secs

[Adım 11]

Şimdi bir döngü seçmemiz gerekiyor. Daha önce de belirtildiği gibi, döngüye göre seçmek en mantıklısıdır çünkü yineleme için kullanılan değişken odaların sayımını kopyalar.

[Adım 12]

Döngü durduğunda oyun biter. Oyuncu başarısı hakkında bilgi vermemiz gerekmektedir. Doğru cevap sayısı, doğru cevap sayacında saklanır. Yanlış cevapların sayısı hesaplanabilir.





[Nihai Kod]





	when clicked
	set correct_answers to 0
	repeat 10
	set number_of_bowls = to pick random 2 to 10
	set bowl_size to pick random 1 to 5
	ask join There'ares number_of_bowls bowls. The bowls been size and
	wait
	IT Tanswer = number_of_bowls × bowl_size
	change correct_answers by 1
	else
	say Thisisinotitheiright number of cats. for (2) secs
	say The game is over. for (2) secs
	say join You'answered'correctly correct_answers time(s) • for 5 secs
	say join You'were'wrongs 10 - correct_answers time(s) 10 for 5 secs
Öğrətmənlər İçin	 Snantoki tüm aktivito
Ogretineniei için	
Araçlar ve Kaynaklar	https://snap.berkeley.edu/project?user=zapusek&project=cat feeding_2
	• Lajovic, S. (2011). Scratch. Nauči se programirati in postani
	računalniški maček. Ljubljana: Pasadena.
	 Vorderman, C. (2017). Računalniško programiranje za otroke. Ljubljana: MK.
Öğrenciler İçin	Snap!'teki şablon
Kaynaklar ve	https://snap.berkeley.edu/project?user=zapusek&project=cat_feedin
Materyaller	g_template
	 Öğrenci için talimatlar (C4G2_InstructionsForStudent.docx)





Öğrenme Senaryosu 11 - Bir barınaktaki kedi sayısını tahmin etmek

Öğrenme Senaryosu	Bir barınaktaki kedi sayısını tahmin etmek
Adı	
Geçmiş Programlama	 koşullu ifadeler (Eğer bloğu (if block))
Deneyimi	 metni yazdırma (Söyle Bloğu Say Block))
Öğrenme Çıktıları	Genel Öğrenme Çıktıları:
	 rastgele değerler
	 değişken atama
	• kullanıcı girişi
	 döngüye kadar tekrarlayın
	karşılaştırma operatörleri
	● sayaç
	Algoritmik düşünceye odaklı özel öğrenme çıktıları.
	 Öğrenci değişkene rastgele değer atar.
	 Öğrenci, bir oyuncudan numara almak için giriş bloğunu kullanır.
	 Öğrenci, oyuncudan tekrar tekrar sayıyı girmesini ve bir değer testi
	gerçekleştirmesini istemek için döngüye kadar tekrarla kullanır.
	• Oğrenci if cümle ve karşılaştırma operatörleriyle değer testi yapar
	ve uygun yanıt verir. • Öğrangi ayunun hitin hitmadiğini tast atmak isin takrar döngüsünün
	• Ogrenci öyünün bitip bitmedigini test etmek için tekrai döngüsünün kosulunu belirler
	 Öğrenci, dolavlı olarak kosullara bağlı olduğu icin ovunun bittiğinde
	test etmesi gerekmediğini fark eder.
	• Öğrenci, oyuncu tahminlerini saymak için bir sayaç uygular ve her iki
	olası sonucu ayırt etmek için nihai değeri kullanır.
Amaç, Görevler ve	Kısa açıklama: Başlangıçta 1'den 100'e kadar rastgele bir sayının bir
Faaliyetlerin Kısa	değişkene rastgele atanacağı basit bir oyun programlayın. Oyuncu
Tanıtımı	sayıları yazarak tahmin etmeye çalışacaktır. Girdi numarası: rastgele
	değerden daha fazla, daha az veya eşit ise bir yanıt alacaktır.
	Görev: Kedi barınağı Martha'yı rastgele olarak kedi sayısını
	belirlemesi, oyuncudan ismini sorması ve ardından görevi ona
	açıklaması için programlayın. Daha sonra Martha, oyuncuyu ismiyle
	selamlamak ve ardından tekrar tekrar bir numara sormak zorundadır.
	Oyuncu tahminini girdiğinde şu şekilde yanıt vermelidir:





	1) Giriş sayısı gerçek sayıdan düşükse, "kedi sayısı daha fazla"
	2) Giriş sayısı gerçek sayıdan yüksekse,: "Kedi sayısı daha az"
	3) Giriş numarası doğruysa, "Mükemmel, doğru sayıyı tahmin ettiniz".
	Her oyuncunun denemesini sayacak bir sayaç programlayın. Oyuncu
	doğru sayıyı tahmin ettiğinde deneme sayısının 5'ten az olup
	olmadığını kontrol etmelisiniz. Bu durumda oyuncu kediyi alır. Aksi
	halde almaz.
	Amaç: Öğrencilere döngüye kadar tekrar etmeleri ve oyunu
	durduran koşulu örtük olarak izlemek için koşulun nasıl ayarlanacağı
	tanıtılacaktır. Ayrıca değişkenleri farklı durumlarda nasıl
	kullanacaklarını da öğreneceklerdir. Örneğin, rastgele bir değer
	ayarlamak, sayaç olarak veya oyuncuların girdisini almak gibi.
Faaliyetin Süresi	45 dk.
Öğrenme ve Öğretme	Aktif öğrenme, işbirlikçi öğrenme, problem çözme
Strateji ve Metotları	
Öğretme Formları	Ön öğrenme
	Bireysel çalışma/İkili çalışma/grup çalışması
Öğretme Özeti	(Motivasyon-Giriş, Uygulama, Düşünme ve değerlendirme)
	Kedi barınağı bekçisi Martha, barınağında bulunan kedilerin tam
	sayısını tahmin etmenizi istiyor. Sayı 1 ile 100 arasında herhangi bir
	yerde olabilir. Oyuncu sayıyı yazdığında, mevcut giriş sayısı doğru
	sayıda kedi sayısından az, daha fazla veya ona eşitse yanıtlar. Bir
	oyuncu beş denemeden daha az denemede kedi sayısını tahmin
	ederse, kediyi alır, aksi halde tekrar oynaması istenir.
	[Adım 1]
	İlk görev, oyun için ilginç bir arka plan oluşturmaktır. Öğrenciler
	kendileri çizebilir veya internetten ücretsiz lisanslı görüntüleri
	kullanabilir. Zaman kazanmak için arka planı önceden hazırlayabiliriz.







[Adım 2]

Varsayılan kaplumbağa karakteri için kedi barınağı koruyucusunu temsil edecek yeni bir kostüm seçmemiz gerekiyor.



[Adım 3]

Kedi sayısı rastgele atanmışsa, bu oyunun bir kereden fazla oynanmasının ilginç olabileceğini öğrenciler ile tartışıyoruz. Bu rasgele sayıyı tahmin etmek için *sayı karşılaştırma tahminin (guess numbers comparisons)* kullanılabilmesi için, onu bir değişkende de saklamamız gerekir. Değişkenler artık (listelerin kavramını henüz bilmediklerini varsayıyoruz) Snap'te belirli bir değeri hatırlamanın tek yoludur. Bu, program başladığında gerçekleşmelidir (Olay/Yeşil bayrak tıklandığında (Event/When green flag is clicked) komutlarını hatırlayın).



[Adım 4]

Barınak bekçisi, oyuncudan onu selamlamak için adını sorar. Bu, Sense





/ ask [string] ve wait block ile yapılır. Oyuncu cevabı, answer adlı yerleşik bir değişkende saklanır. Onu selamlamak için, değişken cevapta saklanan dizeyi bir miktar selamlamayla birleştirmeliyiz. Bu Operatörler (Operators)/join [string1] [string2] bloğu ile yapılır. Metni görüntülemek için, n saniye için Looks /say [string] (Looks/say [string] for n seconds) bloğunu kullanırız. Oyunu oynamak için talimatlar yazmak için de bu blokları kullanıyoruz. Metnin gösterim süresine dikkat etmenin önemli olduğunu da vurgulayabiliriz.



[Adım 5]

Öğrencilerle, doğru sayıyı bulmak için oyuncuların kaç kez tahmin etmek zorunda kalacağını tahmin etmenin mümkün olmadığını tartışıyoruz.

Çok şanslı ise ilk denemesinde tahmin edebilir, belki 5 belki çok daha fazla tahminde bulunmak zorunda kalabilir. Bunu hiç bir zaman bilemeyiz. Verilen görev için doğru döngüyü seçmemizin nedeni budur. Barınak bekçisi, oyuncu doğru sayıyı tahmin edene kadar tekrar tekrar bir numara istemeli ve uygun bir yanıt vermelidir.

İstenilen yürütmeyi gerçekleştirebileceğimiz tek döngü *until[condition] (___e kadar [koşul])* döngüsüdür.

Oyuncu doğru yanıt verinceye kadar döngüyü devam ettirmeliyiz. Yani değişkende bulunan cevap ile cat_number (kedi sayısı) değişkeni içindeki sayı aynı oluncaya kadar.

repeat until answer = number_of_cats





[Adım 6]

Sonra, öğrencilere döngü gövdesine (loop body) girecek komutların ne olduğunu sormalıyız. Oyuncu doğru sayıyı tahmin edene kadar tekrarlanacak etkinlik veya komutlar nedir? İlk olarak, oyuncudan bir sayı girmesini istemeliyiz, sonra bu sayının değerine göre bir yanıt vermeliyiz.



[Adım 7]

Öğrencilerle açıklanması veya tartışılması gereken son şey, bu döngünün ne zaman biteceği ve bunun ne anlama geldiğidir. Oyuncu cevapları kedi sayısına eşit olduğunda, döngü gövdesindeki her iki koşul da yanlış olacaktır. Bu nedenle döngü, döngü koşulunu kontrol ederek bir sonraki yinelemeye gidecektir. Bu sefer koşul doğru olacak, böylece döngü sona erecek ve döngüyü izleyen komutlar çalıştırılacak. Başka bir deyişle, döngü sona erdiğinde oyuncunun sayıyı doğru tahmin ettiğini biliyoruz. Şimdi buna göre cevap verebiliriz.







[Adım 9]

Sayaç rolüne sahip olacak yeni bir değişken oluşturmalı ve onu 0 değerinden başlatmalıyız. Değişken başlatmanın önemini ve değeri belirlemek ile onu artırmak arasındaki farkı öğrencilerle tartışıyoruz. Bir değişkenin değerini belirlediğimizde, önceki değer kaybolur. Bu bir sayaç için uygun değil. Değişken değerini bir sayı kadar artırırsak, o sayıyı daha önce değişken olan değer ne olursa olsun ekleriz. Bu durumda tam olarak istediğimiz şey budur. Bir oyuncu her yeni sayı girdiğinde, onu 1 artırmak istiyoruz.

[Adım 10]

Doğru cevabın ardından, oyuncunun kediyi alıp almayacağına karar vermek için sayaç değişkeninin değerini kontrol etmeliyiz. Snap'in mantıksal operatörlerden yalnızca *"daha az (<)"*a sahip olduğundan yani *"daha az veya eşit"* operatörlere sahip olmadığından, bir oyuncunun kediyi alıp alamayacağına karar verebilmemiz için *cat_counter <6 '*yı kullanabiliriz. Bu aynı zamanda If-Else koşul bloğunu

86





kullanmak için iyi bir örnektir çünkü aslında biz iki durum arasında bir
ayrım yapmış olduk.
[Nihai Kod]
when M clicked
set try_counter to D
set number_of_cats to pick random (1) to (100)
say Hirkam/Martha./Cat/Sheltenkeepert for 2 secs
ask Whatisyourname? and wait
say join 10 name to for 2 secs
say Inhaverarlittlergamerfor yourtoday for 3 secs
say Ifyourcanguess how many cats do Phave in my shelten in Stries. for (4) secs
say Invikgiveryouronel for (2) secs
say Iralways/haverat/least/1/Catrand/the/shelter/max/capacity/is/100/cats. for 5 secs
repeat until answer = number of cats
change ty counter by
if answer < number_of_cats
say No.nol/have/more/Cats/than/that for (2) secs
if answer > number of cats
say Ithaverlessicats for 2 secs
say Awsomell Yourguessed the right number of cats for 2 secs
if try_counter < 3
else
say Yourhavertoormanytriestorgetrarcat. Butryourcan always play again for 4 secs
[Aktivitelerin Temel Versiyonu]
Zaman kazanmak için senaryonun temel versiyonunu kullanabiliriz.
Temel bir versiyonda tüm temel kavramlar dâhil edilmiştir. Yukarıda
açıklanan diğer işlevler daha sonraki yükseltmeler olarak kullanılabilir.





