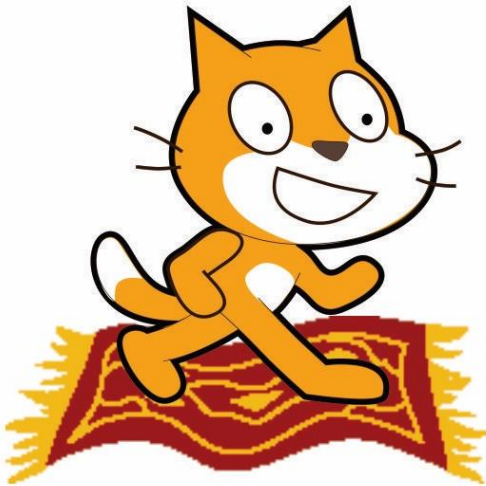


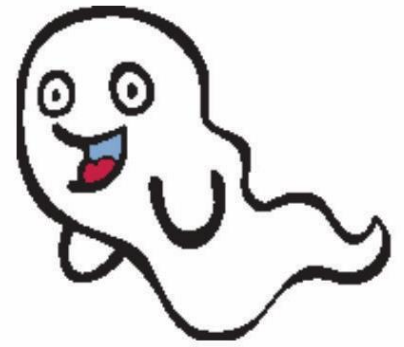
Abdulla Qəhrəmanov, Sevda Sadıqova, Dilərə Hacıyeva,
Könül Rəcəbova, İlahə Cəfərova

SCRATCH 2.0

Proqramlaşdırma dili



I



A.Qəhrəmanov, S.Sadiqova, D.Hacıyeva

K.Rəcəbova, İ.Cəfərova

SCRATCH-2

Programlaşdırma dili

I hissə

Bakı - 2016



AZƏRBAYCAN RESPUBLİKASI

TƏHSİL NAZİRLİYİ

Bu kitabın elektron versiyası Azərbaycan Respublikası Təhsil Nazirliyinin Təhsildə inkişaf və innovasiyalar üzrə grant müsabiqəsi çərçivəsində həyata keçirilən layihə çərçivəsində hazırlanmışdır.

Təhsil Nazirliyi nəşrin məzmununa görə məsuliyyət daşımır.

Scratch – Lifelong Kindergarten MIT (Massachusetts Institute of Technology) Media Lab (Massachusetts Texnologiyalar İnstitutunun Media Laboratoriyasının Ömür boyu Uşaq Bağçası) qrupunun layihəsidir. SCRATCH kiçik yaşlı uşaqlara proqramlaşdırmanı öyrətmək üçün nəzərdə tutulmuş pulsuz yeni tədris vasitəsidir. SCRATCH müasir dövrdə erkən yaşdan proqramlaşdırmanın tədrisi istiqamətində təqdim edilən ən uğurlu layihələrdən biridir. Bu layihə 5-7 yaşlı uşaqlar üçün nəzərdə tutulmuş SCRATCHjr və 7 yaşdan yuxarı uşaqlar üçün nəzərdə tutulmuş SCRATCH proqramlaşdırma dillərini özündə ehtiva edir. Uşaqlar SCRATCH proqramlaşdırma dilində LEGO oyununda olduğu kimi kərpiclərdən öz oyunlarını, animasiya filmlərini və digər kvestlərini (macərələrini) qurur, eyni zamanda riyaziyyatı, məntiqi və proqramlaşdırmanı öyrənirlər. Yaşından asılı olmayaraq SCRATCH proqramlaşdırma dilində işləmək uşaqların çox xoşuna gəlir! Sizə təqdim edilən bu vəsait SCRATCH 2.0 proqramlaşdırma dilinin sərbəst öyrənilməsi üçün Azərbaycan dilində hazırlanmış ilk vəsaitin I hissəsidir.

Vəsaitdə verilən material yaşı 7 - 14 arasında olan, oxumağı bacaran, kompüterini idarə etmək üzrə baza bacarıqlarına malik uşaqlar üçün nəzərdə tutulub. Bu vəsaitlə işləmək üçün uşaq minimum olaraq mouse (siçan) qurğusu ilə işləməyi, qovluq və qovluqlarda axtarış aparmağı, proqramı başlatmağı, faylı yadda saxlamağı, üç rəqəmli ədədləri toplamağı və çıxmağı, adı vurma və bölmə əməllərini yerinə yetirməyi bacarmalıdır. Vəsaitdə bu yaş üçün daha çətin olan riyazi anlayışlar (mənfi ədəd, onluq kəsir, koordinat oxu, dərəcə, faiz) haqqında məlumatlar, musiqi haqqında minimal bilgiler barədə vəsaitə əlavədə izahat verilir.

SCRATCHjr proqramı ilə tanışlığı olan uşaqlar üçün bu vəsaitlə tanışlıq çox asan olacaq. Çünki burada da əmrlər pazlları xatırladan bloklar formasında verilib. Sadəcə olaraq öz hekayənə (kvestinə) uyğun olaraq həmin blokları bir-birinə calaşdırıb proqram qurmaq lazımdır. Bu proqramlar Sənin gələcəkdə peşəkar proqramçı kimi yetişməyində ilk real addımlardır.

Yuxarıda deyildiyi kimi təqdim edilən bu vəsait SCRATCH 2.0 proqramlaşdırma dilinin sərbəst öyrənilməsi üçün Azərbaycan dilində hazırlanmış ilk vəsaitin I hissəsidir. I hissədə əsasən SCRATCH 2.0 proqramlaşdırma dilinin operatorları ilə tanış etmək, nisbətən sadə proqramlar yazmaq və sadə alqoritmləri bu proqramlaşdırma dilində necə realizə etmək nəzərdə tutulub.

ISBN 978-9952-435-86-3

A.Qəhrəmanov, S.Sadıqova, D.Hacıyeva, K.Rəcəbova, İ.Cəfərova – SCRATCH-2
proqramlaşdırma dili (I hissə) -128 səh. – Müəllim Nəşriyyatı – tiraj 300 – Bakı – 2016



Əziz **SCRATCH** sevən oxucu!

Lifelong Kindergarten MIT (Massachusetts Institute of Technology) Media Lab (Massaçuset Texnologiyalar İnstitutunun Media Laboratoriyasının Ömür boyu Uşaq Bağçası) qrupunun layihəsi olan **SCRATCH** müasir dövrdə erkən yaşdan proqramlaşdırmanın tədrisi istiqamətində təqdim edilən ən uğurlu layihələrdən biridir. Bu layihə 5-7 yaşlı uşaqlar üçün nəzərdə tutulmuş **SCRATCHjr** və 7 yaşdan yuxarı uşaqlar üçün nəzərdə tutulmuş **SCRATCH** proqramlaşdırma dillərini özündə ehtiva edir. **SCRATCH** kiçik yaşlı uşaqlara proqramlaşdırmanı öyrətmək üçün nəzərdə tutulmuş pulsuz yeni tədris vasitəsidir.

Uşaqlar **SCRATCH** proqramlaşdırma dilində LEGO oyununda olduğu kimi kərpiclərdən öz oyunlarını, animasiya filmlərini və digər kvestlərini (macərələrini) qurur, eyni zamanda riyaziyyatı, məntiqi və proqramlaşdırmanı öyrənirlər. Yaşından asılı olmayaraq **SCRATCH** proqramlaşdırma dilində işləmək uşaqların çox xoşuna gəlir! Sizə təqdim edilən bu vəsait **SCRATCH 2.0** proqramlaşdırma dilinin sərbəst öyrənilməsi üçün Azərbaycan dilində hazırlanmış ilk vəsaitin I hissəsidir. Müəlliflər bu vəsaiti hazırlayarkən dünyada mövcud ingilis, türk və rus dilində hazırlanmış oxşar bir çox vəsaitləri araşdırmışlar. Təqdim olunan bu vəsait dünya təcrübəsi nəzərə alınmaqla hazırlanmışdır.

Vəsaitdə verilən material yaşı **7 - 14** arasında olan, oxumağı bacaran, kompüteri idarə etmək üzrə baza bacarıqlarına malik uşaqlar üçün nəzərdə tutulub. Bu vəsaitlə işləmək üçün uşaq minimum olaraq mouse (siçan) qurğusu ilə işləməyi, qovluq və qovluqlarda axtarış aparmağı, proqramı başlatmağı, faylı yadda saxlamağı, üç rəqəmli ədədləri toplamağı və çıxmağı, adi vurma və bölmə əməllərini yerinə yetirməyi bacarmalıdır. Vəsaitdə bu yaş üçün daha çətin olan riyazi anlayışlar (mənfi ədəd, onluq kəsr, koordinat oxu, dərəcə, faiz) haqqında məlumatlar, musiqi haqqında minimal bilgilər barədə vəsaitə əlavədə izahat verilir.

SCRATCHjr proqramı ilə tanışlığı olan uşaqlar üçün bu vəsaitlə tanışlıq çox asan olacaq. Çünki burada da əmrlər pazlları xatırladan bloklar formasında verilib. Sadəcə olaraq öz hekayənə (kvestinə) uyğun olaraq həmin blokları bir-birinə calaşdırıb proqram qurmaq lazımdır. Bu proqramlar Sənin gələcəkdə peşəkar proqramçı kimi yetişməyində ilk real addımlardır.

Yuxarıda deyildiyi kimi təqdim edilən bu vəsait **SCRATCH 2.0** proqramlaşdırma dilinin sərbəst öyrənilməsi üçün Azərbaycan dilində hazırlanmış ilk vəsaitin I hissəsidir. I hissədə əsasən **SCRATCH 2.0** proqramlaşdırma dilinin operatorları

ilə tanış etmək, nisbətən sadə proqramlar yazmaq və sadə alqoritmləri bu proqramlaşdırma dilində necə realizə etmək nəzərdə tutulub.

SCRATCH 2.0 proqramlaşdırma dili haqqında vəsaitin II hissəsində isə **SCRATCH 2.0** proqramlaşdırma dilində daha çətin alqoritmlərin necə realizə edilməsi nümunələrlə veriləcək.

Əziz oxucumuz, əgər kitabda nəyisə başa düşməsən, onda aşağıdakı elektron poçt və ya telefon nömrəsi vasitəsilə bizimlə əlaqə saxla və suallarını bizə çatdır.

e-mail: abdullaqehreman@gmail.com

Telefon: (+99450) 3503920

Hörmətlə:

Müəlliflər



Mündəricat

1. SCRATCH nədir?	8
2. SCRATCH proqramının quraşdırılması.....	9
2.1. Online rejimdə SCRATCH proqramı ilə iş.....	9
2.2. Offline rejimdə SCRATCH proqramı ilə iş	13
3. SCRATCH proqramı ilə iş.....	14
3.1. SCRATCH proqramının interfeysi.....	14
3.2. SCRATCH proqramının əmrləri	20
3.3. Əmrlər (команды)	21
3.3.1. Motion. Hərəkət qrupu. Группа движение.....	21
3.3.2. Looks. Görüntü qrupu. Группа внешность	23
3.3.3. Sound. Səs qrupu. Группа звук.....	26
3.3.4. Pen. Qələm qrupu. Группа перо	28
3.3.5. Data. Verilənlər qrupu. Группа данные.	29
3.3.6. Events. Hadisələr qrupu. Группа события.....	34
3.3.7. Control. İdarəetmə qrupu. Группа контроль	36
3.3.8. Sensing. Sensor (hissetmə) qrupu. Группа сенсоры	38
3.3.9. Operators. Operatorlar qrupu. Группа операторы	42
3.3.10. More Bloks. Digər bloklar. Другие блоки.....	44
3.4. Səhnənin quruluşu.....	49
3.5. Spraytlarla iş.....	49
4. İlk proqramımız.....	51
4.1. Nədən başlamalı.....	52
4.2. Geriyə hərəkət.....	59
4.3. Rəqs edən uşaq.....	62
4.4. İstiqamət.....	70
4.5. Klavişlərlə idarəetmə.....	73
4.6. Dəyişənlər	77
4.7. İt və pişik	81
4.8. Proqramda riyaziyyat	90
5. Müxtəlif proqram nümunələri	93
5.1. Üçbucaq, kvadrat və digər çoxbucaqlı fiqurların çəkilməsi.....	94
5.1.1. Üçbucağın çəkilməsi	94
5.1.2. Kvadratın çəkilməsi.....	98

5.1.3. İxtiyari çoxbucaqlının çəkilməsi	99
5.2. Dairələr.....	102
5.3. Mouse qurğusu ilə manipulyasiya.....	105
6. Səs və musiqi	108
6.1. Səs və musiqi	109
6.2. Səsin gücünün dəyişdirilməsi	110
6.3. Səslə iş. Misallar	111
6.4. DO-RE-Mİ – dən “CÜCƏLƏRİM”-ə qədər	111
6.5. “A” = DO “S” = RE “D” = Mİ və sairə	113
7. Son söz əvəzi.....	115
8. Ə L A V Ə L Ə R.....	116
8.1. Saniyənin hissələri və ya kəsr	116
8.2. Mənfi ədəd.....	118
8.3. Dekart koordinat sistemi	120
8.4. Bucaq	122
8.5. Faiz nədir?	124
8.6. Musiqi və notlar	125
İstinadlar	128



1. SCRATCH nədir?

Scratch – Lifelong Kindergarten MIT (Massachusetts Institute of Technology) Media Lab (Massaçuset Texnologiyalar İnstitutunun Media Laboratoriyasının Ömür boyu Uşaq Bağçası) qrupunun layihəsidir.