	<pre>when w clicked set try_counter = to 0 set number_of_cats = to pick random 1 to 100 repeat until answer = number_of_cats ask How:many:cats:do:you:think!!have? and wait if answer < number_of_cats say No.ino!have:more:cats:than:that! for 2 secs if answer > number_of_cats say !have!less:cats. for 2 secs</pre>
	say Awsome!!! Yourguessed the right number of cats! for (2) secs
Öğretmenler İçin	• Snap! te yer alan tüm aktivite
Araçlar ve Kaynaklar	<u>https://snap.berkeley.edu/project?user=zapusek&project=cat</u> <u>s_in_a_shelter</u>
	 Lajovic, S. (2011). Scratch. Nauči se programirati in postani računalniški maček. Ljubljana: Pasadena.
	 Vorderman, C. (2017). Računalniško programiranje za otroke. Ljubljana: MK.
Öğrenciler İçin	 Snap!'teki şablon
Kaynaklar ve	https://snap.berkeley.edu/project?user=zapusek&project=cats in a
Materyaller	shelter template
	 Öğrenci için talimatlar (C4G2_InstructionsForStudent.docx)





İleri Seviye Öğrenme Senaryoları

Öğrenme Senaryosu 12 - Sağlıklı yiyecekler yakalamak

Öğrenme Senaryosu	Sağlıklı yiyecekler yakalamak
Adı	
Geçmiş Programlama	Karakter için test ekleme
Deneyimi	Karakter gösterme ve gizleme
	Bir yönü göster kullanma
	Rastgele özelliğini kullanma
	Noktaları saymak için değişkenleri kullanma
	Döngü tekrarını kullanma
	Sonsuza kadar döngü kullanma
	Koşul ifadeleri kullanma
Öğrenme Çıktıları	Genel Öğrenme Çıktıları:
	 Değişkenler Koşullu ifadeler Döngü Bir yönü göster
	• Rastgele (random)
	Algoritmik düşünceye odaklı özel öğrenme çıktıları:
	 Öğrenci, oyunun kız konuşmayı bitirmeden başlamasını
	önlemek için değişken kullanır (isteğe bağlı).
	 Öğrenci, yiyeceğin hareket etmeye başlayıp başlayamayacağını
	kontrol etmek için (bir değişken yardımıyla) if/eğer ifadesini
	kullanır.
	 Öğrenci, puanlar 5'ten az olana kadar yiyeceğin hareketi için
	döngü tekrarını kullanır.
	 Öğrenci aşağı hareket eden karakterler için 180 (aşağı) yönünü
	kullanır.
	 Öğrenci, adım sayısını seçmek için rastgele kullanır.
	 Öğrenci rastgele pozisyona geçmek için rastgele kullanır.
	 Öğrenci x (rastgele), y (sabit) konuma (isteğe bağlı) geçmek için





	rastgele kullanır.
Amaç, Görevler ve	Kısa açıklama: Kız yiyecek yakalayacak ama dikkatli olmalı, sadece
Faaliyetlerin Kısa	sağlıklı yiyecekler puan kazandırır!
Tanıtımı	Görevler: Öğrenciler iki farklı karakter programlamak zorundadır.
	Talimatlar veren, oyuna başlamak için ne yapması gerektiğini
	söyleyen ve puanları sayan bir kız ve ekranın üstünden rastgele
	düşen yiyecekler.
	Ek olarak, öğrenciler bir kız konuşmayı bırakmadan önce yemek
	hareketini önlemek için bir değişken ve if/eğer ifadesi
	ekleyebilirler.
	Amaç: Öğrenciler, X Adımlar için rastgele hareket etmeyi ve bir
	konum seçmeyi ve ayrıca diğer olayları önlemek için değişkenleri
	ve koşulluları nasıl kullanacaklarını öğrenecekler.
Faaliyetin Süresi	45 dakika
Öğrenme ve Öğretme	Aktif öğrenme, işbirlikçi öğrenme, problem çözme
Strateji ve Metotları	
Öğretme Formları	Bireysel Çalışma / İkili Çalışma
Öğretme Özeti	(Motivasyon-Giriş, Uygulama, Düşünme ve değerlendirme)
	Kız yiyecek yakalıyor. Her sağlıklı besin 1 puan kazandırırken
	sağlıksız her besin 1 puan kaybettir. Oyun bir kız tarafından
	verilen bazı talimatlarla başlar. Sonra kız kaybolur ve yiyecek
	belirir. Oyuncu 5 puan topladığında yiyecekler kaybolur ve bir kız
	tekrar ortaya çıkar.







[Adım 1]

Bu aktivite bireysel çalışma veya ikili çalışma olarak planlanmıştır. Öğretmen birkaç ipucu verip zor kısımları açıklar ve gereken durumlarda yardım eder.

Öğrencilere başlangıçta

- Arka plan
- Kız karakteri

verilir.

Öğrenciler arka planı seçer ve bir ana karakter ekler. Örneğin, bir kız. Kız başta bazı talimatlar verir ve sonra saklanır. Önceki aktivitelerden gördüğümüz gibi *Bayrak Tıklandığında Göster (show when the flag is clicked)* bloğu eklemek iyi olacaktır (tekrar oynarken, hareketli grafik gizli kalırsa).

Aşağıdaki gibi bir kod olabilir.







Bu karaktere daha sonra döneceğiz. Şimdi bir meyve için bir kod yazalım. [Adım 2] Öğrenciler yeni bir karakter (sağlıklı bir yiyecek) ekler. Örneğin, bir elma. İlk olarak yukarıdan aşağıya doğru bir karakter hareketi programlarlar. Bu nedenle aşağıdaki blokları seçerler. point in direction 180 move 💮 steps Elmalarının baş aşağı olmasını istemiyorlarsa, üçüncü seçeneği yani döndürme (don't rotate) seçeneğini seçebilirler. Apple draggabl Costumes Sounds Bir oyunu daha ilginç kılmak için adım sayısı rastgele seçilebilir. Böylece hız her zaman aynı olmayacaktır. Örneğin, move pick random (1) to (2) steps Sıradaki Adım, elma ekranın altına geldiğinde ne olacağını düşünmek. Bu durumda öğrenciler if ifadesinin (if statement) kombinasyonunda bir dokunma kenarı (touching edge) bloğu kullanabilirler. Elma kenara temas ederse, rastgele bir konuma hareket ettirilecektir. Hareket blokları bize bir sonraki bloğu sunuyor. go to random position Bu komut rastgele olarak x herhangi bir y koordinatını seçer ve elma ekranın herhangi bir yerinde görünebilir (resimdeki kırmızı noktalara

bakın).





Elmanın her zaman ekranın üst kısmında görünmesini istiyorsak y değeri sabitlenebilir ve sadece x değeri rastgele seçilir. Aşağıdaki kodla elma her zaman ekranın üst kısmında görünecektir (resimdeki kırmızı noktalara bakın). go to x: pick random 200 to 200 y: 150 [Adım 3] Öğrenciler artık saymak için kullanacakları değişken puanlar (points) oluşturabilirler. Puanlar başlangıçta 0 olarak ayarlanmalıdır (kız karakterinin üzerinde). t points to D [Adım 4] Elmanın tekrar tekrar hareket etmesini istiyorsak, bir döngüye ihtiyacımız var. Öğrenciler bir koşul belirleyene kadar döngü tekrarını kullanabilir. Örneğin, 5 puana ulaştıklarında oyunun bitmesini isteyebilirler. Yani koşul puan = 5 (points = 5) olacak ve koşul yanlış olana kadar döngü tekrarlanacaktır. Koşul doğru olduğunda, bu oyuncu 5 puana ulaşır, döngü durur. repeat until 5 = points [Adım 5] Başta elmanın gösterilmesini istemiyoruz ama kız ona talimat verdikten sonra gösterilebilir. Öğrenciler elmayı tuşa basıldığında gösterecek şekilde programlayabilirler. Tabii ki, döngü tekrarından önce bir blok gösterisi eklemeleri ve ondan sonra gizlemeleri gerekiyordu. Şimdilik kodun tamamı şöyle görünüyor:







[Adım 6]

Elmaya tıklandığında (veya fare-enter yapıldığında) ne olur? Elma saklanmalı, puan sayılmalı, pozisyon değiştirmeli ve tekrar göstermelidir. Puanlar 1 ile değiştirilecek ve pozisyon için öğrenciler öncekiyle aynı kodu kullanabilirler.



[Adım 7]

Şimdi kıza geri dönelim.

Kız yeniden ortaya çıkmalı ve şöyle demeli: Tebrikler!

Sonsuz döngüye ihtiyacımız olacak, bu da 5 puana ulaşıp ulaşmadığımızı kontrol etmeye yarayacak. Eğer 5 puana ulaşırsak, kız ortaya çıkacak ve bir şeyler söyleyecek. Bundan sonra *hepsini durdur* (*stop all*) bloğu ekleyeceğiz. Öğrencilerin bu durmanın ne anlama geldiğini çözmelerine izin verin (kız durmaksızın sonsuza dek "Tebrikler ..." diyecek).



[Adım 8]

Oyunu tekrar oynarken, öğrenciler tüm talimatları ([Adım 1] 'den itibaren) zaten bildikleri için bu kısmı atlamak isteyeceklerdir.





Oyunun başlaması için önceden "S" ye basabilirler. Ancak kız hala konuşmaya devam edecek.

Bunu önlemek için, başlangıçta O olarak ayarlanması gereken başka bir değişken (*başlangıç (start*) adında) oluşturabiliriz. Ardından, kızın talimatlarından sonra, başlangıç değişkeni 1 olarak değişecektir.



Şimdi elmayı sadece başlangıç değişkeni 1'e eşitse başlayacak şekilde programlamalıyız ki öğrenciler bunu if ifadesiyle yapacaklar. Bununla, kız konuşmayı bırakmadan öğrenciler oyun oynayamayacak.

Oyunu tekrar oynadığımızda başka bir şey daha olabilir. Örneğin 3 puanımız varken oyunu durdurursak elma kaybolmayacaktır. Bu durumda oyuna yeniden başlarken, kız talimat vererek bitmeden elma görülecektir. Bunu istemediğimiz için oyunun başlangıcında elmanın gizlendiği bir kodu ekliyoruz.

Şimdi Elma kodu;





<pre>when s key pressed if start = if show repeat until 5 = points point in direction 180 point in direction 10 c 2 steps if touching edge 7 go to x: pick random 200 to 200 y: 150 hide when clicked</pre>
[Adim 9]
Öğrenciler artık elma karakterini birçok kez kopyalayabilir ve
kostümü değiştirebilirler (eğer isterlerse). Kod aynı olacak.
Tek değişiklik, sağlıksız yiyeceklerde tıklandığında 1 puan
kaybedecekler.
change points by
[Nihai Kod]
Кız





<pre>when clicked set points to 0 set start to 0 show say Help/me-to-catch-the-healthy-food for 4 secs say Healthy-food-brings-1-point,-unhealthy-1 for 4 secs say The-game-ends-when-you-reach-5-points for 4 secs say Press-S-to-start-the-game! for 2 secs hide set start to 1 forever ff points = 5 show say Congratulations!-You-have-collected-enough-healthy-food! for 3 secs stop at=</pre>
Elma
when s key pressed if start =] show repeat until 5 = points point in direction 180 move pick random 1 to 2 steps if touching edge 7 go to x: pick random 4200 to 200 y: 150 hide ti touching edge 7 bide ti touching edge 7 bide ti touching edge 7 bide ti touching edge 7 bide ti touching edge 7 bide ti touching edge 7 bide ti touching edge 7 bide ti touching edge 7 bide ti touching edge 7 bide ti touching edge 7 bide ti touching edge 7 bide ti touching edge 7 bide ti touching edge 7 bide
[Ek Görev] Öğrenciler isteklerine göre ek görevler ekleyebilir veya aşağıdaki
görevleri takip edebilirler.
 Oyunu bir kâse karakterinin yiyeceği yakalayacağı şekilde
değiştirin.
• Yeni bir karakter (kâse) ekleyin. Kendiniz çizin ya da çevrim içi
bulun.
 Kâsenin başlangıç konumunu ayarlayın (örneğin ekranın
altında). Kâsenin hareketi için bir kod yazın (yukarı ve aşağı da





	isterseniz sol ve sağ). Yiyecek karakterleri, kâseye dokunarak					
	rastgele bir yerde kaybolmalı ve yeniden ortaya çıkmalıdır (ve					
	daha önce olduğu gibi yiyeceğe fare tıklandığında değil).					
	 Kuralları değiştirin - bir oyuncu 20 puan aldığında 					
	(kazandığında) veya 3 sağlıksız yiyecek aldığında (kaybettiğinde)					
	oyunun bitmesine izin verin.					
	Oyunu daha ilginç hale getirmek için daha fazla yiyecek					
	karakterleri ekleyin.					
	• Bir oyuncu 5, 10, 15 puan skor elde ettiğinde kâse kostümünü					
	değiştirin.					
Öğretmenler İçin	Snap!'teki Tüm Aktivite					
Araçlar ve Kaynaklar	https://snap.berkeley.edu/project?user=mateja&project=					
	Catching%20healthy%20food					
	 Snap!'teki Çözümlü Ek Görevler: 					
	https://snap.berkeley.edu/project?user=mateja&project=					
	Catching%20healthy%20food%20%2B%20Add.%20Task					
	• Lajovic, S. (2011). Scratch. Nauči se programirati in postani					
	računalniški maček. Ljubljana: Pasadena.					
	• Vorderman, C. (2017). Računalniško programiranje za					
	otroke. Ljubljana: MK.					
Öğrenciler İçin	Snap!'teki Yarı Tamamlanmış Aktivite					
Kaynaklar ve	https://snap.berkeley.edu/project?user=mateja&project=					
Materyaller	C4G12 Catching%20healthy%20food%20-%20Part					
	Öğrenci İçin Talimatlar					
	(C4G2_InstructionsForStudent.docx)					
	 Images: bowl1.png, bowl2.png, bowl3.png, bowl4.png 					





Öğrenme Senaryosu	Hikâye Anlatma				
Adı					
Geçmiş	Karakteri gizleme ve gösterme				
Programlama	Koşulluları kullanma				
Deneyimi	Söyle (say) kullanma (Görünüşler grubundan)				
	saniye bekle'yi (wait forseconds) kullanma				
Öğrenme Çıktıları	Genel Öğrenme Çıktıları:				
	 Taşıma ve boyut değiştirme Yayın yapma Hikâye anlatımının yapısını oluşturma Sahnelerin arka planını değiştirme 				
	 Algoritmik düşünceye odaklı özel öğrenme çıktıları. Öğrenciler hikâyedeki karakterlerin diyaloglarını ve etkinliklerini planlar. 				
	 Oğrenciler, karakterler arasındaki diyalog için yayın göndermeyi kullanır. 				
	Öğrenciler, karakterler için hareket ettirme ve boyut değiştirmeyi				
	 Öğrenciler karakter göster ve gizle kullanır. Öğrenciler, karakter kodunu yeniden düzenler ve genişletir. 				
Amaç, Görevler ve	Kısa Açıklama: Tavşan, Alice				
Faaliyetlerin Kısa	Harikalar Diyarında'nın hikâyesini				
Tanıtımı	anlatıyor. Hikâye anlatımına Alice				
	Harikalar Diyarında etiketli arka				
	planda birkaç cümle ile başlar.				
	Alice'in hikâyesi ormanda başlar.				
	Alice yürüyor ve "Neredeyim?" Diye				
	merak ediyor. / Alice'in uzaklaştığını				
	fark etmek için, hareketle yavaş				
	yavaş boyutu küçülür /. Alice bir Then it doesn't matter which way you go."				
	kavşağa gelir ve bir ağaçta Cheshire				
	Kedisi'ni görür. Alice ve Cheshire				
	Kedisi arasında bir konuşma başlar.				





Konuşma	a resim	de su	nulr	nuş	tur.	

Görevler: Öğrenciler, bir bekleme bloğu aracılığıyla diyaloğu senkronize etmeye dayalı olarak Alice ve Kedi arasındaki toplantının öyküsünün kısa bir örneğini denemelidir. Daha sonra, yayın mesajlarını kullanarak hikâyenin ikinci bir versiyonunu gözden geçirirler. Mesajlaşma komutları girilir. Öğrenciler resimdeki metne göre karakterlerin kodunu tamamlarlar. Kediyle tanışmadan önce Alice'i ormanda hareket ettirerek sahne dekorunu değiştirmenin getirilmesi görevi karmaşıktır.

Amaç: Öğrenciler hikâye anlatımını nasıl planlayacaklarını, karakter etkinliklerinin ve sahne değişikliklerinin senkronizasyonu için yayın mesajlarının nasıl kullanılacağını öğrenecekler.