Scratch proqramının ilk test versiyası Mitçel Reznikin (Mitchel Resnick) rəhbərliyi altında yuxarıda adı çəkilən qrup tərəfindən 2002-ci ildə yaradılmışdır. Daha sonra o təkmilləşdirilmişdir. Aşağıdakı cədvəldə Scratch proqramının müxtəlif versiyalarının rəsmi istismara verilməsi tarixləri verilib.

Versiya	Dövr	QEYD
Scratch 0 (0.1). Beta versiya	2002-2006 (Noyabr 2006)	Erkən versiyalardır. İnterfeys və bloklar dəfələrlə dəyişilib. Müəllimlər üçün nəzərdə tutulub.
Scratch 1.0	8 Yanvar 2007	1.X versiyalarının başlanğıcı qoyulub. İctimaiyyətə kütləvi yayım üçün nəzərdə tutulub.
Scratch 1.1	May 2007	
Scratch 1.2	7 Dekabr 2007	
Scratch 1.3	2 Sentyabr 2008	
Scratch 1.4	19 İyul 2009	Bu versiya proqramın populyarlığını artırıb.
Scratch 2.0	9 May 2013	Bu versiya üzərində iş bir neçə il gedib.
Scratch 3.0	Gələcəyin versiyasıdır	Üzərində iş gedir. Ola bilsin ki, çox sayda əsaslı dəyişikliklər olsun.
Scratch Jr	2014	iPad və Android əməliyyat sistemli tabletləri üçün nəzərdə tutulub.

Bu kitabda Scratch 2.0 versiyası ilə iş öz əksini tapıb. Gələcəkdə sadəcə Scratch işarə ediləcək.

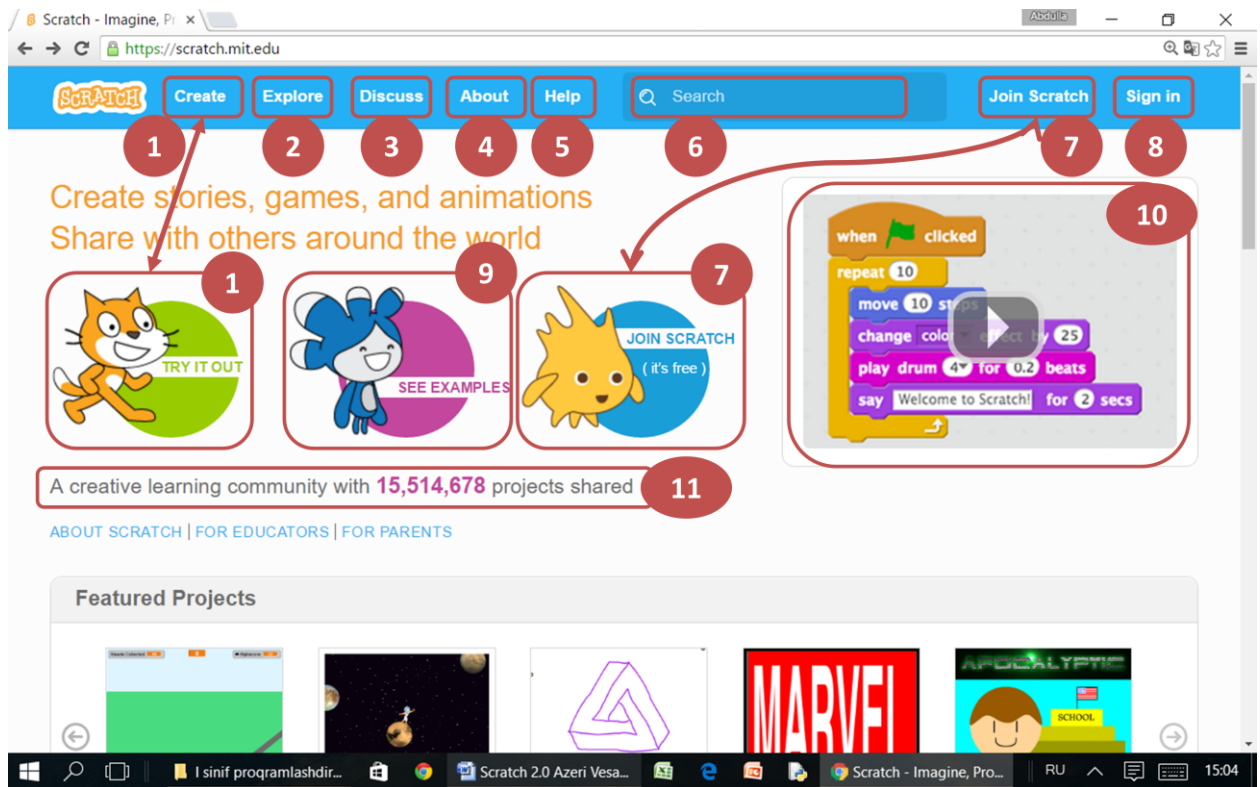
2. SCRATCH proqramının quraşdırılması

Scratch 2.0 versiyası ilə iki üsulla işləmək olar:

- Online
- Offline

2.1. Online rejimdə SCRATCH proqramı ilə iş

MIT təşkilatı öz saytında Scratch proqramı üçün səhifə açmışdır. scratch.mit.edu ünvanı ilə həmin səhifəyə daxil olmaq olar. Səhifəyə daxil olduqda aşağıdakı görüntü alınacaq:



- 1 - **Create (TRY IT OUT)**. İki nöqtədən Scratch 2.0 proqramına giriş.
- 2 - **Explore**. Hazır işləri tədqiq etmək.
- 3 - **Discuss**. Diskussiya aparmaq.
- 4 - **About**. Layihə haqqında məlumat.
- 5 - **Help**. Kömək.
- 6 - **Search**. Axtarış.
- 7 - **Join Scratch**. İki nöqtədən Scratch saytında qeydiyyatdan keçmək üçün giriş.
- 8 - **Sign in**. Daxil olmaq. Artıq qeydiyyatda olanlar üçün.
- 9 - **SEE EXAMPLES**. Saytda yerləşdirilmiş nümunələrə baxmaq.
- 10 - Scratch proqramı ilə işləmək üçün kömək üçün video görüntülər.



11 - Sayta daxil olan halda orada yerləşdirilmiş işlərin sayı.

Join Scratch (7-ci) düyməsini sıxdıqda Scratch saytında qeydiyyatdan keçmək üçün ekranda aşağıdakı pəncərə açılacaq:

Choose a Scratch Username sahəsində özünüzün Scratch məkanında adınızı yazın. Burada öz adınızı yazmaq vacib deyil (Bizim nümunədə *Qartal3*).

Choose a Password sahəsində parolunuzu yazın.

Confirm Password sahəsində parolunuzu təkrar edin.

Daha sonra aşağı sağ küncdəki **Next** düyməsini sıxaraq işinizə davam edin.

Növbəti pəncərə belə olacaq:

Burada **Birth Month and Year** sahəsində doğduğunuz ayı və ili seçin.

Gender sahəsində cinsinizi seçin. **Male** – kişi, **Female** – qadın.

Country sahəsində ölkənizin adını seçin. Azərbaycanda yaşayanlar üçün **Azerbaijan**.

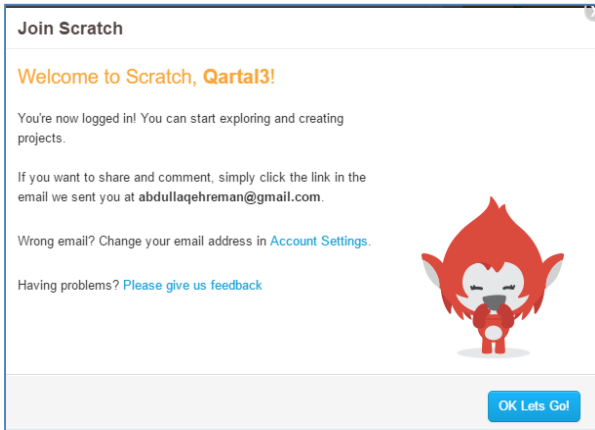
Yenə aşağı sağ küncdəki **Next** düyməsini sıxaraq işinizə davam edin.

Növbəti pəncərə belə olacaq:

Növbəti 2 sahədə valideyninizin və ya qəyyumunuzun e-mail ünvanını yazıb təkrar təsdiq edin.

Sonuncu dəfə aşağı sağ küncdəki **Next** düyməsini sıxaraq işinizə davam edin.

Növbəti pəncərə belə olacaq:

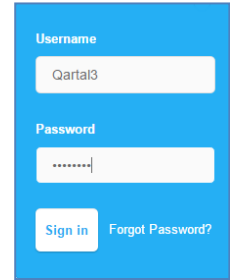


Bu pəncərə artıq Sizin Scratch məkanına qoşulduğunuzu bildirir. Əgər ehtiyac olarsa, **Account Settings** düyməsini sıxmaqla öz qeydiyyat parametrlərinizdə dəyişiklik apara bilərsiniz. Ehtiyac yoxdursa, bu düyməni sıxmayın.

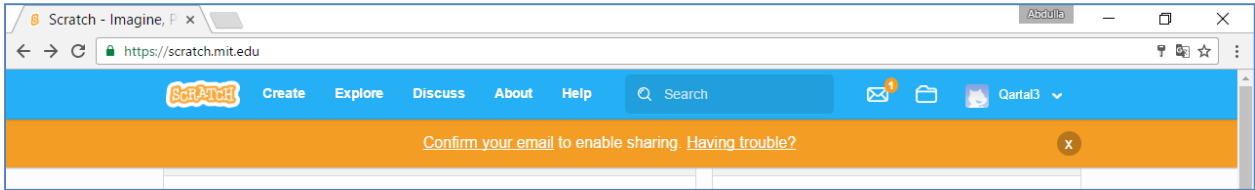
Sonda aşağı sağ küncdəki **OK Lets Go!** düyməsini sıxaraq işinizi sona çatdırın. Siz artıq Scratch məkanında öz


profilindəsiniz. Səhifənin yuxarı sağ küncündə Sizin **Akkauntunuz (adınız)** görünəcək.

Əgər Siz artıq qeydiyyatdan keçmişsinizsə, onda yuxarı sağ küncdəki **Sign in** düyməsini sıxın və açılan pəncərədə öz Akkauntunuzla parolunuzu yığıb aşağıdakı ağ sahədəki **Sign in** düyməsini sıxın.



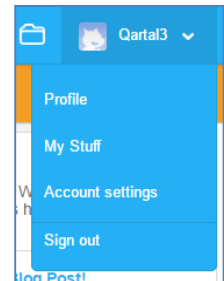
Nəticədə Siz öz profilinizə uyğun səhifəyə girəcəksiniz. Görüntü belə olacaq:



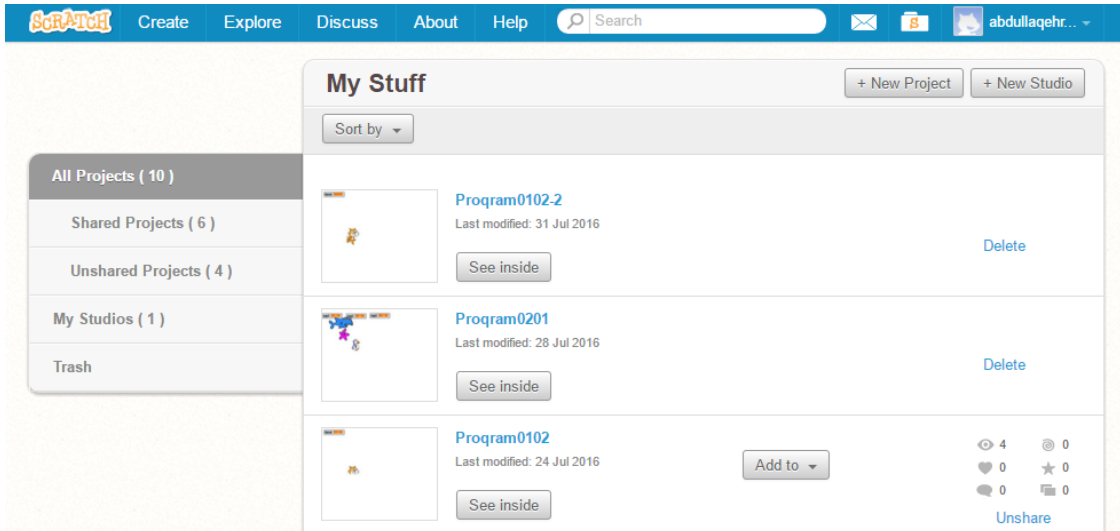
Profilinizdə olarkən  düymələri sıxmaqla Sizə gələn mesajlara və Sizə aid işlərinizə baxa bilərsiniz.

Açılan səhifənin yuxarı sağ küncündə Sizin Akkauntunuz görünəcək. Ondan sağda v düyməsini sıxmaqla Siz sağda görünən menyunu açacaqsınız. Burada aşağıdakı menyulara keçidlər var:

Profile – Profil. Bu düyməni sıxdıqda Siz öz profilinizə əlavə və ya düzəliş edə bilərsiniz.