Faaliyetin Süresi	90 Dakika			
Öğrenme ve	Aktif öğrenme, oyun temelli öğrenme, problem çözme			
Öğretme Strateji ve				
Metotları				
Öğretme Formları	Bireysel Çalışma / İkili Çalışma/ Ön Tartışma			
Öğretme Özeti	(Motivasyon-Giriş, Uygulama, Düşünme ve Değerlendirme)			
	1- Öğretmen öğrencilerle Alice Harikalar Diyarında'nın hikâyesini tartışır			
	ve Alice'in Cheshire kedisi ile tanışmasının resmini gösterir. Alice'in			
	hikâyesinin kodlama kullanılarak yeniden oluşturulabileceğini açıklar.			
	Ardından öğrenciler, projeyi başlatmak ve sprite'ların kodlarına			
	bakmakla görevlendirilir.			
	https://snap.berkeley.edu/project?user=ddureva&project=Alice 1			
	Tartışma: İlk önce kim konuşmaya başlar? Alice ve Kedi ne zaman			
	konuşmaya dahil oluyor? Karakterlerin diyaloğunda neden			
	senkronizasyon yok?			
	Cevap: Karakterlerin her birinin "konuştuğu" ve "bir karakterin yanıtlarını			
	bitirmesini bekleyecek bir zaman aşımının olmaması" zamanlamanın			
	yanlış hesaplanmasından kaynaklanmaktadır.			





Kodlar yorumlanır ve tablo tamamlanır.



Karakter	Aktivite	Başlangıç	Bitiş	Süre
		Zamanı	Zama	
			nı	
Tavşan	Merhaba de! Alice ve	0	14	14
	Harikalar Diyarı'ndaki			
	maceralarını duydunuz			
	mu? Şimdi onun			
	hikâyelerinden birine			
	bakalım.			
Alice	Şunu söyle: Lütfen bana	9	21	12
	söyler misiniz, buradan			
	hangi yöne gitmeliyim?			
Kedi	Şunu söyle: Bu, nereye	10	20	10
	gitmek istediğinize			
	bağlıdır.			

Sonuç şudur: Saniye bekle (wait for... second) bloğu ile

senkronizasyon, hikaye anlatımında karakterlerin davranışlarında hatalara yol açabilir.

2. Öğretmen, proje kodunu başlatmak ve gözden geçirmekle görevlendirilir.

https://snap.berkeley.edu/project?user=ddureva&project=Alice 2

Şimdiye kadar bilinmeyen komutlar neler?











İsimlendirmenin yapılışı

when I receive	-
	any message Alice1 Cet1
	0990

2. Karakter tarafından alınması gereken mesajın seçilmesi.

3. Grup, resimdeki hikâyenin nasıl tamamlanacağını tartışır. Mesajlar nasıl adlandırılır? Örneğin, Kediden Alice'e veya Alice2'ye ve Alice'ten kedi veya kedi1'e mesaj gönderilmesi.

4. Öğrenciler hikâyelerindeki bu kısmı ikili gerçekleştirebilirler.

5. Öğretmen, hikâye anlatmanın genellikle sahne kostümlerinde bir değişiklik gerektirdiğini söyler. "Tavşan hikâyesini giriş zemininde başlatıp, aksiyonu Alice'in yürüdüğü ormana taşıyıp" Neredeyim? Diye merak ederek Alice'in hikâyesini daha eksiksiz hale getirelim. Alice sahneden hareket ederek uzaklaştıkça karakterin boyutu giderek küçülmelidir. Sonra kendini bir dönüm noktasında bulur ve Cheshire kedisini görür. Konuşma ikisi arasında başlar.













Alice ormanda yolundayken yürür ve "dolanır", bu nedenle daha fazla gerçekçilik için boyutu -% 10 azalır. Bu, *tekrar döngüsü (repeat loop)* kullanılarak 5 kez tekrarlanır.

Kavşağa ulaştığında, sahne "Cheshire Kedisiyle Buluş" mesajı ile değiştirilir. Bu mesaj aynı zamanda büyüklüğünü% 80'e düşüren Tavşan tarafından da alınır ve küçültülmüş hali ile hikâyeyi anlatmaya devam eder. Bu aşamada, dekorun bir parçası olduğu için kedi karakteri gösterilmez. Kedi1 (Cat1) mesajında görünür.

Öğretmen, harici bir grafik düzenleyici kullanarak kedinin dekordan çıkarıldığını açıklayabilir. (Ne yazık ki Snap!, Scratch 3.0'dan farklı olarak çok fazla grafik düzenleyici yeteneği sağlamaz).

Tavşan mesajının yayınlanmasının ardından Alice 1 hikâyesi Alice 2





	projesinde olduğu gibi devam eder.			
	3. Öğretmen, bir hikâye anlatmak için kişinin önce bir olay örgüsü icat			
	etmesi gerektiğini söyler. Hikâyenin senaryosunu açıklamak için ek bir			
	tablo kullanılabilir. (Ek 1) öğretmenin takdirine bağlı olarak, bitmiş tablo			
	verilebilir veya kısmen doldurulabilir ve öğrenciler, resimle yönlendirilerek			
	tamamlayabilir.			
	4. Öğrencilere ikişerli gruplar halinde, incelenen senaryoları açıklama ve			
	Alice_2 projesinin hikâyesini tamamlama görevi verilir.			
Öğretmenler İçin	Snap teki tüm aktivite			
Araçlar ve	https://snap.berkeley.edu/project?user=ddureva&project=Alice			
Kaynaklar				
Öğrenciler İçin				
Kaynaklar ve				
Materyaller	Would you tell me, please, which way 1 ought to go from here?" That depends a good deal on where you want to get to," said the Cat. "1 don't much care where," said AlfCe. Then it doesn't matter which way you go," said the Cat. "1 don't much care where," said AlfCe. Then it doesn't matter which way you go," said the Cat. • https://snap.berkeley.edu/project?user=ddureva&project=Alice 1 • https://snap.berkeley.edu/project?user=ddureva&project=Alice 2 • Öğrenci için talimatlar (C4G2_InstructionsForStudent.docx))			





Appendix 1. Story Plots/Scenarios

Ad	Tasarım	Etkinlik	Notlar	
1. Başlangıç	Alice in the Wonderland	Hikâye Yeşil bayrak	Bu arka planda, Tavşan hikâyeyi	
		tıklandığında	tanıtır.	
		yandaki sahneyle		
		başlar.		
	Phat/Add 30	Manzara, Tavşan	Bu arka planda, Alice sahnenin	
	12 Martin	girişini	ortasında konumlanmış	
		tamamladığında	görünür. "Neredeyim?" Diye	
		ortaya çıkar	merak ederek hareket etmeye	
		(Ormana git mesajı	başlar. Karakterinin boyutu	
2.0		gönderilmiştir).	kademeli olarak 5 defa %10	
2. Orman			küçültür. Yolun sonuna	
			ulaştığında (bir kavşakta), sahne	
			buluşma olarak değişir. (Alice	
			mesaj – yayın (message -	
			broadcast) gönderir. Cheshire	
			Kedisi ile Buluşma)	
		Alice'in Cheshire	Burada Alice ve kedi arka planın	
		Kedisi ile Toplantı	bir parçasıdır. Alice'in hareketli	
		mesajı alındığında	karakterini kullanmak için	
		görünür.	mesajdan önce, imajı dekordan	
			kaplayacak şekilde	
			konumlandırılır. Kedi karakteri	
3. Buluşma			daha sonraki bir aşamada	
			görünür.	
			Sahne değiştikçe Tavşan	
			hikâyeyi anlatmaya devam eder.	
			Daha sonra Alice ve Cheshire	
			Kedisi arasında bir konuşma	
			gerçekleşir.	







Karakter	Etkinlik	Arka Plan
Rabbit	Başlangıçta: - Merhaba! (2 sn.) Alice ve Harikalar Diyarı'ndaki maceralarını duydunuz mu? (6 sn.)	Aire is Steveranus start
	Şimdi onun hikâyelerinden birine bakalım! (6 sn.) "Ormana git" (<i>Go to forest</i>) mesajını gönderir.	
Alice	Başlangıçta: Sahneden gizlenir. Orta sahne konumunda ve % 100 boyutta, yeni arka planda görüntülenmeye hazır.	Aler is Structured start
Cat	Başlangıçta: Sahneden gizlenir. x: -74, y: 113 noktasında konumlandırılmıştır. (Konumlar, Kedi karakteri buluşma sahnesinde ayarlandıktan sonra önceden belirlenir).	start
Alice	"Ormana git" mesajını alır. Karakter sahnede gösterilir. 5 kez tekrarlanır: 1 saniye bekle; 5 Adım hareket; boyut küçültme (%-10 kadar değişiklik); merak ediyorum: Neredeyim ben? Bir sonraki dekor için hazırlanır: 5 saniye beklenir; karakterin boyutunu eski haline getirme (% 100 değişiklik) ve x: -187, y: -67 konumunda konumlandırma Mesaj Gönderir: "Cheshire Cat ile buluşma"	forest
Rabbit	Herhangi bir eylem yok. Yalnızca bir önceki dekordan görünür hale gelir.	forest
Rabbit	Mesajı alır: "Cheshire Kedisi ile buluşma" % 80'e yeniden boyutlandırır. Der ki: "Alice dönüm noktasında durur ve nereye gideceğini merak eder" (10 saniye).	meeting




	Der ki: "Alice Cheshire Kedisini ağaçta gördü" (8 saniye).	
	"Alice1" mesajını gönderir.	
	Alice1 mesajını alır.	
Alice	Öne doğru hareket eder (Bu gereklidir. Çünkü Kedi ondan sonra	1. 20
	görünür. Bbu da Alice'in ön katmanda değilse bir diyalog	meeting
	kutusunda görünmesini engeller).	
	Der ki: "Merhaba!" (2 sn.).	
	Der ki: "Lütfen bana buradan hangi yöne gitmem gerektiğini	
	söyler misin?" (10 saniye).	
	Kediye "Cat1" yayın mesajını gönderir.	
	Cat1 mesajını alır.	
Cat	Karakter sahnede gösterilir.	A CAN
	Der ki: "Bu, NEREYE gitmek istediğinize bağlı!" (10 saniye).	meeting
	"Alice2" mesajını gönderir.	
	"Alice 2" mesajını alır.	
Alice	Der ki:	A CAN
	<i>"Cat2"</i> mesajını gönderir.	meeting
	<i>"Cat2"</i> mesajını alır.	
	Der ki:	
Cat	<i>"Rabbit1"</i> mesajını gönderir.	meeting
and the	"Rahhit1" mesajuni alir	KARESPH
	Der ki: "Hikâvenin dersi nedir?" (8 sanive.)	ME .
Rabbit	Der ki: "Hangi voldan gideceğini bilmek icin önce hedefini	
	helirlemeli"	meeting





Öğrenme Senaryosu 14 - Çizim yapma

Öğrenme			
Senaryosu Adı			
	Karakter ekleme		
Geçmiş	Bir yöne bak komutu kullanma		
Programlama	Sayma noktası için değişkenleri kullanma		
Deneyimi	Döngü tekrarını kullanma		
	Koşul ifadeleri kullanma		
	Genel Öğrenme Çıktıları:		
Öğrenme Çıktıları	 Değişkenler Koşullu ifadeler Döngü Bir yöne bakma Operatörler – Dört İşlem Algoritmik düşünceye odaklı özel öğrenme çıktıları: Öğrenci, çizim yapmak için kalem kullanır. Öğrenci, çizim yapmak için döngüler kullanır. Öğrenci, çizim yaparken bir değişkenin değerini değiştirir. Öğrenci, sahnede nesneler çizmek için işaret kullanır. Öğrenci, hareketli grafiği kontrol etmek için yayını kullanır. Öğrenci, aşamayı değiştirmek için koşullu ifadeler kullanır. 		
	Kısa Açıklama: İklim çok değişti. Hava sanayi nedeniyle kirlendi. Hava		
	kalitesini iyileştirmek için ağaçların dikilmesi gerekiyor!		
	Görevler: Hava kalitesini iyileştirmek için öğrenciler, bir karakter		
Amaç, Görevler	programlayarak iki tür farklı ağaç çizmelidir - çam ve meşe ve bu tür		
ve Faaliyetlerin	ağaçları sembolize eden düğmeler. Bir düğmeye tıklandığında, belirli bir		
Kısa Tanıtımı	ağaç türü çizilir.		
	Amaç: Öğrenciler Snap!'te çizim yapmayı, rengi ve kalem kalınlığını		
	değiştirmeyi ve yeni bir olaya neden olan değişkenleri ve koşullu		
	ifadeleri nasıl kullanacaklarını öğrenecekler.		
Faaliyetin Süresi	45 dakika		
Öğrenme ve	Aktif öğrenme, oyun temelli öğrenme, problem çözme		





Öğretme Strateji		
ve Metotları		
Öğretme	Bireysel Çalışma / İkili Çalışma	
Formları		
Öğretme Özeti	Bireysel Çalışma / İkili Çalışma (Motivasyon-Giriş, Uygulama, Düşünme ve Değerlendirme) Oyunun başında sahnede iklim değişikliğine neden olan bir fabrika ve hava kalitesini gösteren bir değişken gösteriliyor. Hava kalitesini iyileştirmek için ağaçların dikilmesi gerekiyor. Çam ve meşe olmak üzere iki farklı ağaç türü çizilebilir. Bir çam eklendiğinde hava 3, meşe eklendiğinde hava 2 birim iyileştirilir. Hava kalitesi 10 birime ulaştığında sahne arka planı bir çayıra dönüşür. [Adım 1] Öğrenciler, arka planlardan (fabrika ve çimen) ve karakterlerden (kurşun kalem, çam ve meşe olarak adlandırılan) bir şablon içeren İklimi İyileştirme (Improve <i>the Climate</i>) programını açmalıdır. Ayrıca, yeni karakter olarak kalem eklemelidirler (önerilen karakter pencil a" (kalem a)). Karakter çok büyük olduğu için% 50'ye	
	düşürülmelidir. Kalemin başlangıç konumu (koordinatlar) da belirtilmelidir. Örneğin X = -10, y = -10.	























set clean air = to 🛛

Bir çam ağacı dikildiğinde hava 2 birim iyileşir. Bu nedenle, çamın her tıklanışında "temiz hava" değişkeninin değerini 2 değiştirecek olan bloğu çam serisine ekleyin.



Bir meşe her çekildiğinde hava 3 birim iyileşir. Bu nedenle "temiz hava" değişkeninin değerini çam her tıklandığında 3 değiştirecek olan bloğu meşe karakterine ekleyin.



[Adım 6]"Temiz hava" değişkeni 10'a ulaştığında, sahne çim olarak değişmelidir. Bu nedenle indirilen materyallerden sahne için yeni bir arka plan yani "çimen" ekleyin (arka plan indirilen materyallerden alınmıştır).







Sonra, > işlemini (operator) ekleyin.
when
Karakterin "temiz hava" değişkeni 10'dan büyük olduğunda "çim"
mesajını yayınlamasını (broadcasts) ayarlayın.
when clean air > 00 broadcast grass *
"Çim" mesajı alındığında kostümü "çim" olarak değiştirmek için sahneye
kodu ekleyin.
when I receive grass -
[İLAVE GÖREVLER]
Hava artık kirlenmediğinde ortaya çıkan hayvanları ekleyerek oyunu
yükseltebilirsiniz.
[NİHAİ KOD]
Çam
when I am clicked m broadcast_draw a pine
Meşe
when I am dicked -
x
when I am clicked - broadcast clean -
Kalem





	wind the first the state is the state i
	Sahne
	when a clicked switch to costume industry
Öğretmenler	Snap! project "Drawing":
İçin Araçlar ve	https://snap.berkeley.edu/project?user=tadeja&project=Improve%20the%20cl
Kaynaklar	<u>imate</u> (9.1.2020)
Öğrenciler İçin Kaynaklar ve Materyaller	 <u>Programming language Snap!: https://snap.berkeley.edu/</u> (9.1.2020) Öğrenci için talimatlar (C4G2_InstructionsForStudent.docx)



Öğrenme Senaryosu 15 - Fareyi yakalamak

Öğrenme Senaryosu	Fareyi yakalamak	
Adı		
	 Öğrenci bir arka plan ekleyebilir. 	
	 Öğrenci, yeni bir hareketli grafik ekleyebilir. 	
	 Öğrenci yeni bir ses ekleyebilir. 	
Goomis	 Öğrenci, karakterin nasıl bir şey söylemesini sağlayacağını bilir. 	
Programlama	 Öğrenci, animasyon yapmak için karakterin kostümünü nasıl değiştireceğini bilir. 	
Deneyimi	 Öğrenci, olayları kullanarak ok tuşlarıyla nesne hareketini gerçekleştirebilir ve kısıtlamaları dikkate alır. 	
	 Öğrenci, iki farklı durumu ayırt edebilir ve bunları mantıksal ifadelerle nasıl ifade edeceğini bilir. 	
	 Öğrenci, koşul ifadelerini nasıl kullanacağını bilir. 	
	Genel Öğrenme Çıktıları:	
	 Sonsuz döngü; Rastgele sayılar; Sayaç Zamanlayıcı. 	
Öğrenme Çıktıları	Algoritmik düşünceye odaklı özel öğrenme çıktıları:	
	 Öğrenci, karakterleri hareket ettirmek için sonsuz döngüyü kullanır. 	
	 Öğrenci, karakterin konumunu belirlemek için, karakterin rastgele hareket etmesi için ve karakterin rastgele dereceler ile dönmesini sağlamak için rastgele sayılar (random numbers) kullanır. 	
	 Öğrenci, yakalanan fareleri saymak için sayaç uygular ve oyuncunun ne kadar başarılı olduğunu ölçmek amacıyla sayaçtaki son değeri kullanır. 	
	 Öğrenci oyunun sonunu belirlemek için zamanlayıcıyı kullanır. 	
	Kısa Tanıtım: Oyuncunun (kedi) fareyi yakalaması gereken bir program yazar.	
Amaç, Görevler ve	Görev: Kedinin fareyi yakalayacağı etkinliği programlayın. Kedi, ok tuşlarına	
Faaliyetlerin Kısa	sahip bir oyuncu tarafından hareket ettirilecek ve fare rastgele hareket	
Tanıtımı	edecektir. Kedi fareye dokunduğunda, fare gizlenecek ve rastgele bir yerde	
	görünecektir. Ayrıca, kedinin fareyi kaç kez yakaladığını sayan bir sayaca	

Co-funded by the Erasmus+ Programme of the European Union





	sahip olmamız gerekir. Oyunu bitirmek için ayrıca bir zamanlayıcıya	
	ihtiyacımız var. Aktiviteden sonra kızın oyuncunun ne kadar başarılı	
	olduğunu özetlemesi gerekir. Bunu oyuncunun fareyi kaç kez yakaladığını	
	söyleyerek gerçekleştirebiliriz.	
	Amaç: Öğrencilere çok değişkenli rastgele değer atama kavramı	
	tanıtılacaktır. Öğrenciler, operatörleri (işlemler) nasıl kullanacaklarını /	
	rastgele [x] - [y] bloklarını nasıl seçeceklerini öğrenecekler.	
Faaliyetin Süresi	45 dk.	
Öğrenme ve		
Öğretme Strateji ve	Aktif öğrenme, işbirlikçi öğrenme, problem çözme, oyun temelli öğrenme.	
Metotları		
Öğretme Formları	Ön Öğrenme	
	İkili Calısma Grun Calısması	
	ikin çalışına, orap çalışınası.	
	(Motivasyon-Giriş, Uygulama, Düşünme ve Değerlendirme)	
	Motivasyon-Giriş	
	Oyunu göstererek öğrencileri motive ediyoruz. Onlarla bu oyunu	
	programlamaya nasıl başlayacaklarını tartışıyoruz. Öğrencilerle birlikte	
	adımların sırasını belirliyoruz.	
	Örneğin:	
	1. Arka planı seçin ve karakteri ekleyin.	
	2. Kediyi ok tuşlarıyla hareket etmesi için programlayın.	
Öğretme Özeti	3. Fareyi rastgele hareket edecek şekilde programlayın.	
	4. Fareyi kediye dokunduğunda gizlenecek (ve rastgele bir yerde görünecek)	
	şekilde programlayın.	
	5. Program sayacını ekleyin.	
	6. Zamanlayıcı ekleyin ve oyunun sonunu belirleyin.	
	7. Kızı ekleyin ve oyuncunun ne kadar başarılı olduğunu özetlemesi için	
	programlayın.	
	8. Kızı fareye dokunduğunda atlayacak şekilde programlayın.	
	9. Kedi / fare sesini ekleyin.	





10. Vb.

Öğrenciler adımlarda yardımcı olabilir veya oyunun kendi kurallarını koyabilirler (ancak adımları takip etmeleri gerekir).



Rastgele değer ataması için operatörü tanıtıyoruz.

pick random (1) to (10)

Öğrenciler, öğretmenin desteği ile aşağıdaki görevleri çiftler / gruplar halinde programlar.

Uygulama

[Adım 1]

İlk adım, oyunun arka planını belirlemektir. Öğrenciler çevrimiçi olarak ücretsiz resim ararlar. Ardından, yeni karakter eklerler - kedi ve fare.





[Adım 2]

Öğrenciler kediyi ok tuşlarıyla hareket etmesi için programlar. Burada, kedinin kenarlarda olması durumunda ne olacağını belirlemek zorundalar.





Co-funded by the Erasmus+ Programme of the European Union



[Adım 3]

Öğrenciler fareyi rastgele hareket edecek şekilde programlamalıdır. Bu durumda mantık, sonsuz döngüdeki farenin rastgele sayıda adım atması ve rastgele bir dereceye kadar dönmesidir. Öğrenciler bunu Hareket/hareket et [x] Adımlar (*Motion/move*[x]Adıms) bloğu ve Hareket/dönüş [x] (*Motion/turn*[x]degrees) derece bloğu ile yaparlar ve içine rastgele [x] ila [y] (*pick random*[x]to[y]) operatörünü yerleştirirler.



[Adım 4]

Sonraki adım, fareyi kediye dokunduğunda gizlenecek şekilde programlamaktır. Buradaki mantık, farenin kediye dokunduğunda rastgele bir yerde gizlenmesi ve görünmesidir. Bu durumda oyun farenin ilk yakalandığı anda bitmemektedir. Öğrenciler buraya kendi kurallarını ekleyebilirler. Her durumda rastgele [x] - [y] seçme (*pick random[x]to[y]*)





operatörünü kullanmaları gerekir.



[Adım 5]

Farenin kaç kez yakalandığını bilmek istememiz durumunda, bir sayaç eklememiz gerekir. Öğrenciler yeni bir değişken oluşturur "– score" (puanla) ve bunu kedinin koduna ekler. Oyunun başındaki skor her zaman sıfır olmalıdır. Öğrenciler bunu *Değişkenlerle/*[x]'e ayarla[değişken] (Variables/set[variable]to[x]) bloğu ile yaparlar. Skorun oyunculara gösterilmesini istiyorsak, öğrencilerin *değişken göster* [değişken] (show variable[variable]) bloğunu eklemesi gerekir. Daha sonra öğrenciler, kedinin fareye dokunup dokunmadığını kontrol etmek için yeni bir kontrol bloğu "(Kontrol / ne zaman) ((Control/when))" ekler. Kedi fareye dokunursa, sonuç 1 artar (Variables/change[score]by[x]).







[Adım 6]

Öğrenciler oyunun ne zaman biteceğini belirlerler. Bunu zamanlayıcıyı ekleyerek yapabilirler. Bir süre sonra (örneğin 30 saniye) fare ve kedi kaybolur. Puan değişkeni gizlenir ve oyun sona erer.



Öğrenciler bu blokları kedi ve farenin komut dosyalarına eklemelidirler.

[Adım 7]

Öğrenciler, oyuncunun ne kadar başarılı olduğunu özetlemek için kızı programlamalıdır. Eğer oyuncu fare yakalamazsa, kız "Sen hiç fare yakalayamadın!" der. Aksi takdirde, "Tebrikler! X fare yakaladınız!" der.



[İLAVE GÖREVLER]

Öğrenciler oyunlarına herhangi bir öğe ekleyebilir. Örneğin, bir fareye her dokunduğunda sıçrayan kız.





when clicked ga to x: -9 y: 100 set size to 100 % show forever if touching Mouse 7 switch to costume ballerina d else switch to costume ballerina a
Öğrenciler ses ekleyebilir. Örneğin, kedinin sesini eklerler. Fare
yakalandığında ses çıkar.
Düşünme ve Değerlendirme
Öğrenciler bu kodları ayarlayacaklar.
 Fare 20 den 60 adıma sonsuza dek gider. Fare kediye dokununca x = 100 konumuna gider. Fare sonsuza dek 90 derece döner. Vb.
[Nihai Kod]
Fare





Co-funded by the Erasmus+ Programme of the European Union

when the clicked show forever If touching Car ? ge to x: pick random 200 to 200 move pick random 10 to 100 ster If on edge, bounce wait 0.35 sters turn pick random 20 to 90 deg If on edge, bounce	y: pick random 200 to 200 ps	when timer = 0
winn (cissed show winn (cissed show show (in cissed show show (in cissed show interesting (in cissed show	Score or Control Score or Control und Water und Water und Water und Water und Water und Water und Water und Water und Water und Water und Water und Water und Water und Water und Water und Water und Water und Water und Water	
KIZ	when timer = 30 set size to 150 % go to x: 185 y: 65 if Score = 0 say You'didn't catch any micel else say Congratulations! for 2 wait 1 secs say join You'caught Score	for ③ secs secs micel ↔ for ③ secs





	Arka plan
Öğretmenler İçin Araçlar ve Kaynaklar	 Snap!'teki tüm aktivite <u>https://snap.berkeley.edu/project?user=tadeja&project=Catch%20th</u> <u>e%20mouse</u> Ücretsiz resim için: <u>https://pixabay.com/</u> Lajovic, S. (2011). Scratch. <i>Nauči se programirati in postani</i> <i>računalniški maček</i>. Ljubljana: Pasadena. Vorderman, C. (2017). <i>Računalniško programiranje za otroke</i>. Ljubljana: MK.
Öğrenciler İçin Kaynaklar ve Materyaller	 Snap!'teki şablon <u>https://snap.berkeley.edu/project?user=tadeja&project=Catch%20th</u> <u>e%20mouse 0</u> Ücretsiz resim için: <u>https://pixabay.com/</u> Öğrenci için talimatlar (C4G2_InstructionsForStudent.docx)

Öğrenme Senaryosu 16 - Piknik için yiyecek satın almak

Öğrenme Senaryosu	Biknik icin vivocok satun almak	
Adı	Piknik için yiyecek satın almak	
	Karakter için metin ekleme	
	Karakter gösterme ve gizleme	
Geçmiş Programlama	Operatörleri kullanma	
Deneyimi	Değişkenleri kullanma	
	Dize birleştirme kullanma	
	Koşul ifadeleri kullanma	
Öğrenme Çıktıları	Genel Öğrenme Çıktıları:	
	Kosullar	
	 Operatörler 	





	Algoritmik düşünceye odaklı özel öğrenme çıktıları: ● Öğrenci, farklı karakterlerin fiyatını belirlemek için değişkenleri kullanır.	
	 Oyuncu yiyecek satın aldığında bütçe değiştiği için öğrenci değişkenlerin değerini değiştirir. 	
	 Öğrenci, paranın kullanılabilirliğini kontrol etmek için "if" ifadesini kullanır. 	
	 Öğrenci, fiyatları ve bütçeyi karşılaştırmak için operatörleri kullanır. 	
	 Öğrenci, değişkenlerin değerini değiştirmek için operatörleri (çıkarma) kullanır. 	
	Kısa Açıklama: Kız pikniğe gidiyor ve yiyecek almak için yardıma	
	ihtiyacı var. 15 EURO var ve daha fazla harcayamıyor. Bir şey satın	
	aldığında bütçenin değeri değişir. Bütçesi çok düşükse, seçtiği yemeği	
	satın alamaz.	
	Görevler: Öğrenciler üç farklı karakter programlamak zorundadır. Bir	
	"kız", "yiyecek" (küçük değişikliklerle çoğaltabilecekleri) ve "bitir	
	düğmesi". Kız talimatlar verir. Oyuncunun ne kadar parası olduğunu	
	söyler ve sonunda (bitir düğmesine tıklayarak) oyuncunun kaç tane	
Amaç, Gorevier ve	sağlıklı ve sağlıksız ürün aldığını söyler. Yiyecek, fare işaretçisi üzerine	
Faaliyetlerin Kisa	geldiğinde fiyatını söyler. Oyuncunun yeterli parası varsa, bir ürün	
Tanitimi	satın alabilir ve bütçenin değeri değişir. Aksi takdirde yiyecek satın	
	alınamaz.	
	Amaç: Öğrenciler değişkenlerle nasıl çalışılacağını öğrenecekler.	
	Farklı başlangıç değerleri belirleme, değişkenlerin değerini	
	karşılaştırmak için koşullu ifadeler kullanma, değişkenlerin değerini	
	değiştirme, sağlıklı yiyecekleri saymak (olmayan) için değişkenler	
	kullanma. Ek olarak, metin eklemeyi, metinleri ve "if" ifadesini	
	birleştirmeyi tekrarlayacaklar.	
Faaliyetin Süresi	45 dakika	
Öğrenme ve	Aktif ääranma ishirliksi ääranma nrohlam säzma ovun tamalli	
Öğretme Strateji ve	öğranma	
Metotları		