My Stuff – Mənim işlərim. Bu düyməni sıxdıqda Sizin profilinizdə olan işləriniz haqqında məlumat olan aşağıdakı səhifə açılacaq.



Səhifənin sol tərəfində aşağıdakı menyü görünəcək:

All Projects (10) – Bütün layihələr. Bu menyunu seçdikdə sağda Sizin profilinizdə olan bütün layihələr görünəcək. Lazım olmayan layihəni **Delete** düyməsini sıxmaqla silə bilərsiniz.

Shared Projects (6) - Bu menyunu seçdikdə sağda Sizin profilinizdə olan və Scratch məkanında paylaşılmış bütün layihələr görünəcək. Lazım olmayan layihəni **Delete** düyməsini sıxmaqla silə bilərsiniz.

Unshared Projects (4) - Bu menyunu seçdikdə sağda Sizin profilinizdə olan və Scratch məkanında paylaşılmamış bütün layihələr görünəcək. Lazım olmayan layihəni **Delete** düyməsini sıxmaqla silə bilərsiniz.

My Studios (1) – Əgər Siz əlavə olaraq studiya da yaratmışınızsa, onda o studiyalar və oradakı layihələr haqqında məlumatı bu menyudan ala bilərsiniz.

Trash – Zibil qutusu. Sildiğiniz layihələr orada saxlanacaq. İstəsəniz həmin menyuya daxil olub oradakıları silə bilərsiniz.

Əvvəlki menyuya qayıdaq.

Account setting – Akkauntun sazlanması. Bu düyməni sıxdıqda Sizin Akkauntunuzda olan məlumatlar (ad, ölkə, parol və e-mail) dəyişdirilə bilər.

Sign Out – Çıxış. Bu düyməni sıxdıqda Siz öz profilinizdən çıxma bilərsiniz.

Səhifənin yuxarisindəki menyuda **Create** və ya yuxarıda göstərdiyimiz kimi **scratch.mit.edu** ünvanı daxil olduğumuz səhifədə **TRY IT OUT (1)** düyməsini sıxdıqda Scratch proqramını online rejimdə başlada bilərik.

2.2. Offline rejimdə SCRATCH proqramı ilə iş

MIT təşkilatı öz saytında Scratch proqramı üçün bölmə açmışdır. scratch.mit.edu ünvanı ilə Siz Scratch 2.0 proqramının Offline versiyasını yükləyə bilərsiniz.

Scratch 2.0 Offline versiyasını yükləmək üçün aşağıdakı ardıcılığını yerinə yetirin.



Yüklənmiş proqramı (versiyadan asılı olaraq proqramın adı fərqli ola bilər. Məsələn, mən proqramı quraşdıran zaman onun adı **Scratch-447.exe** idi.) kompüterə quraşdırırıq. Proqram kompüterdə quraşdırıldıqdan sonra iş masasında



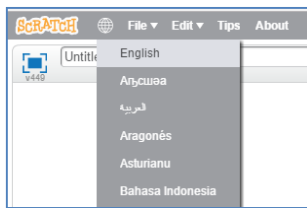
qısayolu yaranacaq. Onun üzərinə iki dəfə sıxsaq, Scratch proqramı offline rejimdə başladılacaq.



3. SCRATCH proqramı ilə iş

3.1. SCRATCH proqramının interfeysi

Online və ya Offline rejimdən asılı olmayaraq Scratch 2.0 proqramının interfeysi aşağıdakı kimidir:



Proqramın yuxarı sol küncündə qlobus işarəsi var. Qlobusun üzərinə sıxdıqda Scratch proqramının interfeysinin hansı dildə işləməsini göstərən siyahı çıxır. Sizə lazım olan dili seçin və interfeysi istədiyiniz dilə köçləyin. Bizim nümunədə ingilis dili seçilib.

Scratch proqramının interfeysində **1** nömrəli sahədə Windows əməliyyat sistemi ilə işləyən proqramlarda olduğu kimi menyü bölməsi yerləşir.

Bu bölmədə aşağıdakı menyular File Edit Tips About mövcuddur:

- **File (Fayl)** – **Edit (Redaktə)** – **Tips (Kömək)** – **About (Haqqında)**

File və Edit menyuları **2** variantda açıla bilər. Adi menyunu seçəndə və Klaviaturada Shift klavişini sıxılı vəziyyətdə saxlamaqla menyunu seçmək. Shift klavişini sıxılı vəziyyətdə saxlayaraq bu iki menyunu açdıqda ekranda genişlənmiş

menyu açılır. Genişlənmiş menyu adı menyudan aşağıda əlavə sətirlərin olması ilə fərqlənir.

File menyusunun açılışı aşağıdakı sətirlərə malikdir:

Menyunun sətiri	Təyinatı	Online	Offline
New	Yeni	+	+
Open	Açmaq		+
Save	Yadda saxlamaq		+
Save now	İndi yadda saxlamaq	+	
Save As	Başqa adla yadda saxlamaq.		+
Save As a copy	Surətini başqa adla yadda saxlamaq.	+	
Go to My Stuff	Mənim işlərimə keçid	+	
Upload from your computer	Sənin kompüterindən yükləmək	+	
Download to your computer	Sənin kompüterinə yükləmək	+	
Record Project Video	Layihənin videosunu çəkmək.		+
Record & Export Video	Videonu çəkib eksport etmək.	+	
Revert	Geriyyə qaytarmaq.	+	
Share to website	Saytda nəşr etdirmək.		+
Check for updates	Yenilənmələrin mövcudluğunu yoxlamaq.		+
Quit	Çıxış.		+
Force help file update	Arayış faylını məcburi yeniləmə.		+
Save Project Summary	Layihənin xülasəsini yadda saxlamaq	+	
Show version details	Versiyanın detallarını göstərmək.	+	+
Import experimental HTTP extension	Eksperimental universal HTTP genişlənməsini import etmək.		+

Qeyd: Sarı fonda əlavə sətirlər göstərilib.



Edit menyusunun açılışı aşağıdakı sətirlərə malikdir:

Menyunun sətiri	Təyinatı	Online	Offline
Undelete	Bərpa etmək	+	+
Small stage layout	Kiçik səhnə	+	+
Turbo mode	Turbo rejim	+	+
Export translation strings: commands	Tərcümə sətirlərinin eksportu-əmərlər	+	+
Export translation strings: UI	Tərcümə sətirlərinin eksportu-UI	+	+
Export help screen names	Kömək ekran adlarının eksportu	+	
Edit block colors	Blok qruplarının rənginin redaktəsi	+	
MediaLib-media	Media kitabxana - media	+	
MediaLib-sprites	Media kitabxana - spraytlar	+	
MediaLib-check JSON file	Media kitabxana – JSON faylının dəyişilməsi	+	

Qeyd: Sarı fonda əlavə sətirlər göstərilib.



Scratch proqramının interfeysində **2** nömrəli sahədə şəkildəki alətlər yerləşir. Bu alətlərin təyinatı belədir:

1 – Duplicate (dublikat). Bu düyməni sıxıb kursoru mövcud spraytlardan birinin üzərinə gətirməli və onun üzərində sıxmalı. Əməliyyatdan sonra həmin spraytın skriptləri ilə birlikdə dublikatı alınacaq.

2 – Delete (silmək). Bu düyməni sıxıb kursoru mövcud spraytlardan birinin üzərinə gətirməli və onun üzərində sıxmalı. Bu halda həmin sprayt silinəcək.

3 – Grow (böyütmək). Bu düyməni sıxıb kursoru mövcud spraytlardan birinin üzərinə gətirməli və istədiyiniz ölçünü alana qədər onun üzərində sıxmalı. Bu halda həmin spraytın ölçüsü böyüyəcək.

4 – Shrink (kiçiltmək). Bu düyməni sıxıb kursoru mövcud spraytlardan birinin üzərinə gətirməli və istədiyiniz ölçünü alana qədər onun üzərində sıxmalı. Bu halda həmin spraytın ölçüsü kiçiləcək.

5 – Block help (blok-skript üzrə kömək). Bu düyməni sıxıb kursoru istifadə edilən skriptlərdən birinin üzərinə gətirməli və onun üzərində sıxmalı. Bu halda həmin skript barədə ingilis dilində köməkçi məlumat görünəcək.

1, 2 və 5-ci düymələrin işini obyektin üzərində sağ düyməni sıxıb alınan kontekst menyudakı uyğun sətiri seçməklə də yerinə yetirmək olar.



Scratch proqramının interfeysi şərti olaraq 4 hissəyə bölünür. Şəkildə gördüyünüz kimi interfeysin sahələri nömrələnib. Bəzi müəlliflər 3 və 4 nömrəli sahələri birgə götürüb sayı 3 göstərirlər.


3 nömrəli sahə **SƏHNƏ (ing. STAGE)** adlanır. Sizin yazdığımız proqram işini səhnədə yerinə yetirəcək. Scratch proqramlaşdırma dilində istifadə edilən personaj (lar) **SPRAYT** adlanır. Proqram hər dəfə yenidən yükləndə və ya yeni fayl açılarda proqramda cəmi bir ədəd sprayt olur. O da xoş sifətli pişikdir. Səhnənin aşağı sağ küncündə cari spraytın və kursurun səhnədə durduğu yerin koordinatı göstərilir. Məsələn: **x: 240 y: -180**.

Ümumiyyətlə səhnədəki nöqtələrin yerini göstərmək üçün riyaziyyatda istifadə edilən **dekart koordinat sistemindən** istifadə edilir. Səhnənin mərkəzinin koordinatı **x: 0 y: 0** olacaq. **x** və **y** koordinatlarının qiyməti aşağıdakı diapazonda dəyişir:

$$-240 < x < 240 \text{ və } -180 < y < 180.$$

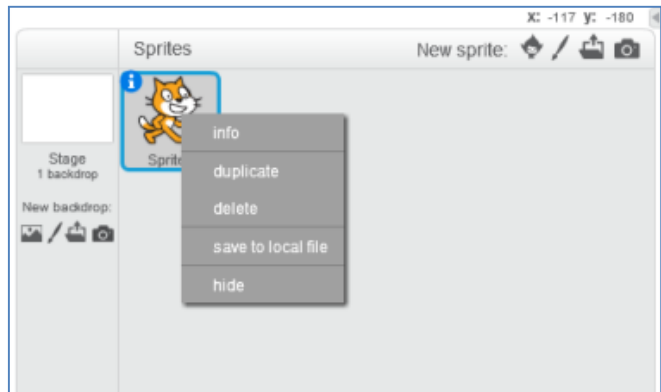
Bu vəsaitin əlavəsində dekart koordinat sistemi haqqında geniş məlumat ala bilərsiniz.

Səhnənin yuxarı sol küncündə  işarəsi var. Həmin işarənin üzərində sıxdıqda səhnə tam ekranda görünür. Tam ekran rejimində həmin nöqtədə  işarəsi görünür. Bu işarəyə sıxdıqda isə səhnə tam ekran rejimindən adi rejimə qaydır.

Səhnənin yuxarı sağ küncündə qoşa iki düymə var . Yaşıl bayraq proqramın skriptlərini başlatmaq, qırmızı daire isə proqramı dayandırmaq üçündür. Tam ekran rejimində proqramı işlədəndə yalnız bu iki düymədən istifadə etmək olur. Digər elementlər tam ekran rejimində görünür.

Səhnədən aşağıda **4** nömrəli sahəni spraytlar sahəsi adlandırmaq olar.

Sol tərəfdə Stage (Səhnə) adlanan hissədə proqramın Səhnə sahəsi





haqqında məlumat əks olunur.

Stage sözünün altında səhnədəki fonların (backdrop) sayı göstərilir.



New backdrop bölməsində yeni fon yaratmaq (əlavə etmək) üçün 4 yol göstərilib. Bunlar aşağıdakılardır:

- 1 – Scratch proqramının kitabxanasından fon seçmək. Bu düyməni sıxdıqda Scratch proqramının kitabxanasındakı mövcud fonlar görünəcək. Fonlar kateqoriyalara və mövzulara görə qruplaşdırılır. Bu forma axtarışı asanlaşdırır. All rejimini seçəndə bütün fonlar görünür.
- 2 – Scratch proqramının daxili Paint proqramından istifadə etməklə yeni fon çəkmək. Daxili Paint proqramı ilə iş sonrakı bölmələrdə təsvir ediləcək.
- 3 – İxtiyari formatlı şəkil faylı formasında olan fonu yükləmək.
- 4 – Kompüterin kamerasından istifadə etməklə yeni fon yaratmaq.

4 nömrəli sahənin sağ hissəsi sırf spraytlar (sprites) haqqında məlumatı əks etdirir.