Öğretme Formları	Bireysel Çalışma / İkili Çalışma
	(Motivasyon-Giriş, Uygulama, Düşünme ve değerlendirme)
	Kız bir bakkaldan piknik için yiyecek alıyor. 15 EURO var. Fare işaretçisi
	üzerine geldiğinde yiyeceğin fiyatını görebilir ve seçili yiyeceği
	tıklayarak satın alabilir. Sadece yeterli parası olana kadar yiyecek satın
	alabilir. Bitir düğmesine tıklayarak, oyuncunun kaç tane sağlıklı ve
	sağlıksız ürün aldığını söyler.
Öğretme Özeti	Cake costs 8 EUR.
	[Adım 1]
	Bu faalivet, bir birevsel calısma veva ciftler halinde calısma olarak
	planlanmıştır. Öğretmen bazı ipucları verir. Bazı zor kısımları acıklar ve
	gerektiğinde yardımcı olur. Öğrenciler arka planı secer ve bir ana
	hareketli grafik ekler. Örneğin bir kız. Kız başta bazı talimatlar verir.
	Örneğin:
	say Hellol for (2) secs say Ishavera-picnic-today, help-me-to-buy-some-food for (4) secs
	[Adım 2]
	Bu oyunda birkaç değişkene ihtiyacımız olacak.
	 Kullanılabilir para miktarını ayarlamak için bütçe.
	 Oyuncunun kaç sağlıklı öğe satın aldığını saymak için healthy_food





EUR. 10 for (2) secs

(sağlıklı_yiyecek). • Oyuncunun kaç sağlıksız öğe satın aldığını saymak için unhealthy_food (sağlıksız_yiyecek). • Her yiyecek ve her yiyeceğin fiyatını belirlemek için bir değişken (örneğin, karpuz fiyatı). Başlangıçta bütçe değişkeni örneğin 15 EURO olarak ayarlanabilir. Diğer iki değişken 0 olarak ayarlanmıştır. Bu kod [Adım 1]'den kız kodunun önüne eklenebilir. hen **Clicked** budget = to 15 ithy_lood = to D unhealthy_food = to [Adım 3] Öğrenciler bir karakter (yiyecek) ekler ve kostümünü seçerler. Gıdanın (karpuz) kodu üç kontrol olayına ihtiyaç duyar: a) Yeşil bayrak tıklandığında: Gıdanın fiyatını göstermek ve ayarlamak için. Değişkenin fiyatının makul bir şekilde belirlenmesine izin verin (tabii ki 0 değil ve 1'den büyük). b) Fareyle girildiğinde: oyuncuya ürünün maliyetinin ne kadar olduğunu söylemek için. Öğrenciler Görünümler-Düşünce (Looks – thinking) bloklarını kullanabilirler. - birleştirme metni - değişken değeri - metin kullanımıyla düşünme bloğunu kullanabilir. Örneğin: when 🔁 clicked show watermelon_price - to 4 when I am mouse-entered

think join Watermelon-costs-





c) Tik 1) (ala 2) 3) . 4)	klandığında: burada küçük bir yansıma yapmaları gerekir. Oyuncu ürünü hangi durumda satın alabilir ve hangisinde ımaz? Yiyecekleri satın alırsa bütçeye ne olur? Satın alınan ürünleri nasıl sayarız? Raftaki yiyeceklere ne olur?
1)	Oyuncu yeterli parası varsa ürünü satın alabilir. Bu nedenle
	öğrencilerin iki değişkeni karşılaştırması gerekir. Bunlar
	bütçe ve karpuz_fiyatıdır. Karpuz, sahip olduğundan daha
	pahalıysa satın alamaz. Öğrenciler oyuncuya bu ürünü satın
	alamayacağını söylemek için metin ekleyebilirler.
	If watermelon_price > budget say Yourdon't have enough money for (5) secs else
2)	Oyuncu 15 EURO'ya sahipse ve 4 EURO karşılığında bir
	karpuz satın alırsa, şimdi 15 - 4 = 11 EURO'dur. Yani bütçe
	değeri artık: 11'dir.
3)	Önceki Büte Değeri – Karpuz Fiyatı (previous <i>budget value</i> –
	watermelon_price)
Öğren	ciler bazı metinleri de ekleyebilir.
	say Greatchoicel for (2) secs set budget to budget — watermelon_price)
4)	Satın alınan ürün adedinin sayılması sağlıklı gıda (<i>healthy_food</i>) değişkeni 1 değiştirilerek gerçekleştirilecektir.
5)	Yiyecek tıklandığında gizlenir.
Mümkün o	olan çözümlerden biri;





when 1 am clicked If watermelon_price > budget say Yourdonthave enough money. for 6 secs else say Great choice! for 2 secs set budge! to budget watermelon_price change healthy_food by 1
[Adım 4]
Raflarda daha fazla yiyecek bulundurmak için öğrenciler karpuz
karakterini kopyalayabilir. Diyelim ki, ikinci yemek pasta olacak. [Adım
3] kodunda bazı değişikliklerin yapılması gerekiyor.
Öğrencilerin yapması gerekenler:
 Kostümü değiştirmek. Yeni değişken oluşturmak kek_fiyatı (<i>cake_price</i>). Kek_fiyatı'na bir değer belirlemek. Bloklarda yer alan tüm (karpuz_fiyatı) <i>watermelon_price</i> kodunu kek_fiyatı (<i>cake_price</i>) ile değiştirmek. Kek satın almadaki geri dönüşleri değiştirmek. <i>Change healthy_food by 1</i> komutunu <i>change unhealthy_food by 1 ile değiştirmek</i>.
Örneğin, Kek için tıklandığında(<i>when clicked</i> code) kodu:
when I am clicked = If cake_price > budget say Yourdon't have enough money! for 6 secs else say Toomuch sugar! for 2 secs set budget = to budget = cake_price change unhealthy_food = by 1 hide
[Adım 5]
Oyuncu satın almayı bitirdiğinde, bitir (finish) düğmesine tıklar.
Programa, oyuncunun düğmeye tıkladığını (yiyecek satın almakla





bittiğini) söylemek için bir mesaj yayınlıyoruz.
when I am clicked - broadcast finish
[Adım 6]
Sonunda kız karakterine dönüyoruz. Oyuncu alışverişini bitirdiğinde
kızın kendisine kaç tane sağlıklı ve sağlıksız ürün aldığını söylemesini
istiyoruz. Oyuncu bitir düğmesine tıkladığında, bir bitiş mesajı
gönderilir. Kız, mesajın bittiğini aldığında, "X sağlıklı ürün ve Y sağlıksız
ürün seçtiniz" gibi mesaj gönderilir.
when I receive finish say join Yourchose healthy food thealthy products and unhealthy food unhealthy products. () 5 secs
[Adım 7]
Oyuncu, oyun sırasında herhangi bir zamanda kızın bütçesini
bilgisayarın faresini kullanarak kontrol edebilir. Örneğin, şöyle bir şey
söyleyebilir/düşünebilir:
when I am mouse-entered wi say join Youthave budget EUE *** for ② secs
[Nihai Kod]
Kız





<pre>when is clicked set budgel = to 15 set healthy_food = to 0 set unhealthy_food = to 0 set unhealthy_food = to 0 set unhealthy_food = to 0 set unhealthy_food = to 0 set index = picnic today; heip-me-to-buy-some-food for () secs when I receive index = set index = ind</pre>
say join You have budget EUR () for 2 secs
when clicked show set cake_price to a
when I am mouse-entered withink join Cake-costs cake_price EUR. () for (2) secs
<pre>when 1 am clicked = it cake_price > budget say You don't have enough money. for 5 secs else say Too much sugar! for 2 secs set budget = to budget = cake_price change unhealthy_food = by 1 hide</pre>
Bitiş Düğmesi when I am cicked w





	[İlave Görevler]
	Öğrenciler isteklerine göre ek görevler ekleyebilir veya aşağıdaki görevleri takip edebilirler:
	 Oyunu her yiyeceği 3 kez satın alabilmek için değiştirin.
	 Başlangıçta oyuncuya daha fazla para verin.
	 Sonunda kız ayrıca kaç ürün satın aldığınızı da söyler. Örneğin, "2x
	karpuz, 1x üzüm, 2x patates kızartması aldınız".
	 Snap!'teki tüm aktivite: https://snap.berkeley.edu/project?user=mateia&project=Buvi
	ng%20food%20for%20a%20nicnic
Öğretmenler İcin	 Snap!'te cözümleri ile birlikte ek görevler:
Araçlar ve Kaynaklar	https://snap.berkeley.edu/project?user=mateja&project=Buyi
	ng%20food%20for%20a%20picnic%20%2B%20Add.%20Task
	 Lajovic, S. (2011). Scratch. Nauči se programirati in postani računalniški maček. Ljubljana: Pasadena. Vorderman, C. (2017). Računalniško programiranje za otroke. Ljubljana: MK.
Öğrenciler İçin	
Kaynaklar ve	Öğrenci için talimatlar (C4G2_InstructionsForStudent.docx)
Materyaller	





Öğrenme Senaryosu 17 – Operasyonlar/İşlemler

Öğrenme Senaryosu	Operasyonlar/işlemler
Adı	
	Puanları saymak ve sahne ile karakter kostümünü seçmek için değişkenler
	kullanma
Goomic	Karakter için sahne dekoru ve kostüm seçmek için rastgele sayı kullanma
Drogramlama	Tekrar döngüsünü kullanma
Programiana	Koşul ifadeleri kullanma
Deneyimi	Karşılaştırma için işlemleri kullanma
	Diyalog için algılamayı kullanma (sorun ve bekleyin/ askand wait)
	Yayın (broadcast) etkinliklerini kullanma
	Genel Öğrenme Çıktıları:
	 Değişkenler
	● Koşullu ifadeler
Öğrenme Çıktıları	● Döngü
	 Algılama blokları (Sensing blocks) Yayın etkinlikleri
	Algoritmik düşünceye odaklı özel öğrenme çıktıları: ● Öğrenciler, puan sayımı için değişkenleri kullanır.
	 Öğrenciler, puan sayımı için değişkenleri başlatır.
	 Öğrenciler koşullu ve mantıksal işlemleri kullanır.
	 Öğrenciler, karakterlerin grafiğini değiştirmek ve nihai sonucu hesaplamak için yayın olayını kullanır.
	Kısa Açıklama:
	Bir oyun oynarken oyuncunun Snap!'deki aritmetik işlemlerde ustalaşıp
Amac Görevlerve	ustalaşmadığını kontrol edelim. Kurallar aşağıdaki gibidir:
Faaliyatlarin Kusa	İlk işlenen sayısı 6 olan bir aritmetik işlem rastgele seçilir. İkinci işlenen
Faaliyetlerin Kısa Tanıtımı	1'den 3'e kadar bir sayı olacak şekilde rastgele seçilir. Oyuncu doğru cevabı
	girmelidir. Doğru ve yanlış cevaplar sayılır. Oyunun sonunda doğru sonuç
	bildirilir.
	Görev: Öğrenciler sahneyi / sahne dekorunu / ve karakterin kostümünü





	tanımlamalıdır. Gerekli değişkenleri planlamaları, hangi bloklara ihtiyaç
	duyduklarını belirlemeleri, sonunda sahne ve hareketli grafik için kodlar
	oluşturmaları gerekir.
	Ek görevler şunlar olabilir:
	 Sonuca bağlı olarak karaktere aşağıdakiler söyletilebilir:
	"Sizin için iyi! / "Good for you!" " veya "Snap'deki aritmetik
	işlemleri iyi bilmiyorsunuz! (You don't know well the arithmetic
	operations in Snap)
	Amaç: Öğrenciler değişkenler, rastgele sayılar, döngüler, yayın hakkında
	önceden edindikleri bilgileri geliştirecekler.
Faaliyetin Süresi	45 dakika
Öğrenme ve	Aktif Öğranma (tartışma, bir önçaki oyundan öğranilanlari danayimlama)
Öğretme Strateji ve	Aktir Ogrenime (tartişina, bir öncekir öyünüdir ögrenmemen deneyinmeme),
Metotları	oyun temelli ogrenine, problem çozme
Öğretme Formları	
	Bireysel çalışma / İkili çalışma/ Tüm sınıf ile ön çalışma
	(Motivasyon-Giriş, Uygulama, Düşünme ve Değerlendirme)
	Ogretmen, Snap'deki aritmetik işlemlerde ustalaşılıp ustalaşılmadığını ve projede gösterilip gösterilmediğini belirlemek için bir oyunun ihtiyaçına
	ilişkin problemi ortaya koyar.
	https://snap.berkeley.edu/project?user=ddureva&project=operations3
Öğretme Özeti	correct wanted water
	6 + 1
	input answert





1. Öğretmen, görevin koşulunun nasıl formüle edileceğini tartışır. Görev formüle edilmiştir. Rastgele bir şekilde on kez birinci sayının 6 olduğu bir aritmetik işlem seçilir. Ayrıca 1'den 3'e kadar sayılardan rasgele bir şekilde ikinci sayı seçilir. Oyuncu doğru cevabı girmelidir. Doğru ve yanlış cevaplar sayılır. Sonuç, oyunun sonunda bildirilir. 2. Değişkenler, değişkenlerin tanımlanması, başlatılması ve değiştirilme şekilleri öğrenciler ile beraber yorumlanır. 3. Rastgele sayı komutları, aritmetik ve mantık işlemleri, olay (event) yayın komutları öğrenciler ile revize edilir. 4. Temel kodun sahne mi yoksa karakter mi olduğu tartışılır. Örnekte, ana kod sahne içindir ve hareketli karakterin kodunda kostümü değiştirmek ve nihai sonucu hesaplamak için komut dosyaları vardır. operations3 1 number2 rivegashie





5. Operasyon/işlemi seçmek için aşağıdaki komutlar kullanılır:







set operation - to pick random (1) to (4) switch to costume operation

Karakter için kostüm seçimi, Sayı karakterine yayın kullanılarak (broadcast) yapılır. Seçilen kostüm numarası, projedeki tüm nesneler için tanımlanan Kostüm Numarası (Costume Number) değişkeninde saklanır ve bu nedenle sahne kodunda kullanılabilir.

Sahne / sahne dekoru / ve karakter kostümü rastgele seçildikten sonra, oyuncuya aşağıdaki komutla işlem için doğru yanıtı girmesi için bir soru sorulur.

ask Inputanswer! and wait

Girilen yanıt, seçilen işlemlerin sonucu ile karşılaştırılır.

Aşağıdaki komut kullanılır:

Eğer (Koşul) if (conditional)

başka6 / (else6 /)

"-" işlemi seçilirse, 6 - "Karakterin kostüm numarası"(6 - "Sprite's costume number) sonucunun cevapla eşleşip eşleşmediği kontrol edilir. Eğer eşleşirlerse, doğru değişken artar, aksi takdirde yanlış cevapların sayısı için değişken artar.



Komutların geri kalanı için komut dosyası (script) benzerdir. Ancak seçilen işlemler farklıdır.

İşlemlerin geri kalanı için tekrarlanan kod sıralamasından kaçınmak için

öğrencilere kodun bir bölümünü nasıl kopyalayacakları ve VIIII'da yer

alan aritmetik işlemi nasıl değiştirecekleri öğretilebilir.





Kod kopyalama: 1. Farenin sağ tuşu ile komut dosyasına tıklayın. 2. Çoğalt (duplicate)'ı tıklayın. name - of Stage - = help... make a copy relabel. and pick it up answer duplicate delete ect = by 🚹 script pic. ringify change wrong - by (1) 3. Kopyalanan komut dosyasını ilgili konuma yerleştirmek için fareyi kullanın. Öğretmenin takdirine bağlı olarak, öğrenciler kodun bir kısmını kendilerinin nasıl kopyalayacaklarını bulmakla görevlendirilebilir. İşlemleri değiştirmek. 1. İşlem işaretini farenin sağ tuşuyla tıklayın. Context (Bağlam) menüsü görünecektir. costume name - of Stage -if 6 answei help. relabel. 1 duplicate else delete script pic 1 ringify 2. Yeniden etikelendir (relabel)'i seçin, işlemler (operations) görünecektir. costume name * of Stage * -= answer if 6 el 3. İşlemleri seçin. Not: Öğrencilerin yaşı ve aritmetik işlem bilgisi izin veriyorsa, görev işlemler,

derecelendirme (^) ve modüle göre bölme (mod) ile genişletilebilir.





	6. Öğrenciler, karakter için kendi sahnelerini / sahne dekorlarını / ve
	kostümlerini oluşturan takımlar halinde çalışırlar. Zaman kısıtlamaları
	varsa, sahne ve hareketli grafiği içeren "yarı destekli" bir proje
	kullanılabilir.
	Snap!'teki tüm aktivite
Öğretmenler İçin	https://snap.berkeley.edu/project?user=ddureva&project=operations3
Araçlar ve	Scratch'teki tüm aktivite
Kaynaklar	 Дурева Д., М. Касева, Г. Тупаров, Компютърно моделиране, 4. клас, Просвета, 2018, София (Dureva, D., М. Kaseva, G. Tuparov, Kompyutarno modelirane, 4. klas, Prosveta, 2019, Sofia)
Öğrenciler İcin	
5 3	 Snap!'teki yarı bitmiş aktivite
Kaynaklar ve	
-	https://snap.berkeley.edu/project?user=ddureva&project=operations half
Materyaller	
-	 Öğrenci için talimatlar (C4G2_InstructionsForStudent.docx)





Öğrenme Senaryosu	Geri dönüsüm
Adı	Seri donuşum
	Bir karakteri gösterme ve gizleme
	Noktaları saymak için değişkenleri kullanma
Geçmiş Programlama	Sonsuza kadar döngüsünü kullanma
Deneyimi	Koşul ifadeleri kullanma
	Karşılaştırma için işlemleri kullanma
	Renk algılama özelliğini kullanma
	Genel Öğrenme Çıktıları:
	 Değişkenler
	Koşullu ifadeler
	● Döngü
	Bir yöne dönme
	Algılama blokları
Öğrenme Çıktıları	Kodları yeniden düzenleme
	 Algoritmik düşünceye odaklı özel öğrenme çıktıları: Öğrenciler, oyunu bitirmek için bekleme ve mantıksal işlemlerini kullanır. Öğrenciler, sahneyi değiştirmek için bekle ve engelle komutunu kullanır. Öğrenciler, puanları saymak için değişkenleri kullanır. Öğrenciler, koşul ve mantıksal işlemleri kullanır. Öğrenciler, benzer karakterlerin kodlarını karşılaştırırlar. Öğrenciler, kodları yeniden düzenlerler. Öğrenciler, karakter konumlandırmayı kullanır (ek bir görevde rastgele konumlandırma kullanır).
	Kısa Tanım:
Amaç, Görevler ve Faaliyetlerin Kısa Tanıtımı	Birisi okulun önüne çöp attı. Oyuncudan, çöp toplamaya kâğıt ve camı geri dönüşüm için ayırarak yardım etmesi istenir. Çöp doğru konteynere yerleştirildiğinde çöp gizlenir. Çöp yanlış konteynere yerleştirilirse "Bu bir kağıt konteyneri değil" veya "Bu bir cam konteyneri değil" mesaiları belirir ve cöp orijinal konumuna geri
	Konteynen degi mesujian benni ve çop orijinar konullulla ger

Co-funded by the Erasmus+ Programme of the European Union





	döner. Tüm çöpler doğru konteynerlere konduğunda oyun sona erer.
	Görev: Öğrenciler, sahne ve karakterlerin kodlarını keşfetmeli. Atık kağıt ve atık cam şeklindeki karakterlerin kodlarını karşılaştırmalı. Yeni
	karakterler ile kodları eklemeli ve sahnede yeni karakterlere göre
	kodları değiştirmeli.
	Ek görevler şunlar olabilir:
	 Karakterlerin rastgele koordinat seçimi ile atık karakter pozisyonunu
	değiştirmek.
	 Sahne sayısını azaltmak ve robotu ayrı bir karakter olarak çıkartmak
	(Robot, sahne arka planının bir parçasıdır).
	Amaç: Öğrenciler önceden edindikleri bilgileri geliştirecekler ve oyun
	senaryosunu yeni nesneler, kodlar ve yeni karakterlere göre değişen
	kodlarla genişletecekler. Ayrıca öğrenciler kodları yeniden
	düzenleme konusunda eğitileceklerdir.
Faaliyetin Süresi	45 dakika
Öğrenme ve Öğretme	Aktif öğrenme (tartışma, bir önceki hazırlanan oyun ile deneyimleme),
Strateji ve Metotları	oyun temelli öğrenme, problem çözme
Öğretme Formları	
	Bireysel Çalışma / İkili Çalışma/ Tüm Sınıfla Ön Çalışma





Öğretme Özeti	(Motivasyon-Giriş, Uygulama, Düşünme ve Değerlendirme)
	 Öğretmen, çöpleri ayırma problemini ortaya koyar ve farklı çöp türleri için kutuların renkleri hakkında yorum yapar (kâğıt için mayi plastik için yoşil)
	 Öğrencileri oyunu oynamaya ve kelimelerle açıklamaya yönlendirir. Kaç sahne izliyorlar ve kaç tane karakter var? Oyun nasıl başlıyor? Hangi karakter oyuncunun adını sorar? Kaç değişken kullanılır ve bunlar nasıl adlandırılır? Kâğit, bir cam konteynerine konulduğunda ve kâğit konteynerine konulduğunda ne olur?
	Edit Download Embed Add to Collection Detete
	1. Çalışılan komutların güncellenmesi
	Kullanıcı ile diyaloğa girme komutları geri çağrılır. Değişen sahneler
	hakkında bir yorum yapılır. (Robot ile Sahne 1, okul ve çöp ile Sahne
	2 ve Robot ile Sahne 3 ve Bravo!) Yazısı.
	Dani, In front of the school there is some garbage made from paper and galss.
	Edit Download Embed Add to Collection Delete




Çöpün bir konteynere uygun şekilde yerleştirildiğinin kontrolünün, bir koşullu blok ve sensing grubundan dokunma koşullarına sahip bir blokla yapılması gerektiği tartışılmaktadır. Sözlü bir açıklama yapılır. Kâğıt konteynerine bir parça kâğıt çöp değerse çöp gizlenerek (doğru çöp kutusuna yerleştirilir) toplanan kâğıt atığının puanları 1 artar. Cam konteynerine bir parça kâğıt çöp değerse, konteyner "-" Bu bir kâğıt konteyneri değil" der. Aynı şey cam çöp için de geçerlidir. recycling anama



6. Sahne ve karakter kodlarının incelenmesi.

Problemi çözme olasılıkları tartışıldıktan sonra sahne ve karakterler için kodlar tartışılır.

Sahne kodu şu hususlara vurgu yapılarak yorumlanmıştır:

- İsim değişkeninin başlangıç değerinin ayarlanması ve kullanıcı ile diyalog içerisinde kullanılması.
- Sahne dekorunu (kostümleri) ve oyunu bitirme koşulunu değiştirmek.



Karakter kodlarına bakarken, bunların tek bir slaytta gösterilmesi veya basılı hurda kâğıt ve hurda cam parçalarının her birine iki kod verilmesi tavsiye edilir. Kodlardaki ortak ve farklı unsurlar arasında karşılaştırma yapılır.





10 m		paper4		M	paper1 V meganie	
5	icripta	Costumes Sounds		Scripts	Costames Sounds	
wh		clicked		when 🍋 i hide	dicken	
	-	I costume name of	State = school	wait until chow	Costume name of Stage - School	
go	lo xi	(103) v: (-86)		an to x: C	79) v: C141)	
H	tou	ching glass tin 7		I Could	ing gass bin 7	
l	00 TO 1	41 -103 VI -85	2 secs	go to st	- (79) v: (141)	
ii E	Note	ching paperbit 7		If touch	ning paperbin 7	
	dep :	e coperat by 1		change stop mi	paper of by	
B		Gass1		100 E	glass4 a megadie	
	ioriphi	Costumes Sounds		Seripte	Costumes Bounds	
		clicked		when Pa	Clickord	
	it and	Costume name - O	Stage - Robert	wait until	Costume name of State = actool	
go 1	TO X:	-82) v: -133)		go to at	6) y: (-162)	
H	tou	china poperten 7		if touc	hing papertin 2	
	ao to a	x: (-82) v: (-133)	2, 50005	go to x	6 v: 6162	
H E	tide:	ching glassein - 7		il touc	hing dass bin + 7	
	chang stop 1	e crass in by 🔳		channe stop	t stans - by 🚹	
01						
7	'. (:	Oyunu iki ye sin hir görov	ni karakter (k	âğıt çö	p ve cam çöp) ile tamam tarlara bir kad atama ya	lamak
	i i	le çöp kutusı	u kodlarını de	u karak ğiştiriln	nesi.	Same
IKI	yer	ni karakterin i	nasii oluşturu	iacagi t	artışılır. Seçenekler:	
	1-	Mevcut ola	nları çoğaltın	ve Snap	o!'de dûzenleyîn.	
	2-	Bir grafik o	düzenleyicide	yenile	rini oluşturun veya inte	rnette
		ücretsiz da	ğıtılan görünt	üleri ara	ayıp oyuna aktarın.	
Ayr	rıca	oyunun tar	mamlanmasıy	'la ilgili	sahne kodundaki değişi	iklikler
hał	kkır	nda yorum ya	ıpmak gerekir	·.		
De	ğişk	kenlerin başl	angıç değerle	erinin i	ki konteynerin kodunda	değil,
sah	nne	kodunda ay	varlanması ve	buna	göre bir ayarlamanın mü	ümkün
olu	ро	ılmadığı da ta	artışılmalıdır.			





	Öğretmenin takdirine bağlı olarak görevin durumu karmaşık hale
	getirilebilir.
	 Oyun başlarken çöpler uygun bir yere bırakılmalıdır. Burada çöpün dağıtılabileceği koordinatların gerçek görünmesi için sınırlı bir alanın seçilmesi gerektiğini not etmeliyiz. Örneğin, çöplerin alanı kırmızı dikdörtgenin koordinatları ile sınırlanmıştır.
	 Veni bir Bobot karakteri tanıtın ve sabnedeki öğelerin savışını
	 Yeni bir Robot karakteri tanttiri ve sannedeki ögelerin sayısını azaltın. Mavi bir konteyner karakteri yerine oyuncu ile diyaloğa girmesi için uygun kodu Robota yazın.
Öğretmenler İçin	Snap!'teki tüm aktivite
Araçlar ve Kaynaklar	https://snap.berkeley.edu/project?user=ddureva&project=recycling
	Scratch'teki tüm aktivite
	 Дурева Д., М. Касева, Г. Тупаров, Компютърно моделиране, 4. клас, Просвета, 2018, София (Dureva, D., M. Kaseva, G. Tuparov, Kompyutarno modelirane, 4. klas, Prosveta, 2019, Sofia)
Öğrenciler İçin	 Snap!'teki yarım tamamlanmış aktivite
Kaynaklar ve	https://snap.berkeley.edu/project?user=ddureva&project=recycling
Materyaller	 Öğrenci için talimatlar (C4G2_InstructionsForStudent.docx)





Öğrenme	Piyano calma						
Senaryosu Adı	· · / ···· · · · · · · · · · · · · · ·						
	Noktaları saymak için değişkenleri kullanma.						
Comis	Düğmesine basıldığında olayını (event When I am pressed) kullanma.						
Geçmiş	Tekrar döngüsünü kullanma.						
Programlama	Koşul cümlelerini kullanma.						
Deneyimi	Sahne / sahne dekorunu değiştirmek ve hareketli karakterlerin						
	etkinliklerini yönetmek için yayın etkinliklerini kullanma.						
	Genel Öğrenme Çıktıları:						
	 Değişkenler; 						
	 Koşullu ifadeler; Döngü: 						
	 Yayın etkinlikleri; 						
	 Sesler; Müzik programlama; 						
	• Muzik programiama;						
Öğrenme	Algoritmik düşünceye odaklı özel öğrenme çıktıları:						
Çıktıları	 Öğrenciler, puan sayımı için değişkenleri kullanır. 						
	 Öğrenciler, puan sayımı için değişkenleri başlatır. 						
	 Öğrenciler, elde edilen puanları tahmin etmek için koşullu ifadeler kullanırlar. 						
	 Öğrenciler, sahne / sahne dekorunu değiştirmek ve karakterlerin hareketleri için yayın etkinliğini kullanır. 						
	 Öğrenciler, melodileri oluşturmak için Ses grubundaki blokları kullanırlar. 						
	 Öğrenciler, komut dosyalarındaki blok sayısını azaltmak için tekrar döngüsü belirler. 						
	 Öğrenciler, oyunun işlevselliğini genişletir. 						
	Kısa Açıklama:						
Amaç, Görevler	Kraliçe Mary'nin harika dünyasına girelim. Oyuncuyu müzik dinlemesi için						
ve Faaliyetlerin	sarayına davet ediyor. Balo salonunda küçük dinozor arkadaşı Dino						
Kısa Tanıtımı	piyano çalıyor. Oyunda Dino birkaç müzik tonu çalar ve oyuncular bunun						
	hangi ton/nota olduğunu anlamalıdır. Doğru tahmin ederlerse, doğru						