Yuxarı sağ küncdə **New sprite** bölməsində yeni əlavə etmək üçün 4 yol göstərilib. Bunlar aşağıdakılardır:

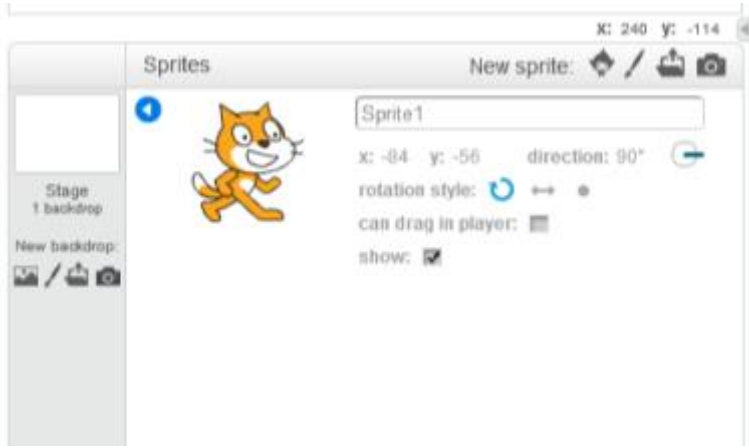
- 1 – Scratch proqramının kitabxanasından hazır sprayt seçmək. Bu düyməni sıxdıqda Scratch proqramının kitabxanasındakı mövcud spraytlar görünəcək. Spraytlar kateqoriyalara, mövzulara və tipə görə qruplaşdırılır. Bu forma axtarışı asanlaşdırır. All rejimini seçəndə bütün spraytlar görünür.
- 2 – Scratch proqramının daxili Paint proqramından istifadə etməklə yeni sprayt çəkmək. Daxili Paint proqramı ilə iş sonrakı bölmələrdə təsvir ediləcək.
- 3 – İxtiyari formatlı şəkil faylı formasında olan spraytı yükləmək.
- 4 – Kompüterin kamerasından istifadə etməklə yeni sprayt yaratmaq.

4 nömrəli sahənin orta hissəsində hal-hazırda proqramda istifadə edilən spraytlar əks olunur. Yeni proqram səhifəsi açılanda orada yalnız bir sprayt (pişik) əks olunur.

Konkret hansısa spraytın üzərində Mouse (siçan) qurğusunun sağ düyməsini sıxanda aşağıdakı menyu açılır:

info- informasiya
duplication– dublikatını almaq
delete– silmək
save to local file– lokal fayla yazmaq
hide(show)– gizlətmək (göstərmək)

info rejimini seçdikdə və ya spraytın nişanının solundakı **i** düyməsini sıxdıqda ekranda cari sprayt haqqında aşağıdakı məlumat görünəcək.

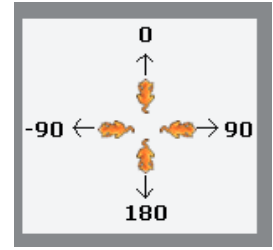



Sprayt şəklin sağındakı sahədə (bu nümunədə orada **Sprite1** yazılıb) spraytın adını istədiyiniz ada dəyişə bilərsiniz. Məsələn, “Pişik” adı vermək olar.


Obyektə öz istədiyiniz adı qoymaq proqramın oxunaqlığını asanlaşdırır.

x: və **y:** dəyişənlərinin qarşısındakı ədədlər (bu nümunədə x: -84, y: -56 göstərir) cari spraytın səhnədəki koordinat sistemindəki cari mövqeyini göstərir. Həmin koordinatlar paralel olaraq **6** nömrəli sahənin yuxarı sağ küncündə cari spraytın solğun şəklinin altında da verilir.

direction: dəyişəninin qarşısındakı ədəd (bu misalda 90 dərəcə) cari spraytın cari istiqamətini göstərir. Bucaq ixtiyarı ola bilər. Dörd qiymət xüsusi halları göstərir (0 - yuxarı, 90 - sağa, 180 - aşağı, -90 - sola).




direction dəyişənindən sağda  açarı var. Dairənin bir tərəfində yerləşən göy rəngli xəttin istiqamətini dəyişməklə istədiyimiz istiqaməti göstərə bilərik.

rotation style: fırlanma stili. Bu yazıdan sonra  üç stil göstərilmişdir. Bunlar **can rotate** (fırlana bilər), **only face left-right** (yalnız sola və sağa dönə bilər) və **don't rotate** (dönmək olmaz). Bu stilləri seçməklə cari spraytın hərəkətinin stilini vermək olar.

can drag in player: ifa zamanı daşımaq. Bu göstəricinin yanındakı xananı “✓” işarəsi ilə qeyd eləsək, onda tam ekranda ifa (icra) zamanı spraytın yerini dəyişmək olar.

show: göstərmək. Bu göstəricinin yanındakı xananı “✓” işarəsi ilə qeyd eləsək, onda sprayt görünəcək, əks halda gizlənəcək (görünməyəcək).

info rejimindən çıxmaq üçün spraytın şəklindən sağda yerləşən  düyməsini sıxmaq lazımdır.



3.2. SCRATCH proqramının əməlləri

Scratch proqramının **5** nömrəli sahəsi skriptlər (əməllər) sahəsi adlandırılır. Bu hissədə Scratch proqramlaşdırma dilinin bütün əməlləri toplanıb. Əməllər 10 kateqoriya üzrə qruplaşıb. Qrupların adları sahənin yuxarı hissəsində verilib.

Motion	Events
Looks	Control
Sound	Sensing
Pen	Operators
Data	More Blocks
Движение	События
Внешность	Управление
Звук	Сенсоры
Перо	Операторы
Данные	Другие блоки
Hareket	Olaylar
Görünüm	Kontrol
Ses	Algılama
Kalem	İşlemler
Veri	Özel Taşlar

Hərəkət qrupu. Motion. Группа движение.

Görüntü qrupu. Looks.

Группа внешность.

Səs qrupu. Sound. Группа звук.

Qələm qrupu. Pen. Группа перо.

Verilənlər qrupu. Data. Группа данные.

Hadisələr qrupu. Events. Группа события.

İdarəetmə qrupu. Control. Группа контроль.

Sensor (hissetmə) qrupu. Sensing. Группа сенсоры.

Operatorlar qrupu. Operators. Группа операторы.




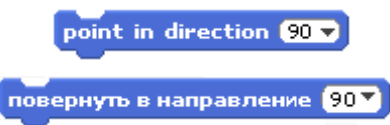
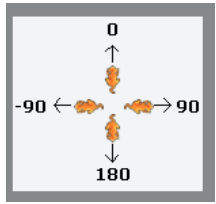
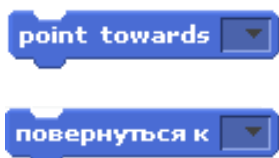


Digər bloklar. More Bloks. Другие блоки.

Şəkildən görüldüyü kimi hər qrupun öz rəngi var. Hansısa qrupu seçərkən aşağıda həmin qrup daxil olan əməllərin siyahısı görünür. Əməllərin rəngi qrupun rəngi ilə eyni olur. Əməlin rəngi onun hansı qrupa aid olmasını təsdiq edir.

Əgər diqqət etmişsinizsə, qrupların adları Azərbaycan, ingilis, rus və türk dillərində verilmişdir. Aşağıdakı cədvəldə qruplar üzrə bütün əməllər və onların təyinatı verilib. Təyinat **Azərbaycan** və **rus** dillərində verilib. Bu da Azərbaycan ümumtəhsil məktəblərində təlimin Azərbaycan və rus dillərində aparılması səbəbindəndir. Gələcəkdə SCRATCH proqramının interfeysini Azərbaycan dilinə çevirməyi planlaşdırırıq. Belə olan halda ekranda əməllər Azərbaycan dilində görünəcək.

3.3. Əmrlər (команды)

3.3.1. Motion. Hərəkət qrupu. Группа движение.

Blok (Блок)	Təyinatı (Назначение)
	<p>Spraytı göstərilən rəqəm qədər irəli (+) və ya geriye (-) hərəkət etdirir.</p> <p>Передвигает спрайт на указанное число шагов вперед (+) или назад (-).</p>
	<p>Spraytı saat əqrəbi istiqamətindən döndərir.</p> <p>Поворачивает спрайт по часовой стрелке</p>
	<p>Spraytı saat əqrəbinin əksi istiqamətindən döndərir.</p> <p>Поворачивает спрайт против часовой стрелки</p>
	<p>Göstərilən istiqamətə dönməli (0 - yuxarı, 90 - sağa, 180 - aşağı, -90 - sola, x - ixtiyarı bucaq)</p> <p>Повернуться в указанном направлении (0-вверх, 90-направо, 180-вниз, -90- налево, x-любой угол)</p> 
	<p>Spraytı kursora və ya digər sprayta istiqamətləndirir. Əmr yerinə yetirilən anda həmişə sistemdə reaksiya verilə biləcək mövcud obyektlərin siyahısı görünür. Ən sadə halda heç bir obyekt olmasa, onda siçan qurğusunun göstərdiyi kursora istiqamətləndirir.</p> <p>Направляет спрайт на курсор или на другой спрайт. При выполнении команды всегда существует перечень объектов, которые в данный момент присутствуют в системе и на которые можно реагировать. В самом простом случае, когда других объектов нет, предлагается повернуться в сторону, где находится указатель мышки.</p>
	<p>Spraytı x və y üçün verilmiş koordinata daşıyır.</p> <p>Переносит спрайт в точку с координатами x и y</p>
	<p>Siçanın göstəricisi olan kursurun və ya digər spraytın mövqeyinə adlamaq.</p> <p>Перейти в точку, где расположен указатель мыши или другой спрайт.</p>



Blok (Блок)	Təyinatı (Назначение)												
	<p>Spraytı verilmiş (x,y) koordinatlı nöqtəyə göstərilən saniyə ərzində rəvan daşımalı.</p> <p>Плавно переместить спрайт в точку с указанными координатами (x,y) за указанное время. На перемещение затратится время, указанное в секундах.</p>												
	<p>x oxu üzrə koordinatın yerini verilmiş ədəd qədər dəyişir.</p> <p>Изменить положение по оси x на указанное число шагов.</p>												
	<p>x koordinatına verilmiş qiymət mənimsədir.</p> <p>Координате x спрайта присваивается данное значение</p>												
	<p>y oxu üzrə koordinatın yerini verilmiş ədəd qədər dəyişir.</p> <p>Изменить положение по оси y на указанное число шагов.</p>												
	<p>y koordinatına verilmiş qiymət mənimsədir.</p> <p>Координате y спрайта присваивается данное значение</p>												
	<p>Ekranın kənarında olduqda, geriyə təkan verir. Əgər siz öz qəhrəmanınızı nəzərdən qaçırmaq istəmirsinizsə, onda bu halda faydalı əmrdir.</p> <p>Кənara çatdıqda spraytın hərəkəti əks istiqamətə dəyişir.</p>												
	<p>Fırlanma stilini seçmək.</p> <p>Выбрать стиль вращения.</p> <table border="1" data-bbox="691 1458 1358 1722"> <thead> <tr> <th>Style</th> <th>Stil</th> <th>Стиль</th> </tr> </thead> <tbody> <tr> <td>Left-right</td> <td>Sola - sağa</td> <td>Влево-вправо</td> </tr> <tr> <td>Don't rotate</td> <td>Fırlanmır</td> <td>Не вращать</td> </tr> <tr> <td>All around</td> <td>Daim fırlanır</td> <td>Кругом</td> </tr> </tbody> </table>	Style	Stil	Стиль	Left-right	Sola - sağa	Влево-вправо	Don't rotate	Fırlanmır	Не вращать	All around	Daim fırlanır	Кругом
Style	Stil	Стиль											
Left-right	Sola - sağa	Влево-вправо											
Don't rotate	Fırlanmır	Не вращать											
All around	Daim fırlanır	Кругом											
	<p>Spraytın x koordinatına (-240-dan 240-dək) istinad edir. Digər əmrlərlə birgə tətbiq edilir.</p> <p>Ссылается на координату x спрайта (от -240 до 240) Применяется вместе с другими командами.</p>												
	<p>Spraytın y koordinatına (-180-dan 180-dək) istinad edir. Digər əmrlərlə birgə tətbiq edilir.</p> <p>Ссылается на координату y спрайта (от -180 до</p>												

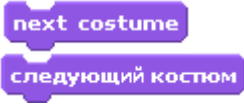



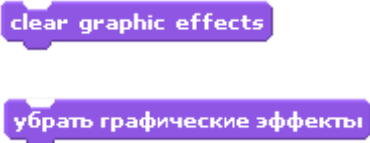

Blok (Блок)	Təyinatı (Назначение)
	180). Применяется вместе с другими командами.
	Spraytın istiqamətinə müraciət edir (0 - uxağı, 90 - sağa, 180 - aşağı, -90 - sola, x - ixtiyarı bucaq) Ссылается на направление спрайта (0-вверх, 90-направо, 180- вниз, -90- налево, x-любой угол)

3.3.2. Looks. Görüntü qrupu. Группа внешность

Blok (Блок)	Təyinatı (Назначение)
	Verilmiş fonu seçməklə səhnəni dəyişir. Меняет вид сцены, выбирая заданный фон.
	Səhnə açılan siyahıdan ya konkret, ya əvvəlki, ya da sonrakı fona keçəcək və gözləyəcək. Сцена сменит фон на или выбранный из выпадающего списка, или предыдущий, или следующий и будет ждать.
	Səhnə üçün siyahıdan növbəti fonu seçir. Sonuncu fondan yenə birinci fona keçir. Yalnız birdən çox fon olduqda işləyir. Выбирает для сцены следующий фон из списка фонов, от последнего фона переходит снова к первому. Имеет смысл только тогда, когда имеется несколько фонов.
	Fonun adı. Damanı qeyd edəndə səhnədə cari fonun adı görünür. Имя фона. При отметки ячейки на сцене появляется имя текущего фона.
	Səhnənin cari fonunun sıra nömrəsini verir. Возвращает порядковый номер текущего фона сцены.
	Verilmiş vaxt ərzində spraytın yanında haşiyədə əmrdə verilmiş mətn çıxır. Рядом со спрайтом в выноске выводится заданный в поле команды текст в течение заданного времени.