Co-funded by the Erasmus+ Programme of the European Union





	cevap için bir puan alırlar. Bilmiyorlarsa yanlış cevap için puanlar azaltılır.				
	Tonları tanımladıktan sonra, daha karmaşık bir görev belirlenir. Dino bir				
	melodi çalar ve oyuncunun hangi şarkı olduğunu tanıması gerekir. Düzgün				
	tanımlanmış bir melodi için oyuncu 5 puan alır.				
	Görev: Öğrenciler, sahne / sahne dekoru / ve karakter kostümlerine sahip				
	yarı destekli bir dosya kullanır. Gerekli değişkenleri planlamaları, ihtiyaç				
	duydukları blokları belirlemeleri, ses grubunun bloklarını ve notaları				
	çalmanın yolunu tanıyın. Birkaç melodiyi çalmak için komut dosyaları				
	oluşturun.				
	Amaç: Öğrenciler melodileri kodlama ve çalma hakkında bilgi edinecek				
	ve değişkenler, döngüler, koşullu, yayın ve diğer olaylar hakkında				
	önceden edindikleri bilgileri geliştireceklerdir.				
Faaliyetin	90 dakika				
Süresi	30 Ganina				
Öğrenme ve					
Öğretme	Aktif öğrenme (tartışma, bir önceki hazırlanan oyun ile deneyimleme),				
Strateji ve	oyun temelli öğrenme, problem çözme				
Metotları					
Öğretme	Birevsel Calısma / İkili Calısma/ Tüm Sınıfla Ön Calısma				
Formları					
	(Motivasyon-Giriş, Uygulama, Düşünme ve değerlendirme)				
	1. Öğretmen, oyunu oluşturma görevini belirler. Görevin				
	tamamlanabileceği araçlar tartışılır. Bir melodiyi programlamak				
Öğretme Özeti	için mevcut kod yazma kaynaklarından şu anda haberdar				
	olmadıkları sonucuna varılmıştır.				
	2. Öğretmen bir melodi programlayarak oyunun bir bölümünü				
	gösterir.				
	https://snap.berkeley.edu/project?user=ddureva&project=Play_a_Piano_1				





			Play	_a_P	iano_	1 by define	10					
							White	h is the nic C or F?				
3	. Öğr	retme	n ko	odu	göste	rerek	ses	gru	ıbu	komut	larının	nasıl
Sna	kull p!'tek	lanılal i kit	oilece aplıkt	ğini a aki s	çıklar. sesler	ve	bilgi	sayar	daki	seslei	r ile	çeşitli
ens	trüma	anlard	an çık	an se	sler k	ullanı	labilir					
Aracı	(tool)) seçn	nek içi	in kon	nutlar	ı kulla	nın (o	comn	nand):		
set	instrume	nt to 1	7									
			(1) sine (2) squa (3) sawt (4) triang	re poth gle								
No	t: Sc	ratch	'da g	ok o	daha	fazla	sı m	evcu	ttur.	Öğrer	nciler	birçok
e	enstrü	imani	n sesi	ni der	neyebi	ilirler.						
4	. Öğr	retme	n mü:	zik no	talarıı	nın na	sıl ay	arlan	acağı	nı girer	:	
"	olay not	e 60 •	for 0.5) beats	" (N	lotayı.	san	iye ç	al) k	omutu	kullar	ulabilir.
Βι	ırada	ilk nu	umara	nota	ıyı aya	arlar	ve iki	nci n	umar	a nota	nın ne	e kadar
sü	reyle	çalına	icağın	ı belir	tir.							
İlk	sayır	nın ya	nında	ıki ok	a tıkla	adığın	ızda,	bir p	oiyan	o klavy	esi be	lirir ve
or	idan b	oir not	ta seç	ilebili	r. Bu p	biyanc	klavy	/esi il	ki okt	ava yay	yılır.	
	lay note	60 • fo	05 b	oats								
) III		Ţ										
					-	I					-	
С	C#	D	Eb	E	F	F#	G	G#	A	Bb	В	C
60	61	62	63	64	65	66	67	68	69	70	71	72
.,			_	. I . V		1						
He	er bir	notan	in uzi	uniugi	u raka	amlar	iie be	irtili	r. 1-	tam no	., 05 .,	- yarım
nc	ota, O,	,25 çe	eyrek	nota.	(Ond	ialik s	ayılar	i bilr	neye	n oğrer	ncilere	kesirli





olarak gösterilebilir : ½, ¼, 1/8 gibi)

play note 59 for (1) / (4) beats play note 60 for (1) / (2) beats

Öğretmenin takdirine bağlı olarak öğrenciler komutları kendi başlarına deneyebilir veya destek talep edebilir.

5. Jingle Bells (yılbaşı şarkısı) melodi kodları, müzik notaları kullanılarak da tartışılır.



- 6. Görev, koddaki tekrarlanan satırların sayısını azaltmak olarak belirlenmiştir. Kullanılacak komut (*döngü tekrarlama / repeat loop*) tartışılmıştır. Öğrenciler, dersin başında belirlenen oyunu oluşturmaları gereken takımlara ayrılırlar. Her takım oyun senaryosunu tartışır ve açıklama sayfasında oyun planını açıklar (Ekli SNAP_Program_Design_ve_Planning Çalışma Sayfası.docx). Sahnelerdeki ve karakterlerdeki eylemlerin detayları için açıklamaya tablolar eklenebilir. Dinozorun oynarken dans etmesi için bir koşul (condition) eklenebilir (Önceden hazırlanmış dosyada dinozor birkaç kostüme sahiptir).
- 7. Öğretmen dosyadan senaryoların bazı bölümlerini görüntüleyebilir.
 https://snap.berkeley.edu/project?user=ddureva&project=PlayAP
 iano





Öğretmenler	Snap!'teki tüm aktivite
İçin Araçlar ve	https://snap.berkeley.edu/project?user=ddureva&project=Play_a_Piano
Kaynaklar	<u>_1</u>
	https://snap.berkeley.edu/project?user=ddureva&project=PlayAPiano
Öğrenciler İçin	 Snap!'teki yarı tamamlanmış aktivite
Kaynaklar ve	https://snap.berkeley.edu/project?user=ddureva&project=Play a Piano
Materyaller	Half backed
	 Öğrenci için talimatlar (C4G2_InstructionsForStudent.docx)

Öğrenme Senaryosu 19.2 - Piyano çalma

Öğrenme Senaryosu Adı	Piyano çalma				
Geçmiş	Döngü tekrarını kullanma				
Programlama	Değişkenleri kullanma				
Deneyimi	Koşul ifadeleri kullanma				
	Genel Öğrenme Çıktıları:				
	KoşullarDöngüler				
Öğrenme					
Çıktıları	Algoritmik düşünceye odaklı özel öğrenme çıktıları:				
	 Öğrenci, müzik çalmak için döngü tekrarını kullanır. Öğrenci, karakterlerin girdiye tepki vermesini sağlamak için kod kullanır. Öğrenci, karaktere sesler ekler. 				
	 Öğrenci, bir karakterin kostümünü değiştirmek için kod kullanır. 				
	Kısa açıklama: Öğrenci, verilen notlara göre piyanoda şarkı çalmak				
	zorundadır.				
Amaç,	Görevler: Öğrenciler piyano tuşlarını programlamalıdır (her tuşun belirli				
Görevler ve	bir tonu çalması gerekir). Sahnede, biri notaları görüntülemek ve diğeri				
Faaliyetlerin Kisa Tanitimi	melodiyi çalmak için iki farklı düğme gösterilmelidir.				
	Amaç: Öğrencilerin bir karaktere tıklayarak müzik çalmayı ve kostümü değiştirmeyi öğrenmeleridir.				





Faaliyetin Süresi	45 dakika
Öğrenme ve Öğretme Strateji ve Metotları	Bireysel öğrenme, oyun temelli öğrenme, problem çözme
Öğretme Formları	Aktif Çalışma / İkili Çalışma
	(Motivasyon-Giriş, Uygulama, Düşünme ve Değerlendirme)
	Başlangıçta sahnede bir piyano gösterilir. Piyanonun yanında iki düğme
	olmalıdır. İlk düğmeye tıklandığında şarkının notaları ve sözcükleri
	görüntülenmeli ve ikinci düğmeye tıklandığında tekrarlanması gereken
	melodi çalınmalıdır. Ek olarak, piyanonun yanında projeyi yeniden
	başlatacak "X" düğmesi bulunmalıdır.
	[Adım 1]
	Piyano çalma (Play a piano) programını açın. Program, bu görev için gereken tüm arka planları ve karakterleri içerir.
Öğretme Özeti	Arka plan ile C tuşu ve bir siyah anahtar için karakter (sprite) verilir.
	Öğrencilerin C anahtarını kopyalaması, doğru konuma getirmesi ve
	yeniden adlandırması gerekir. Tuşlar şu sırada olmalıdır: C, D, E, F, G, A,
	B. Klavye aşağıdaki resimdeki gibi görünmeli ve tuşların altında yazılı
	tonları yeniden üretmelidir.
	C D E F G A B
	60 62 64 65 67 69 71





5 siyah anahtar almak için "black_key (siyah anahtar)" karakterini 4 kez çoğaltın ve bunları siyah anahtar 2'den siyah anahtar 5'e kadar adlandırın. D ve E, F ve G, G ve A ve A ve B tuşlarının arasına yeni siyah anahtarlar yerleştirin.

Eğer siyah anahtar beyaz anahtarın arkasında gizli isi şu kodu kullanın:

go to front a layer

B tuşu için de aynısını yaparak bunları klavyenin sonuna yerleştirin. "Sürüklenebilir/draggable" düğmesinin işaretini kaldırın. Böylece tuş karakter oynatma sırasında taşınamaz.



[Adım 2]

Karakter tuşlarına basarak sesleri çalmayı etkinleştirin. "C" anahtarı için "*Tıklandığında (when I am clicked)*" şapka bloğunu ekleyin ve "C" mesajını yayınlamasına izin verin.



Bir tuşa basıldığında ses üretmek için, "*C mesajı aldığımda/When I receive c*" şapka bloğu ekleyin ve ardından 0.5 vuruş için 60 çalma notası ekleyin.



Hangi tuşa basıldığını vurgulamak için o karakterin kostümü geçici olarak değiştirilmelidir. C1 kostümünü C karakterine aktarın. "Tıklandığımda (When I am clicked)" bloğunda, kostümü 0,2 saniyeliğine C1 olarak





değiştirin. Sonra C kostümüne dönün. d 🛥 0.2 ch to costume a [Adım 3] İkinci adımı tüm beyaz anahtarlar için tekrarlayın. [Adım 4] Klavyeyi kullanarak piyano çalmak için c tuş karakterine "c tuşuna basıldığında - When c key pressed" bloğu ekleyin ve "Tıklandığında -When I am clicked" bloğundan kodun geri kalanını kopyalayın. costume a1 0.2 vitch to costume a Klavye üzerindeki c tuşu basılı tutulursa tuşa basıldığı sürece sesin tekrarlanacağına dikkat edin. Bunun nedeni "a" mesajının tekrar tekrar yayınlanmasıdır. Bir mesajı yayınlamayı durdurmak için kodun sonuna Kontrol paletinden bir "... e kadar bekle - wait until" bloğu ekleyin. wait until 🔵 Bir mesajı yayınlamayı bitirmek için, "değil" "not" operatörünü kullanın ve "a düğmesine basıldı- key a pressed" bloğuna ekleyin. a1 me a – til (not (key a – pre Beyaz anahtarlar için de aynısını yapın.





[Adım 5]

Yeni bir karakter oluşturun ve bir keman anahtarının resmini kostüm olarak içe aktarın. Bu, çalınacak kelimeleri ve notaları görüntülemek için bir düğme olacaktır.



Notaları görüntülemek için düğmeye tıklandığında "*akorlar- chords*" mesajının yayınını etkinleştirin.



Sahne için yeni bir "akorları- chords" kostümünü içeri aktarın.



Sahnenin bir "akor- chords" mesajı aldığında kostümü "akorlara- chords" çevirmesini sağlayan bir kod ekleyin.



[Adım 6]

Kostüm olarak nota şeklinde bir karakter bulun. Bu, tekrarlanması gereken şarkıyı çalmak için bir düğme olacaktır.







Kod, şarkının ilk iki dizesi için yazılmış olup diğer dizelerin kodunu yazmanız gerekmektedir. Bu kod müziklerde görünen şarkının aynısıdır.











mesajının yayınını etkinleştirin.
when clicked when I am clicked set size to 50 % broadcast blank
"Boş-blank" mesajını aldıktan sonra kostümü "boş-blank" olarak
değiştirmek için sahneye bir "Aldığımda - When I receive" şapka bloğu
ekleyin.
when I receive chords when I receive blank when I r
[Ek görevler]
Öğrenciler isteklerine göre ek görevler ekleyebilir veya aşağıdaki görevleri takip edebilirler.
 Nota karakterini çoğaltın (ve arka planda konumunu değiştirin) ve başka bir şarkı için bir program yazın.
 Yeni şarkı için akorları olan bir arka plan ekleyin.
[Nihai Kod]
Anahtar
when I am clicked broadcast a switch to costume a wait 0.2 secs switch to costume a when I receive a play note 60° for 0.5 beats
Keman anahtari
when I am clicked broadcast chords
Nota











	Sahne when I receive chords switch to costume chords switch to costume blank
Öğretmenler İçin Araçlar ve Kaynaklar	Snap! – Piyano çalma projesi: <u>https://snap.berkeley.edu/project?user=ifrankovic&project=Play%20a%</u> <u>20Piano</u>
Öğrenciler İçin Kaynaklar ve Materyaller	Snap!'teki yarı tamamlanmış aktivite <u>https://snap.berkeley.edu/project?user=ifrankovic&project=Play%20Pia</u> <u>no</u> (27.1.2020) Resimler: • Karakter resimleri: • a.png, a1.png • b.png, b1.png • violin_key.png • Arka planlar: notes.png