Blok (Блок)	Təyinatı (Назначение)
	<p>Spraytın yanında haşiyədə əmrdə verilmiş mətn çıxır. Mətni yox etmək üçün bu əmr boş sahəli (mətnsiz) təkrar edilir. Рядом со спрайтом в выноске выводится заданный в поле команды текст. Для удаления выноски - такая же команда с пустым полем.</p>
	<p>Verilmiş vaxt ərzində spraytın yanında haşiyədə əmrdə verilmiş mətn çıxır və sanki “düşünmə”-ni imitasiya edir. Выводит в выноске рядом со спрайтом заданный текст в течение заданного времени, имитируя обдумывание.</p>
	<p>Spraytın yanında haşiyədə əmrdə verilmiş mətn çıxır və sanki “düşünmə”-ni imitasiya edir. Mətni yox etmək üçün bu əmr boş sahəli (mətnsiz) təkrar edilir. Выводит рядом со спрайтом выноску, имитируя обдумывание. Для удаления выноски - такая же команда с пустым полем.</p>
	<p>Spraytı səhnədə göstərməklə görünən edir. Показывает спрайт на сцене, делает его видимым.</p>
	<p>Spraytı görünməz edir. Делает спрайт невидимым.</p>
	<p>Açılan siyahıdan digər kostyumu seçməklə spraytın xarici görünüşünü dəyişir. Меняет внешний вид спрайта, переходя к другому костюму выбранного из выпадающего списка.</p>
	<p>Açılan siyahıdan digər fonu seçməklə səhnənin cari fonunu dəyişir. Меняет текущий фон сцены, переходя к другому фону выбранного из выпадающего списка.</p>

Blok (Блок)	Təyinatı (Назначение)																								
	<p>Sprayt üçün növbəti kostyumu seçir. Sonuncu kostyumdan birinci kostyuma keçir. Spraytın yalnız birdən çox kostyumu olduqda işləyir. Выбирает для спрайта следующий костюм из списка, от последнего костюма переходит снова к первому. Имеет смысл только тогда, когда у спрайта имеется несколько костюмов.</p>																								
	<p>Cari halda spraytın göründüyü kostyumunun sıra nömrəsini verir. Возвращает порядковый номер костюма, который в данный момент одет спрайт.</p>																								
	<p>Spraytın xarici görünüşü seçilən effektdə və verilən ədədə uyğun olaraq dəyişir. Effekti menyudan seçmək olar. Изменяет внешний вид спрайта согласно выбранного эффекта на указанную величину.</p> <table border="1" data-bbox="769 987 1437 1339"> <thead> <tr> <th>Effect</th> <th>Effekt</th> <th>Эффект</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>rəng</td> <td>Цвет</td> </tr> <tr> <td>fish-eye</td> <td>balıq gözü</td> <td>рыбий глаз</td> </tr> <tr> <td>whirl</td> <td>burulğan</td> <td>завихрение</td> </tr> <tr> <td>pixelate</td> <td>piksel</td> <td>укрупнение</td> </tr> <tr> <td>mosaic</td> <td>mozaika</td> <td>мозаика</td> </tr> <tr> <td>brightness</td> <td>parlaqlıq</td> <td>яркость</td> </tr> <tr> <td>ghost</td> <td>xəyal</td> <td>призрак</td> </tr> </tbody> </table> <p>Эффекты можно выбрать из меню.</p>	Effect	Effekt	Эффект	color	rəng	Цвет	fish-eye	balıq gözü	рыбий глаз	whirl	burulğan	завихрение	pixelate	piksel	укрупнение	mosaic	mozaika	мозаика	brightness	parlaqlıq	яркость	ghost	xəyal	призрак
Effect	Effekt	Эффект																							
color	rəng	Цвет																							
fish-eye	balıq gözü	рыбий глаз																							
whirl	burulğan	завихрение																							
pixelate	piksel	укрупнение																							
mosaic	mozaika	мозаика																							
brightness	parlaqlıq	яркость																							
ghost	xəyal	призрак																							
	<p>Nömrəyə uyğun olaraq spraytın görünüşünün rəngini dəyişir. Əksər effektlərin dəyişmə intervalı -100 və 100 arasında olur. Задает указанное значение эффекта для внешнего вида спрайта. Значения большинства эффектов в интервале от -100 до 100.</p>																								
	<p>Sprayt üzərində aparılmış bütün qrafiki effektlər silinir. Удаляет все графические эффекты спрайта.</p>																								
	<p>Spraytın cari ölçüsünü göstərilən ədədə uyğun %-lə dəyişir. Ədəd müsbətdirsə artır, mənfidirsə azalır.</p>																								



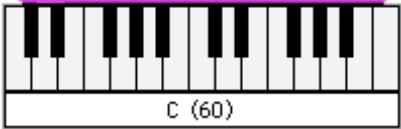
Blok (Блок)	Təyinatı (Назначение)
	İzменяет размер спрайта на заданную величину в %. Увеличивается (если число положительное) или уменьшается (если со знаком "-").
	Spraytın əsl ölçüsünə nəzərən faizlə ölçüsünü dəyişir. İzменяет размер спрайта на заданное значение процента от оригинала.
	Obyekti üst (birinci) qata daşıyır, onun qarşısını heç nə kəsmir. Объект перемещается в первый слой, его ничто не закрывает.
	Obyekt, göstərilən ədədə uyğun sayda, arxa qata aparılır. Переносит спрайт назад на заданное количество слоев.
	Spraytın əsl ölçüsünə nəzərən faizlə cari ölçüsünü verir. Возвращает значение размера спрайта в процентах относительно оригинала

3.3.3. Sound. Səs qrupu. Группа звук

Scratch proqramının müxtəlif obyektlərinə öz musiqi parçalarını əlavə edə bilərsiniz. Beləliklə musiqi duetləri, triolar, kvartetlər, kvintetlər və s. yaratmaq olar.

К разным объектам в Скретче можно добавлять свои музыкальные партии. Таким образом можно собирать музыкальные дуэты, трио, квартеты, квинтеты и т.д.

Blok (Блок)	Təyinatı (Назначение)
	Menyudan seçilmiş səs ifa edilir və həmin andaca növbəti əmrə keçir. Səs davam edir. Начинает играть выбранный из меню звук и сразу переходит к следующей команде, звук продолжается.
	Səsi ifa edir və səs bitdikdən sonra növbəti əmrə keçir. Onu qeyd etmək lazımdır ki, səsləri həm Scratch proqramının kitabxanasından seçmək olar, həm də











Blok (Блок)	Təyinatı (Назначение)
	<p>.wav, .mp3 formatlı öz səs fayllarınızı əlavə etmək olar.</p> <p>Воспроизводит звук и после окончания звука переходит к следующей команде. При этом звук можно выбрать и из библиотеки программы, и можно добавлять свои местные звуки в формате .wav, .mp3.</p>
<p>stop all sounds остановить все звуки</p>	<p>Bütün səsləri kəsir. Прекращает все звуки.</p>
<p>play drum 48 ▼ for 0.2 beats барабану 48 ▼ играть 0.2 тактов</p>	<p>Seçilmiş zərb aləti (belə alətlərin sayı 18-dir) göstərilən sayda takt ifa edir. Выбранный ударный инструмент (количество инструментов 18) играет указанное число тактов.</p>
<p>rest for 0.2 beats оставшиеся 0.2 тактов</p>	<p>Verilmiş taktların sayı qədər pauza. Пауза в течение заданного числа тактов.</p>
<p>play note 60 ▼ for 0.5 beats  ноту 60 ▼ играть 0.5 тактов</p>	<p>Müəyyən notu verilmiş vaxt müddətində (saniyələrlə) ifa edir. Notlar rəqəmlərlə işarə edilib. Hər bir rəqəmə uyğun notun səslənməsi verilir. Играть определенную ноту указанное количество времени (в секундах). Ноты записаны в цифрах, но против каждой цифры стоит ее звучание.</p>
<p>set instrument to 1 ▼ выбрать инструмент 1 ▼</p>	<p>İfa ediləcək alət seçilir. Belə alətlərin sayı 23-dir. Выбрать инструмент, который будет играть. Количество инструментов 23.</p>
<p>change volume by -10 изменить громкость -10</p>	<p>İşarədən asılı olaraq cari səsin gücünü ya azaldır, ya da ucaldır. Увеличивает (если число положительное) или уменьшает (если число отрицательное) текущую громкость.</p>
<p>set volume to 100 % установить громкость 100 %</p>	<p>Səsin gücünü faizə görə təyin edir. Устанавливает громкость в процентах.</p>
<p>volume громкость</p>	<p>Səsin cari gücünü %-lə verir. Возвращает значение текущей громкости звука в %.</p>



Blok (Блок)	Təyinatı (Назначение)
	<p>Səsin cari tempini artır (ədəd müsbət olsa) və ya azaldır (ədəd mənfi olsa).</p> <p>Увеличивает (если число положительное) или уменьшает (если число отрицательное) текущий темп.</p>
	<p>İfa tempini təyin edir (dəqiqədə taktların sayı).</p> <p>Задаёт темп игры (количество тактов в минуту)</p>
	<p>İfa tempinin ədədini verir (dəqiqədə taktların sayı).</p> <p>Возвращает значение темпа игры (количество тактов в минуту)</p>


3.3.4. Pen. Qələm qrupu. Группа перо

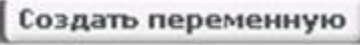

Blok (Блок)	Təyinatı (Назначение)
	<p>Səhnədə şəkil və spraytları silir.</p> <p>Стирает на сцене рисунки и спрайты.</p>
	<p>Spraytı səhnədə həkk edir.</p> <p>Печатает спрайт на сцене.</p>
	<p>Qələmi endirir. Bu əmrdən sonra hərəkət zamanı qələm çəkir.</p> <p>Опускает перо. После этой команды при движении перо рисует.</p>
	<p>Qələmi qaldırır. Bu əmrdən sonra hərəkət zamanı qələm çəkmir.</p> <p>Поднимает перо. После этой команды при передвижении перо не рисует.</p>
	<p>Qələmin rəngini verilmiş qiymətə uyğun addım dəyişir, məsələn 10.</p> <p>Изменяет цвет пера на указанное значение, например на 10.</p>
	<p>Qələmin rəngini dəyişir. Kursoru rəng olan sahənin üzərinə gətirib sol düyməni sıxırıq, barmaq işarəsi görünəndə kursoru ekranın üzərində gəzdirməklə istədimiz rəng seçilir.</p> <p>Изменяет цвет пера. Курсор подводится на поле цвет, левой кнопкой нажимается на это поле, когда появляется палец пливём на экране и выбираем нужный цвет с помощью этого пальца.</p>


 	<p>Qələmin rəngini təyin edir. 100 – ağ, 0 – qara. Устанавливает цвет пера, 100 – белый, 0 – черный.</p>
 	<p>Qələmin kölgəsinin rəngini verilmiş qiymətə uyğun addım dəyişir, məsələn 10. Изменяет цвет тени пера на указанное значение, например на 10.</p>
 	<p>Qələmin kölgəsinin rəngini təyin edir. 100 – ağ, 0 – qara. Устанавливает цвет тени пера, 100 – белый, 0 – черный.</p>
 	<p>Qələmin ölçüsünü verilmiş qiymətə uyğun addım dəyişir, məsələn 1 Изменяет размер пера на указанное значение, например на 1</p>
 	<p>Qələmin ölçüsünü müəyyənləşdirir. Устанавливает размер пера.</p>


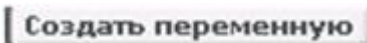
3.3.5. Data. Verilənlər qrupu. Группа данные.

Öncə bu qrupda iki blok olur:  və 



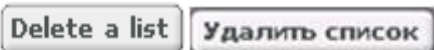

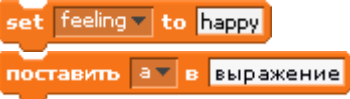




Bir neçə dəyişən yaratmaq olar.  düyməsinə sıxdıqda pəncərə görünür. Həmin pəncərədə dəyişənin adını yığmaq lazımdır. Dəyişəni yaradandan sonra yeni bloklar əmələ gəlir.




Изначально в этой группе лежат 2 блока:  и 

Можно создавать несколько переменных. По щелчку на кнопке  появляется окошко, в котором необходимо напечатать имя переменной. После создания переменной появляются новые блоки.










Блок (Блок)	Тəyinatı (Назначение)
 	Dəyişən yaradır. Dəyişənin adı və tətbiq






Blok (Блок)	Təyinatı (Назначение)
	sahəsi (qlobal və ya lokal) göstərilir. Создает переменную, задается ее имя и область действия (глобальная или локальная).
	Siyahı yaradır. Dəyişənin adı və tətbiq sahəsi (qlobal və ya lokal) göstərilir.
	Dəyişəni silir. Удаляет переменную.
	Siyahını silir. Удаляет список.
	Dəyişənin adı. Dəyişənin adına müraciət edir. Əgər işarələmə qutusu qeyd edilibsə, onda dəyişən qiyməti ilə səhnədə görünəcəkdir. Имя переменной. Ссылается на значение переменной. Если флажок включен, то на сцене отображается монитор со значением переменной.
	Siyahıdan seçilmiş dəyişənə qiymət verir. Присваивает заданное значение выбранной из меню переменной.
	Siyahıdan seçilmiş dəyişənin qiymətini verilmiş kəmiyyət qədər dəyişir. Изменяет значение выбранной из меню переменной на заданную величину.
	Siyahıdan seçilmiş dəyişəni qiyməti ilə birlikdə səhnədə göstərir. Отображает на сцене монитор со значением выбранной из меню переменной.
	Siyahıdan seçilmiş dəyişəni qiyməti ilə birlikdə səhnədə görünməz edir. Прячет (делает невидимым) на сцене монитор со значением выбранной из меню переменной.
	Siyahının adı. Bütün siyahıya müraciət edir. Əgər işarələmə qutusu qeyd edilibsə, onda siyahı səhnədə əks olunur. Имя списка. Ссылается на весь список. Если флажок включен, то список

Blok (Блок)	Təyinatı (Назначение)
	<p>otobrazhaetsya na tsene.</p>
	<p>Verilmiş element siyahının sonuna əlavə edilir. Element ədəd, mətn və ya ifadə ola bilər. Добавляет заданный элемент в конец списка. Элемент может быть числом, любым текстом или задан с помощью выражения.</p>
	<p>Siyahının bir və ya bütün elementləri silinə bilər. Silinəsi elementin sıra nömrəsini (ədəd və ya dəyişən) vermək olar, ya da açılan menyudan uyğun variantı seçmək olar. "last (последний)" sonuncu elementi, "all (все)" bütün elementləri silir. Удаляет один или все элементы из списка. Можно задать порядковый номер удаляемого элемента (число или переменная) или выбрать подходящий вариант в раскрывающемся меню. Вариант "последний" (last) удаляет последний элемент, вариант "все" (all) удаляет все элементы.</p>
	<p>Elementi seçilmiş siyahının verilmiş mövqeyinə (sonuncu, ixtiyari, sıra nömrəsi) əlavə edir. Siyahının ölçüsü 1 vahid artır. "last (последний)" variantı elementi siyahının sonuna, "any (любой)" variantı elementi ixtiyari mövqeyə yerləşdirir. Добавляет элемент в заданную позицию (последний, любой из, порядкового номера) в выбранный список. Длина списка увеличивается на 1. Вариант "последний" (last) добавляет элемент в конец списка, вариант "любой" (any) добавляет элемент в</p>






Blok (Блок)	Təyinatı (Назначение)
	произвольную позицию.
 	<p>Seçilmiş siyahının verilmiş mövqeyində (sonuncu, ixtiyari, sıra nömrəsi) olan elementi dəyişir. "last (последний)" variantı siyahının sonuncu elementini, "any (любой)" variantı ixtiyari mövqedə yerləşən elementi dəyişir. Siyahının ölçüsü dəyişmir.</p> <p>Заменяет элемент в заданной позиции (последний, любой из, порядковый номер) в выбранном списке. Вариант "последний" (last) заменяет элемент в конце списка, вариант "любой" (any) заменяет элемент в произвольной позиции. Длина списка не меняется.</p>
 	<p>Sıra nömrəsi verilmiş və ya açılan menyudan seçilmiş varianta uyğun olan elementə müraciət edir. Digər bloka daxil ola bilər.</p> <p>Ссылается на элемент списка с заданным порядковым номером или с соответствующим вариантом раскрывающегося меню. Может помещаться в другой блок.</p>
 	<p>Siyahıdakı elementlərin sayını göstərir. Показывает количество элементов в списке.</p>
 	<p>Veilmiş qiymət siyahıda varsa, Həqiqi qiymət qaytarır.</p> <p>!!! Böyük və kiçik hərflər fərqlənir. Получает значение Истина, если данное значение содержится в списке.</p> <p>!!! Различаются большие и маленькие буквы.</p>
	<p>Siyahı ekranda görünəcəkdir. Список будет показан на экране.</p>

Blok (Блок)	Təyinatı (Назначение)
	
 	Siyahı görünməz olacaq. Список скроется из вида.



3.3.6. Events. Hadisələr qrupu. Группа события.


Blok (Блок)	Təyinatı (Назначение)
	<p>Yaşıl bayrağa sıxdıqda skriptlər yerinə yetirilir. Запускает скрипт в ответ на щелчок по зеленому флажку.</p>
	<p>Seçilmiş klavişi sıxdıqda əmrlər bloku yerinə yetirilir. İdarəni klavişə verməyə imkan yaradır. Запускает выполнение блока команд в ответ на нажатие выбранной клавиши. Позволяет передать управление на клавиатуру.</p>
	<p>Verilmiş sprayta sıxdıqda skriptin altında yerləşən əmrlər yerinə yetirilir. Запускает выполнение команд расположенного ниже скрипта в ответ на щелчок по заданному спрайту.</p>
	<p>Fondan asılı olaraq skriptləri idarə edən blok. Əgər fon dəyişirsə, blok yerinə yetirilir. Блок управления скриптом в зависимости от фона. Выполняется, если произошло изменение фона.</p>
	<p>Ya səsin səviyyəsindən (loudness), ya taymerin qiymətindən (timer), ya da videonun hərəkətindən (video motion) asılı olaraq skriptləri idarə edən blok. Блок управления скриптом в зависимости от уровня громкости, либо от значения таймера, либо от движения видео.</p>
	<p>Verilmiş məlumatı aldıqda cavab olaraq skriptin altında yerləşən əmrlər yerinə yetirilir. Запускает выполнение команд расположенного ниже скрипта в ответ на получение заданного сообщения.</p>
	<p>Məlumatın (mesajın) ötürülməsi bloku. Bu bloka "when I receive message" bloku cavab verir. Məlumat ötürülən zaman skriptin yerinə</p>

Blok (Блок)	Təyinatı (Назначение)
	<p>yetirilməsi dayandırılmışdır.</p> <p>Блок передачи сообщения, на которое реагирует блок «когда я получу сообщение». При передаче сообщения с помощью этого блока выполнение скрипта не приостанавливается.</p>
 	<p>Məlumatın (mesajın) ötürülməsi bloku. Bu bloka “when I receive message” bloku cavab verir. Məlumat ötürülən zaman skriptin yerinə yetirilməsi dayandırılır. Digər spraytlar “when I receive message” blokunun skriptlərini yerinə yetirib qurtarana qədər gözləmə davam edir. Bundan sonra cari skriptin yerinə yetməsi davam edir.</p> <p>Блок передачи сообщения, на которое реагирует блок «когда я получу сообщение». При передаче сообщения с помощью этого блока выполнение скрипта приостанавливается, и он ждёт, пока другие спрайты выполнят все скрипты блока «когда я получу сообщение» до конца. Затем выполнение текущего скрипта продолжается.</p>



3.3.7. Control. İdarəetmə qrupu. Группа контроль


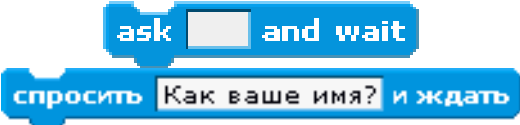
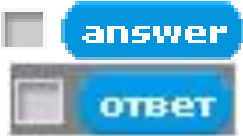
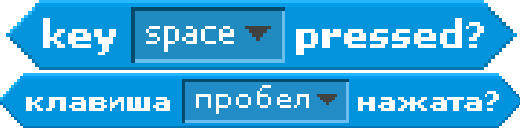
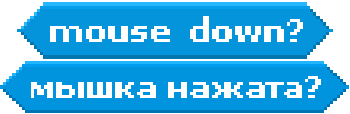

Blok (Блок)	Təyinatı (Назначение)
	<p>Skriptin yerinə yetirilməsi göstərilən vaxt müddətində dayandırılır və sonra yerinə yetirilir.</p> <p>Выполнение скрипта приостанавливается, через указанное время работа продолжается.</p>
	<p>Təkrarla. Daxili bloklar göstərilən sayda yerinə yetirilir.</p> <p>Внутренние блоки выполняются указанное количество раз.</p>
	<p>Nəmişə. Daxili bloklar sonsuz sayda yerinə yetirilir.</p> <p>Внутренние блоки выполняются бесконечно.</p>
	<p>Əgər şərt həqiqidirsə, daxili bloklar yerinə yetirilir, əks halda heç bir fəaliyyət yerinə yetirilmir.</p> <p>Если условие ИСТИННО, то выполняются внутренние блоки, если ложно, то никаких действий не происходит.</p>
	<p>Əgər şərt həqiqidirsə, if daxilindəki bloklar yerinə yetirilir, əks halda isə then daxilindəki bloklar yerinə yetirilir.</p> <p>Если условие ИСТИННО, то выполняются блоки в части если, если ложно, то - в части или.</p>
	<p>Şərt yerinə yetirilincəyə qədər gözləməli.</p> <p>Ждать, пока не выполнится условие.</p>
	<p>Şərt yerinə yetirilməyə qədər daxili bloklar yerinə yetirilir. Başqa sözlə, nə qədər ki, şərt ödənmir, dövr yerinə yetiriləcək, şərt ödənen halda isə dövrün yerinə yetirilməsi dayanacaq və növbəti skript yerinə yetiriləcək.</p> <p>Выполняет внутренние блоки до тех пор, пока заданное условие не будет выполнено. То есть цикл работает до тех пор, пока условие в заголовке ложно, как только оно станет</p>

	<p>истинным, выполнение цикла завершится, и продолжится выполнение скрипта.</p>
 	<p>Seçimə uyğun olaraq skriptin (lərin) yerinə yetirilməsini saxlayır. Прекращает выполнение скрипта (ов) согласно выборки.</p>
 	<p>Spraytın klonları ilə iş bloku. Bu blok spraytın klonu yaranan zaman işləyəcək. Блок управления клонами спрайта. Этот блок выполняется при создании клона спрайта.</p>
 	<p>Klon yaradan blok. Açılan siyahıdan klonu necə yaratmağı (cari spraytın və ya mövcud digər spraytın klonu) seçirik. Блок создания клонов. Из выпадающего списка можно выбрать, клон какого спрайта необходимо создать, если выбрать «себя самого», то будет создан клон текущего спрайта.</p>
 	<p>Blok cari spraytın klonunu silir. Klonların sayı çox olanda proqram işləyərkən ilişə bilər. Блок удаляет клон текущего спрайта. При большом количестве клонов программа может зависнуть.</p>









3.3.8. Sensing. Sensor (hissetmə) qrupu. Группа сенсоров

Blok (Блок)	Təyinatı (Назначение)
 	<p>Əgər sprayt vermiş sprayta, səhnənin kənarına və ya siçanın kursoruna toxunursa, Həqiqi ifadə qaytarır. Obyekt menyudan seçilir.</p> <p>Возвращает значение Истина, если спрайт касается заданного спрайта, края сцены или курсора мыши. Объект выбирается из меню.</p>
 	<p>Əgər sprayt vermiş rəngə toxunursa, Həqiqi ifadə, əks halda Yalan ifadə qaytarır. Obyekt menyudan seçilir. Rəngi seçmək üçün rəng sahəsinə sıxmalı və görünən pipet vasitəsilə spraytda və ya səhnədə olan istədiyiniz rəngin üzərinə sıxmalı.</p> <p>Возвращает значение Истина, если спрайт касается заданного цвета, иначе - Ложь. Для выбора цвета щелкнуть по полю с цветом и, с помощью появившейся пипетки, щелкнуть по нужному цвету на спрайте или на сцене.</p>
 	<p>Əgər seçilmiş birinci rəng ikinci rəngə toxunursa, Həqiqi ifadə qaytarır. Obyekt menyudan seçilir. Rəngi seçmək üçün rəng sahəsinə sıxmalı və görünən pipet vasitəsilə spraytda və ya səhnədə olan istədiyiniz rəngin üzərinə sıxmalı.</p> <p>Возвращает значение Истина, если выбранный первый цвет касается второго цвета на сцене или на другом спрайте. Для выбора цвета щелкнуть по полю с цветом и, с помощью появившейся пипетки, щелкнуть по нужному цвету на спрайте или на сцене.</p>
	<p>Verilmiş sprayta və ya siçanın kursoruna qədər məsafəni bildirir.</p> <p>Возвращает расстояние от данного</p>