Öğrenme Senaryosu 20 - Test

Öğrenme	Tost					
Senaryosu Adı						
	Karakter gösterme ve gizleme					
	Noktaları saymak için değişkenleri kullanma					
Geçmiş	Döngüyü sonsuza kadar kullanma					
Programlama	Koşul ifadeleri kullanma					
Deneyimi	Karşılaştırma için işlemleri (operator) kullanma					
	Renk algılama özelliğini kullanma					
	Sahneyi değiştirme					
	Genel Öğrenme Çıktıları:					
	• Değişkenler					
	 Koşullu ifadeler Döngü 					
	Algılama blokları					
Öğrenme Çıktıları						
	Algoritmik düşünceye odaklı özel öğrenme çıktıları:					
	• Öğrenciler, cevabı tahmin etmek için koşullu ifadeler kullanır -					
	 Dogru veya Yanlış. Öğrenciler, sahnenin kostüm değişikliği için bloklar kullanır. 					
	 Öğrenciler, puan sayımı için değişkenleri kullanır, 					
	 Ogrenciler, mantıksal işlemleri kullanır. Öğrenciler, asamaların karmasık arka planlarını hazırlamak için 					
	harici grafik düzenleyici kullanır.					
	Kısa Açıklama:					
	Öğretmeninizin Snap'de kullanılan komutları test etmek için Quest					
	Tabanlı Oyun oluşturarak Snap! Bilginizi ölçmesine yardım edin.					
Amaç, Görevler	Görev: Öğrenciler örnek oyunu keşfetmeli, "yarı destekli" oyundan					
ve Faaliyetlerin	seçim yapmalı, soruları belirleyecek kendi karakterlerini bulmalı veya					
Kısa Tanıtımı	tasarlamalı, "yarı destekli" oyundan seçim yapmalı veya ilk aşama arka					
	planını ve uygun sorularla sahne arka planlarını tasarlamalıdır.					
	Amaç: Öğrenciler önceden edindikleri bilgileri geliştirecekler ve oyun					
	senaryosunu yeni aşamalara göre yeni arka plan, kod ve değişen					





	kodlarla genişletecekler.					
Faaliyetin Süresi	90 dakika					
Öğrenme ve	Aktif öğrenme (tartısma, bir önceki ovundan öğrenilenleri					
Öğretme Strateji	denevimleme), ovun temelli öğrenme, problem cözme,					
ve Metotları						
Öğretme						
Formları	Bireysel çalışma / İkili çalışma/ Tüm sınıfla ön çalışma					
	(Motivasyon-Giriş, Uygulama, Düşünme ve değerlendirme)					
	1. Öğretmen, programlama bilgisini test etmek için bir oyun testi					
	oluşturma ihtiyacını ortaya kovar.					
	2. Öğrencileri oyunu oynamaya ve kelimelerle açıklamaya yönlendirir.					
	Kaç sahne dekorasyonu ve karakter gözlemliyorlar? Oyun nasıl başlıyor?					
	Kaç değişken kullanılır? Nasıl adlandırılır? Ne için kullanılır? Cevap doğru					
	/ yanlış olduğunda ne olur? Testte sorular nasıl sunulur?					
	3. Soru sormak ve cevaplamak için kullanılan algoritma hakkında yorum					
	yapın /					
	 Bir sahne kostümüne geçme (soruyu içerir). 					
	 Abby'ye soru sorması için bir kostüm atamak. 					
Ogretine Ozeti	 Abby der ki! - Evet veya Hayır olarak yanıtlayın. 					
	 Oyuncu bir cevap girer - Evet veya Hayır. 					
	• Cevap doğruysa, Abby "Doğru" der ve doğru cevapların sayısı					
	artar. Aksi takdirde Abby "Yanılıyorsun" der ve yanlış cevapların					
	sayısı artar.					
	4. Tüm soruları yanıtladıktan sonra ne olacağı hakkında yorum yapın.					
	 Sahnede kostüm / arka plan değişikliği. 					
	• Abbey, doğru ve yanlış cevapların sayısını gösterir ve bir tahmin					
	verir.					
	5. Oyundaki kodları test etme					
	"Kullanıcı ile diyaloğa grime", "sahne dekorunu ve karakter kostümünü					





değiştirme komutları" ve "koşullu komutlar" yorumlanır. Her karakterin
kodları incelenir. Bir değişken oluşturma üzerine yorum yapılır.
stant2 series Contames Sounds when Estimated Sounds when Estimated Contames Sounds when Estimated Contames Sounds when Estimated Contames Sounds
Broadmast State
Abby Stription
an Abby daggable
Rempt Constant when L rectaive: Stati- save Vicestant
Doğru cevabın EVET olduğu ve doğru cevabın HAYIR olduğu durumlar
yorumlanır.
Not verme kodu ve toplam değişkeninin neden kullanıldığı ayrıntılı





olarak tartışılır.



Bireysel sorular için sahne dekorunu tasarlamanın yolu tartışılır.

Çünkü Snap'te! Kostümlerde ve sahnelerde yazı yazmak mümkün değil. Harici bir grafik editörü kullanmak gerekmektedir. Diğer bir seçenek de soruyu oluşturmak ve karşılık gelen metin kutusunu grafik biçiminde dışa aktarmak için MS Powerpoint kullanmaktır.

Snap'teki kostüm ekleme özelliği gözden geçirilebilir.

1. Grubu 2 veya 3 öğrenciden oluşan takımlara ayırmak.

2. Test soruları için konunun yayınlanması. Örneğin - Değişkenleri Kullanma; Döngüler; Hareket, Algılama, Aritmetik ve Mantıksal İşlemler.

3. İlgili ekip tarafından bir konu üzerine sorular içeren sahnelerin tasarlanması. Gerekirse öğretmen öğrencilere soruların içeriği hakkında tavsiyelerde bulunur. Sorular tartışılır ve her ekip en az iki soru için bir sahne oluşturur.

4. Kodu oluşturma. Öğrencilerin kullanması için yarı tamamlanmış sahne ve karakter kostümlerinden oluşan bir dosya verilir. İsterlerse kendilerine ait bir dosya da oluşturabilirler. Çalışma, model testine benzetilerek yapılır.





	Snap!'teki tüm aktivite
Öğretmenler İçin	https://snap.berkeley.edu/project?user=ddureva&project=test2
Araçlar ve	Scratch'teki tüm aktivite
Kaynaklar	 Дурева Д., М. Касева, Г. Тупаров, Компютърно моделиране, 4. клас, Просвета, 2018, София (Dureva, D., М. Kaseva, G. Tuparov, Kompyutarno modelirane, 4. klas, Prosveta, 2019, Sofia)
Öğrenciler İçin	
Kaynaklar ve	Öğrenci için talimatlar (C4G2_InstructionsForStudent.docx)
Materyaller	





Öğrenme Senaryosu 21 - Basitleştirilmiş PACMAN oyunu

Öğrenme Senaryosu	Basitleştirilmiş PACMAN oyunu
Adı	
Geçmiş Programlama	Koşullu ifadeler.
Deneyimi	 Birden çok nesneyi kodlama.
	• Tek renk algılama.
	 Döngüler (sonsuza kadar, şu ana kadar tekrarlayın).
	 Olaya dayalı nesne hareketi.
	Rastgele sayılar.
Öğrenme Çıktıları	Genel Öğrenme Çıktıları:
	Bir nesneyi klonlamak.
	Bir klonun davranışını tanımlama.
	• Yayın mesajları.
	 Mantıksal ifadelerde Boolean değer okumaları.
	 İki farklı oyun durumunu tanımlama, ayırt etme, dinamik
	olarak kontrol etme ve bunlara yanıt verme.
	Algoritmik düşünceye odaklı özel öğrenme çıktıları:
	 Öğrenci, olayları kullanarak ok tuşlarıyla nesne hareketini
	gerçekleştirir ve kısıtlamaları dikkate alır.
	 Öğrenci, orijinal nesnenin örneklerini oluşturmak için klonlar
	kullanır.
	 Öğrenci her klonun bir davranışını nasıl kodlayacağını bilir.
	 öğrenciler mesaj göndermenin anlamını bilir.
	 Öğrenci klondan artırma sayacına bir mesaj göndermeyi uygular.
	 Öğrenci, nesne tarafından alınan mesajın nasıl tespit edileceğini bilir
	ve uygun bir yanıt verir.
Amaç, Görevler ve	Kısa açıklama: Ana karakterin rastgele konumlandırılmış yıldızları
Faaliyetlerin Kısa	toplayıp bir hayalet tarafından kovalandığı oyunu programlayın.
Tanıtımı	Görevler: Öğrenciler, ana karakteri bir labirentin içinde hareket
	edecek şekilde programlamalıdır. Ana karakterin duvarlardan





	geçmemesi için hareket kısıtlamaları uygulamalılar. Daha sonra, oyun
	başladığında kendini klonlayacak bir yıldız nesnesi programlamaları ve
	ardından karakterin yıldızı her topladığında rastgele yeni bir konumda
	belirecek şekilde programlamaları gerekiyor. Toplanan yıldızların
	değerini saklamalı ve oyuncu 20 yıldız topladığında oyunu bitirmelidir.
	Oyunu daha ilginç hale getirmek için labirentte rastgele hareket
	edecek kötü bir hayalet programlayabilirler. Bir oyuncu hayalete
	dokunursa oyun biter.
	Bu aktivite ile öğrenciler, önceki aktivitelerde öğrendikleri duyu renk
	bloğunu kullanarak bir labirent içindeki hareket hakkındaki
	bilgilerini gözden geçirecekler. Nesneyi konum kısıtlamaları ile
	klonlama kavramına ve kendi rastgele hareketiyle çok basit bir
	oyuncu olmayan karakterin nasıl yaratılacağına tanıtılacaklar.
Faaliyetin Süresi	90 dakika
Öğrenme ve Öğretme	Aktif öğrenme, isbirlikci öğrenme, problem cözme
Strateji ve Metotları	
Öğretme Formları	Ön öğretme
	Birevsel calısma/ ikili calısma/ grup calısması
Öğretme Özeti	(Motivasyon-Giriş, Uygulama, Düşünme ve değerlendirme)
	Oyuncu, kırmızı bir hayalet tarafından kovalanırken rastgele
	yerleştirilmiş yıldızları toplar. Bir oyuncu ve hayalet çarpışırsa oyun
	biter. Bir oyuncu 20 yıldızı toplarsa kazanır.
	[Adım 1]
	Öğrencilere, oyuncunun hareket etmesine izin verilen alanın tek renkli
	(örneğin mavi) olduğu bir labirent ve oyuncu hareketini durduran
	başka bir renkte (örneğin siyah) renklendirilmiş duvarlar tasarlamaları
	talimatını veriyoruz. Zaman kazanmak için labirentin arka plan resmini
	önceden hazırlayabiliriz.







[Adım 2]

Öğrencilerin Pacman ve kırmızı hayaleti çizmeleri gerekiyor. Yıldız çizmek için Snap! içine bir daire çizebiliriz.



[Adım 3]

Pacman'i hareket ettirmek için farklı olasılıklar kullanabiliriz. Aşağıdaki örnek bunlardan biridir. İçinde hangi tuşa, sola, sağa, yukarı veya aşağı basıldığını tespit etmek için bir event (olay) sistemi kullanıyoruz. Bu olayların her biri gerçekleştikten sonra, hareket etmesine izin verilen alanın rengine dokunup dokunmadığını test etmeliyiz. Doğru renge dokunduysa önce o yöne döner ve hamleyi yapar. Ancak duvarların rengine dokunursa geri hareket etmesi gerekir. Zira ilk belirlediğimiz koşul nedeniyle duvara sıkışıp kalacaktır.







[Adım 4]

Sonraki görev yıldızları programlamaktır. Birçok yıldız olacak ancak hepsi aynı olmalı. Bu durumda, birden çok özdeş nesne yapmaktansa (bizim durumumuzda 20), bir nesne yapıp sonra onun klonlarını oluşturmak daha iyidir. Oyunun başında ilk klon labirentin içinde rastgele görünecek, ardından oyuncu onu topladığında kaybolacak ve farklı rastgele bir yerde yeni bir tane oluşturulacak. Oyunun başında ilk klonu oluşturmak için bu kodu bir Sahne kod alanına koyuyoruz.



Orijinal bir nesneyi gizlemek ve sadece klonları göstermek için bunu programın başında yapmalıyız.

Uygun rastgele konumlar bulmak için belirli kısıtlamalara uymamız gerekir. Bir duvarda bir yıldız yaratılırsa oyuncu ona ulaşamaz. Yani onu oraya yerleştiremeyiz. Burada kullanacağımız strateji:





1. Yıldız klonunun rastgele x, y konumunu bulmalıyız. Hem x hem de y koordinatları aynı aralıktadır [-140, 140]. Bu nedenle, her ikisi için de bu aralıktan rastgele bir sayı seçiyoruz.

2. Daha sonra bu klonun duvarın renkli kısmına değip değmediğini kontrol ederiz. Bu durumda konumu uygun değildir.

3. Konum uygunsa, klonu göstermeliyiz (unutmayın, orijinal olan gizlidir ve klon da gösterme bloğunu kullanmazsak gizlenecektir) ve sonsuz döngüde oyuncuyla çarpışma olup olmadığını kontrol etmeliyiz.

4. Konum uygun değilse, yeni bir klon oluştururuz (yeni klon için rastgele sayıların seçileceğini ve böylece uygun bir yere yerleştirileceğini umuyoruz) ve bunu siliyoruz.

5. Toplanan klonları saymak için, klonun dışında (örneğin oynatıcıda) tanımlanması gereken toplam yıldız sayısı bilgisini vermemiz gerekir. Bu, çarpışmanın meydana geldiğine dair bir mesaj yayınlayarak yapılabilir. Sonra onu silebiliriz.



[Adım 5]

Sonra bir hayalet (ghost) programlıyoruz. Labirentte rastgele hareket etmesi ve duvara çarptığında yönünü değiştirmesi gerekiyor. Hareketini rastgele hale getirmek için, çarpmadan sonra rastgele bir yönde hareket etmesini istiyoruz.

Snap'!te yönler derece ile ifade edilir.





1. 0 derece - YUKARI (UP)
2. 180 derece - AŞAĞI (DOWN)
3. 90 derece - SAĞ (RIGHT)
4. 270 derece – SOL (LEFT)
Diğer bir deyişle, O'dan 3'e rastgele bir sayı seçip 90 (derece) ile
çarparsak, rastgele bir yön elde ederiz.
Bir Pacman ile çarpışana kadar hareket etmesi gerekiyor. Çarpışınca
oyun bitecektir.
<pre>when i clicked set direction = to] repeat until touching pecman ? point in direction direction move • steps if touching ? move • steps set direction * to pick random • to 3 * 00 sy GAMEOVER for ? secs top al *</pre>
[Adım 6]
Şimdi oyuncunun oyunu ne zaman kazanacağını programlamalıyız.
Oyuncu 20 yıldız topladığı zaman oyunu kazanmış olacaktır. Pacman
senaryosunda bir yıldız sayacımız var. Başlangıçta onu O olarak
başlatırız ve ardından klon, oyuncunun topladığına dair bir mesaj
gönderdiğinde sayaç değerini 1 arttırırız. Sayaç 20'ye gelirse Pacman
kazanır ve oyunu durdurmamız gerekir.





	when clicked go to x: • y: • set tooke * to • set tooke * to • set tooke * to • say Pacmanwins! for • secs stop all*
Öğretmenler İçin	Snap!'teki tüm aktivite
Araçlar ve Kaynaklar	https://snap.berkeley.edu/project?user=zapusek&project=pac man_clone
	 Lajovic, S. (2011). Scratch. Nauči se programirati in postani računalniški maček. Ljubljana: Pasadena.
	 Vorderman, C. (2017). Računalniško programiranje za otroke. Ljubljana: MK.
Öğrenciler İçin	Snap!'teki şablon:
Kaynaklar ve	https://snap.berkeley.edu/project?user=zapusek&project=pacman_te
Materyaller	<u>mplate</u>
	 Öğrenciler için talimatlar (C4G2_InstructionsForStudent.docx)



REFERANSLAR

Co-funded by the Erasmus+ Programme of the European Union

Lajovic, S. (2011). Scratch. *Nauči se programirati in postani računalniški maček*. Ljubljana: Pasadena.

Rugelj, J. (2019). Game design based learning of programming.

Vorderman, C. (2017). *Računalniško programiranje za otroke*. Ljubljana: MK.