Blok (Блок)	Təyinatı (Назначение)
	<p>спрайта или от курсора мыши</p>
	<p>Skriptin yerinə yetirilməsini saxlayır, spraytın sağında haşiyədə verilmiş mətni və səhnənin aşağısında daxil etmə sahəsini göstərir. İstifadəçi cavab yazıb Enter klavişinə və ya <input checked="" type="checkbox"/> düyməsinə sıxdıqda daxil edilmiş məlumat cavab blokunun qiymətinə mənimsədilir və skriptin işi davam edir.</p> <p>Приостанавливает выполнение скрипта, выводит рядом со спрайтом выноску с заданным текстом и в нижней части сцены поле ввода. Когда пользователь введет ответ и нажмет на клавишу Enter или щелкнет по флажку, введенное значение присваивается значению блока ответ и выполнение скрипта продолжается.</p>
	<p>ask (спросить) əmri vasitəsilə daxil edilmiş sonuncu qiyməti özündə saxlayır. Bütün spraytlar üçün ümumidir (qlobal dəyişən kimi).</p> <p>Содержит введенное с помощью команды Спросить последнее значение. Является общим для всех спрайтов (как глобальная переменная)</p>
	<p>Verilmiş klaviş sıxılıbsa, Həqiqi, əks halda Yalan qiymət qaytarır.</p> <p>Возвращает значение Истина, если нажата заданная клавиша, иначе – Ложь.</p>
	<p>Sıçanın sol düyməsi sıxılıbsa, Həqiqi, əks halda Yalan qiymət qaytarır.</p> <p>Возвращает значение Истина, если нажата левая клавиша мыши, иначе – Ложь.</p>
	<p>Sıçanın kursorunun x oxu üzərində qiymətini qaytarır (x koordinatı).</p>



Blok (Блок)	Təyinatı (Назначение)
	Возвращает значение курсора мыши по оси x (координата x).
	Siçanın kursorunun y oxu üzərində qiymətini qaytarır (y koordinati). Возвращает значение курсора мыши по оси y (координата y).
	Mikrofonun səsinin ucılığı (1...100) qaytarır. Возвращает значение громкости (1...100) звука микрофона.
	Blok işləyən proqramın yaddaşından cari spraytda və ya səhnədə video təsvirlər üzərində hərəkətlərin və ya istiqamətlərin sayını qaytarır. Блок возвращает из памяти работающей программы количество движений или направлений над видеоизображением в текущем спрайте, или сцены.
	Video təsvirin çevrilməsi (söndürmə, qoşma və çevirmə) bloku. Блок поворота видеоизображения (выключить, включить и перевернутый).
	Video təsvirin şəffaflığının dəyişilməsi bloku. Блок изменения прозрачности видео изображения.
	Taumerin başlanğıc andan olan vaxtını saniyələrlə qaytarır. Возвращает текущее значение таймера в секундах, начиная с начального момента.
	Taumerin vaxtını sıfır edir. Обнуляет таймер.
	Seçilmiş spraytın seçilmiş parametrini (x və y oxunda vəziyyəti, kostyumun nömrəsi, ölçüsü, həcmi, səsin gücü) qaytarır. Выбранное значение (размер, объем,

Blok (Блок)	Təyinatı (Назначение)
	костюм, звук, положение по оси x или y) у выбранного объекта.
 	<p>Blok kompüterin yaddaşından cari zaman vahidinin qiymətini (il, ay, tarix, həftənin günü, saat, dəqiqə, saniyə) ədəd formatında qaytarır.</p> <p>Блок возвращения из памяти компьютера текущих значений даты и времени (год, месяц, дата, день недели, час, минута, секунда). Все данные возвращаются в числовом формате.</p>
 	<p>Blok kompüterin yaddaşından əsrin əvvəlindən cari ana qədər keçən günlərin sayını qaytarır.</p> <p>Блок возвращения из памяти компьютера количества дней, прошедших сначала века.</p>
 	<p>Blok scratch.mit.edu saytının yaddaşından programı icraya buraxan istifadəçinin adını qaytarır.</p> <p>Блок возвращения из памяти сайта scratch.mit.edu имени пользователя, который запустил программу.</p>



3.3.9. Operators. Operatorlar qrupu. Группа операторы

Hesablama operatorlarını yalnız daxilində sahə olan bloklarda istifadə etmək olar. Onlar ayrılıqda istifadə edilmirlər. Onlar qiymət qaytarırlar.




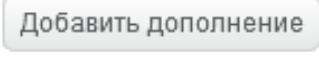
Операторы вычислений можно использовать только внутри строительных блоков. Сами по себе как строительный материал эти кирпичики использовать нельзя. Они возвращают результат.

Blok (Блок)	Təyinatı (Назначение)
	İki ədədin toplanması. Операция сложения двух чисел.
	Çıxma. Вычитание.
	Vurma. Умножение.
	Bölmə. Деление.
	Verilmiş intervalda ixtiyari ədəd verir. Возвращает случайное число в заданном интервале.
	Əgər birinci ədəd ikincidən kiçikdirsə, həqiqi (true) qiymət verir. Возвращает значение Истина , если первое число меньше второго.
	Əgər birinci ədəd ikinciyə bərabərdirsə, həqiqi (true) qiymət verir. Возвращает значение Истина , если первое число равно второму.
	Əgər birinci ədəd ikincidən böyükdürsə, həqiqi (true) qiymət verir. Возвращает значение Истина , если первое число больше второго.
	Hər iki şərt həqiqidirsə, Həqiqi qiymət qaytarır. Возвращает значение Истина , если оба условия истинны.
	Şərtlərdən biri həqiqidirsə, Həqiqi qiymət qaytarır. Возвращает значение Истина , если хотя бы одно условие истинно.
	Şərtin qiyməti Yalandırsa , Həqiqi , Həqiqidirsə Yalan qiyməti qaytarır. Возвращает значение Истина , если значение условия ложно и наоборот.

<p>join Salam dünya</p> <p>слить привет мир</p>	<p>İki mətni və ya ifadəni birləşdirir. Объединяет два текста или выражения.</p>
<p>letter 1 of dünya</p> <p>буква 2 в мир</p>	<p>Verilmiş mətndə və ifadədə göstərilmiş mövqedəki simvolu qaytarır. Возвращает символ в заданной позиции текста или выражения.</p>
<p>length of dünya</p> <p>длина строки мир</p>	<p>Verilmiş mətnin və ya ifadənin uzunluğunu qaytarır. Возвращает длину данного текста или выражения.</p>
<p>mod</p>	<p>Birinci ədədin ikinciyə bölünməsindən alınan qalığı qaytarır. Возвращает остаток от деления первого числа на второе.</p>
<p>round</p> <p>округлить</p>	<p>Ədədi yuvarlaqlaşdırır. Округлить число.</p>
<p>sqrt of 10</p> <p>квадратный корень от 10</p>	<p>Funksiyanın (abs, sqrt, sin, cos, tan, asin, acos, atan, ln, log, e[^], 10[^]) tapılmış qiymətini qaytarır. Возвращает найденное значение функции (abs, sqrt, sin, cos, tan, asin, acos, atan, ln, log, e[^], 10[^]).</p>



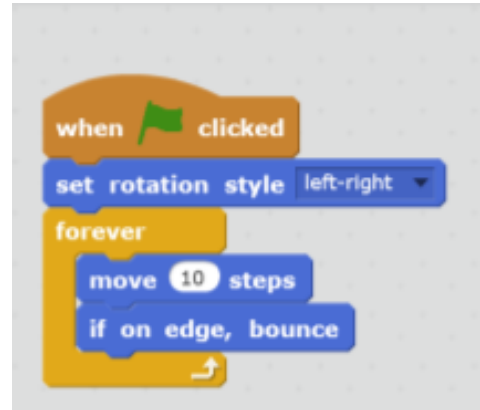
3.3.10. More Bloks. Digər bloklar. Другие блоки.

Blok (Блок)	Təyinatı (Назначение)
 	<p>Blok yaratmaq. Bu halda siz öz blokunuzu yarada bilərsiniz.</p> <p>Создать блок. В этой ситуации вы можете создать свой блок.</p>
 	<p>Artırma əlavə etmək. Bloku istifadə etmək üçün kompüterə PicoBoard və ya LEGO WeDo 1.0 (2.0) qurğuları qoşulmalıdır.</p> <p>Добавить дополнение. Для использования блока компьютер должен быть соединен с устройством PicoBoard или LEGOWeDo 1.0 (2.0)</p>

Cədvəldəki əmrlərin quruluşuna diqqət etsəniz görərsiniz ki, əmrlər əsasən pəzl formasındadır, yuxarı hissədə girinti və aşağı hissədə çıxıntı var. Yəni onları bir-birinə calamaq olar.

Ancaq bəzi qruplardakı bir qisim əmrlərin forması fərqlidir.

Hadisələr (Events) və İdarəetmə (Control) qruplarında bəzi əmrlərin yuxarisında girinti yoxdur. Əvəzində yuxarı hissə yarım dairə formasındadır. Belə əmrlər toplanmış əmrlər blokunda yalnız birinci gələ bilirlər, yəni onlar baş əmrlərdir.



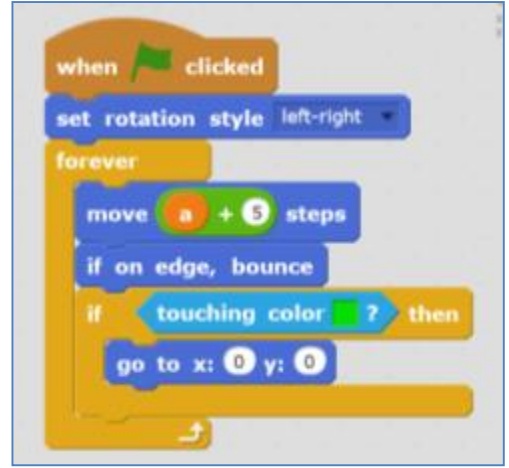
Hərəkət (Motion), Görüntü (Looks), Səs (Sound), Verilənlər (Data), Sensor (Sensing) və Operatorlar (Operators) qruplarında oval formalı əmrlər var.



Bundan başqa Sensor (Sensing) və Operatorlar (Operators) qruplarında uzadılmış altıbucaqlı formalı əmrlər də mövcuddur.

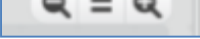


Həm oval, həm də altıbucaqlı əmrlər sərbəst heç bir iş görə bilmirlər. Onlar yalnız digər əmrlərin tərkibində olan uyğun formalı oyuqlara yerləşdiriləndə həmin əmrlə birlikdə iş görürlər. Başqa sözlə, belə əmrlərin köməyi ilə biz dinamik (proqramın işi zamanı qiymətləri dəyişən) proqram qura bilərik. Sağdakı proqram fraqmentində bu aydın görünür.



Scratch proqramının 6 nömrəli sahəsi proqram yazılması üçün işçi sahə adlandırılır. Scratch proqramlaşdırma dilinin 5 sahəsindəki əmrləri tutulub dardılmaqla bu hissədə proqram yazılır, daha doğrusu pəzl formalı əmrlər yığılır.



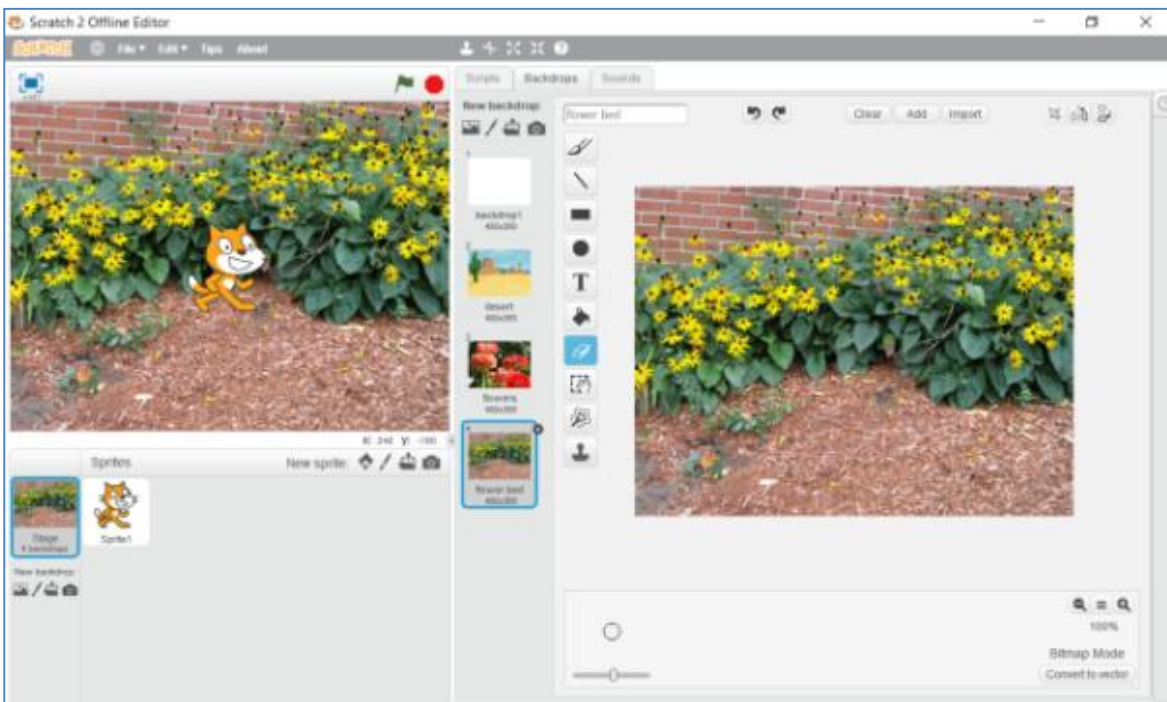
6 nömrəli sahənin aşağı sağ küncündə 3 ədəd  düymələri var. Bu düymələr vasitəsi 6 nömrəli sahədə yazılmış proqramın əmrlərinin ölçülərini kiçiltmək (-), böyütmək (+) və standart ölçüyə gətirmək (=) olar.

Scratch proqramının 7 nömrəli sahəsi obyektlərin (səhnə və spraytların) xassələrini əks etdirmək üçün menyudur. Obyektlərin 3 vacib xassəsi vardır. Bunlar obyekt üçün yazılmış **skriptlər (Scripts)**, **fonlar-kostyumlar (səhnə üçün Backdrops-spraytlar üçün Costumes)** və **səslərdir (Sounds)**.



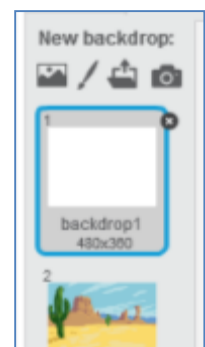
Scripts düyməsini sıxdıqda biz ekranda **5** və **6** sahələrini görürük. Proqram yeni yükləndə də bu rejim işləyir. Skript rejimində səhnə və spraytlar üçün olan skriptlər fərqli ola bilər.

Yəqin ki, teatrda olmuşunuz və orada səhnənin qat-qat sallanmış dekorasiyalardan təşkil olunduğunun şahidisiniz. Tamaşanın müxtəlif hissələrində istifadə edilən arxa dekorasiyalar müxtəlif olur. Dekorasiyaları dəyişməklə tamaşanın görümlüyünü artırır. Burada da səhnə üçün fonlar bir və ya bir neçə ədəd ola bilər. Fonu seçməklə görünüşdə dəyişiklik etmək olar. Cari proqramda hansı fonun (fonların) olmasını öyrənmək üçün **7** nömrəli sahədəki **Backdrops** düyməsini sıxmaq lazımdır. Bu halda ekranda sağ tərəfdə mövcud fonların siyahısı və daxili **Paint** qrafik redaktoru açılacaq. Bundan başqa **4** nömrəli sahədəki **New backdrop** düymələri daxili **Paint** qrafik redaktorundan sol tərəfdə də peyda olacaq. Artıq səhnəyə yeni fon artırmaq iki nöqtədən mümkün olacaq.



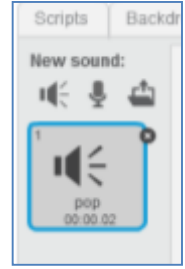
Şəkildən görüldüyü kimi hal-hazırda səhnədə 4 fon var. Ağ rəngli fonu silmək üçün həmin fonu seçməli, bundan sonra onun sağ yuxarı küncündəki **⊗** düyməsinə sıxmaqla həmin fonu silmək olar.

Bundan başqa istədiyim fonu tutub daşımaqla onların ardıcılığını dəyişmək olar. Bu, artıq mövcud fonların ardıcılığını istədiyimiz kimi nizamlamağa imkan verir.



Daxili Paint proqramı istənilən fonda dəyişiklik etməyə imkan verir. Yəni biz istənilən fona hansısa element artırma və azaldıqca bilərik.

7 nömrəli sahədəki **Sounds** (Səslər) düyməsini sıxdıqda səhnə fonuna qoşulmuş səslərin siyahısını görəəcəyik. Adətən yeni boş fayl açanda orada artıq bir səs olur. Həmin səs silmək üçün də fonu silmək qaydasından istifadə edilir. Bunda başqa boş fonun özü də ayrıca səsə malikdir.



Sounds menyusunda 3 düymə var.

1 – düyməsi Scratch proqramlaşdırma dilinin kitabxanasından hazır səslərdən birini yükləməyə imkan verir.

2 – düyməsini sıxdıqda alınan menyuda mikrofon vasitəsilə yeni səs yazmaq olur. Bu menyuda proqramın personajlarını səsləndirməyə imkan verir.

3 – düyməsini sıxdıqda biz proqrama istənilən kitabxanadan kənar səs faylını yükləyə bilərik. Məsələn, Siz Azərbaycan milli rəqs musiqisini yükləməklə proqramda milli rəqs edən personaj yarada bilərsiniz.

2 düyməsini sıxdıqda növbəti şəkildə gördüyümüz menyuda açılacaq:

Bu menyuda 4 düyməsi mikrofonun səs yazmasının səviyyəsini (Microphone volume) tənzimləyir.



3 düyməsi sıxdıqda səs yazmağa başlamaq olur. Bu halda səs yazıldıqca solda səs yazmanın səviyyəsini bildiren indikator görünəcək və düymənin rəngi qırmızı olacaq. Həmin qırmızı düyməni sıxdıqda səs yazma dayanır.

1 düyməsini sıxdıqda yazılan səs oxutmaq olur. 2 düyməsini sıxdıqda səs əvvəlinə keçid olur.

5 və ya 6 düyməsini sıxdıqda uyğun olaraq **Redaktə (Edit)** və ya **Effektlər (Effects)** menyuları açılır.



Redaktə menyusunda aşağıdakı sətirlər var:



undo – ləğv etmə
redo – təkrar
cut – kəsmək
copy – köçürmək
paste – yerinə qoymaq
delete – silmək
select all – hamısını seçmək

Effect menyusunda aşağıdakı sətirlər var:



fade in – böyütmək
fade out – gizlənmək
loude – səsi ucaltmaq
softer – səsi yumşaltmaq (azaltmaq)
silence – səssiz etmək
reverse – revers (tərsinə ifa)

3.4. Səhnənin quruluşu

Scratch proqramlaşdırma dilində səhnə (stage) qat-qat (lay-lay) düzülmüş fonlardan ibarətdir. Bunu adi teatr səhnəsindəki dekorasiyalarla müqayisə etmək olar. Səhnənin arxa dekorasiyasını dəyişməklə tamaşada müxtəlif məkanları göstərmək olar. Eyni vəziyyət Scratch proqramlaşdırma dilində baş verir. Səhnəyə bir neçə fon əlavə edib onların görünüşünü idarə etməklə proqramın işini rəngarəng etmək olar.



Səhnəyə yeni fon əlavə etməyi interfeysin 4 nömrəli sahəsini təsvir edərkən izah etmişdik. Səhnəyə yeni fon əlavə edəndə yeni əlavə edilən fon həmişə ən son qatda yerləşdirilir və buna görə də siyahıda aşağıda olur. Siz fonların qatlarda yerləşmə ardıcılığını dəyişə bilərsiniz. Bunun üçün yerini dəyişmək istədiyiniz fonu tutub daşımaqla qatını dəyişirsiniz.

Bundan başqa, yuxarıda deyildiyi kimi, Scratch proqramının daxili Paint proqramı vasitəsilə istənilən fonu dəyişmək olar. Bu isə Sizə səhnənin tərtibatını zövqünüzə uyğun dəyişdirməyə imkan verir.

3.5. Spraytlarla iş

Scratch proqramlaşdırma dilində əsas rolu *sprayt* adlanan qəhrəmanlar oynayır. Yuxarıda 4 nömrəli sahə haqqında danışarkən **New sprite:**



bölməsi haqqında da danışmışdıq. Bu bölmədə yeni əlavə etmək üçün 4 yol

göstərilir. Bunlar aşağıdakılardır:

1 – Scratch proqramının kitabxanasından hazır sprayt seçmək. Bu düyməni sıxdıqda Scratch proqramının kitabxanasındakı mövcud spraytlar görünəcək. Spraytlar kateqoriyalara, mövzulara və tipə görə qruplaşdırılır. Bu forma axtarışı asanlaşdırır. All rejimini seçəndə bütün spraytlar görünür.

2 – Scratch proqramının daxili Paint altproqramından istifadə etməklə yeni sprayt çəkmək. Daxili Paint proqramı ilə iş sonrakı bölmələrdə təsvir ediləcək.

3 – İxtiyari formatlı şəkil faylı formasında olan spraytı yükləmək.

4 – Kompüterin kamerasından istifadə etməklə yeni sprayt yaratmaq.



Səhnə üçün olduğu kimi spraytlar üçün də 3 növ xassə mövcuddur. Bunlar obyekt üçün yazılmış **skriptlər (Scripts)**, **spraytlar üçün kostyumlar (Costumes)** və **səslərdir (Sounds)**. Bu xassələrə giriş yuxarıda dediyimiz kimi Scratch proqramının **7** nömrəli sahəsindən mövcuddur.

Yuxarıda dediyimiz kimi, Scripts düyməsini sıxdıqda biz ekranda 5 və 6 sahələrini görürük. Proqram yeni yükləndə də bu rejim işləyir.

7 nömrəli sahədən **Costumes (kostyumlar)** rejimini seçdikdə cari spraytın kostyumlarının siyahısı və Scratch proqramının daxili Paint altproqramının interfeysini görərik. Səhnədə olduğu kimi hər bir sprayt da bir neçə qat şəkillərdən ibarət ola bilər. Spraytın belə şəkillərinə kostyum (costume) deyilir. Məsələn, bu kostyumları növbə ilə göstərməyi idarə edə bilsək obyektə hərəkətə gətirə bilərik. Aşağıdakı rəqs edən qızın 4 kostyumu var. Əgər növbə ilə bu 4 kostyumun ekranda görünməsini idarə edə bilsək, onda rəqs edən qızın görüntüsü alınacaq.



Scratch proqramının **7** nömrəli sahəsindən **Sounds (Səslər)** düyməsini sıxdıqda sprayt üçün səslər menyusuna daxil oluruz. Bu menyu Səhnə üçün olan oxşar menyu ilə tam eynidir. Buna görə də bu menyunun əlavə izahına ehtiyac yoxdur.

4. İlk proqramımız

Biz də ilk proqramımızı yazarkən özümüzdə sual vermişdik: **Nədən başlamalı?** Bu sualımıza cavab vermək üçün kitabxanalara, kitab mağazalarına üz tuturduq və istədiyimiz cavabı almaq üçün çox sayda kitablara müraciət edirdik.

İndiki nəsil çox xoşbəxtdir ki, onların informasiya mənbələri çoxdur. Bu mənbələrdən də ən ucuzu və dolğunu **İNTERNET**dir. İndi bir çox suallarınıza qısa zamanda cavab tapa bilərsiniz. Ancaq heyf ki, alqoritmləşdirmə və proqramlaşdırma sahəsində Azərbaycan dilində mənbələr yox dərəcəsindədir.

Biz də bu boşluğu doldurmaq üçün bu vəsaitdə Sizə kömək məqsədilə **Nədən başlamalı** adlı suala az da olsa cavab verməyə çalışdıq.



4.1. Nədən başlamalı

Artıq Scratch proqramlaşdırma dilinin interfeysi, alətləri və əmrləri (skriptləri) haqqında kifayət qədər bilgi əldə etmişik. Bu bilgiler bizə ilk proqramımızı yazmağa imkan verir. Boş səhifə açırıq. Adətən proqramı yeni açanda boş səhifə açılır. Yox əgər Siz proqramda işləməyə başlamısınızsa və yeni səhifə açmaq istəyirsinizsə, onda aşağıdakı əməliyyatlar ardıcılığını yerinə yetirməliyik:


File → New

Əgər Sizin yaddaşda saxlanmamış proqram varsa, onda Scratch həmin proqramı yadda saxlamağı istəyəcək. Əgər bu proqramı yadda saxlamağa ehtiyac varsa, bunu edin. Ehtiyac yoxdursa yadda saxlamaqdan imtina edə bilərsiniz.

Boş səhifənin hansı özəllikləri var:

- Yeni boş səhifənin (başqa sözlə yeni proqramın) adı yoxdur və ona Scratch proqramı **Untitled** adı verir. Biz **3** sahəsindən ona istədiyimiz ad verə bilərik.
- Yeni boş səhifənin fonu ağ rəngdədir.
- Yeni boş səhifənin fonunun səs xassəsinə baxsaq görərik ki, orada artıq “**pop**” adlı səs faylı qoşulmuşdur.
- Yeni boş səhifədə **Sprite1** adlı bir sprayt (pişik) var.
- **Sprite1** adlı spraytın 2 kostyumu mövcuddur.
- **Sprite1** adlı sprayta **meow** adlı səs (pişik səsi) də qoşulub.

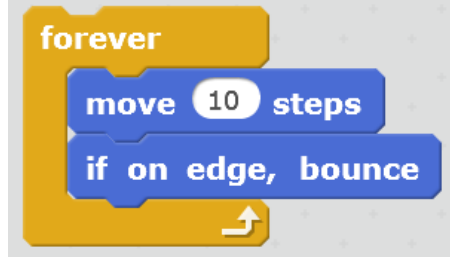
Beləliklə əlimizdə olan materialdan istifadə etməklə öz proqramımızı yazaq. Bizim proqram hansı işi görməlidir. Pişik ekran boyu sola-sağa qaçaraq səs çıxarmalıdır. Spraytı və onun Scripts xassəsinə seçək.

Hərəkət (Motion) qrupundan  əmrini skript sahəsinə daşıyaq. Bundan sonra həmin əmrin üzərinə sıxdıqda Səhnədə görəəcəyik ki, pişik 10 addım (nöqtə) sağa hərəkət etdi. Əlbəttə ki, hər bir dəfə əmrin üzərinə sıxmaqla proqramı yerinə yetirmək olmaz. Buna görə də **İdarəetmə (Control)** qrupundan forever əmrini proqrama əlavə edək. Görünüş belə olacaq:






Proqram parçasının üzərinə sıxdıqda pişik sağa doğru hərəkət edəcək və sağ divara (sərhəddə) çatdıqda orada ilişib qalacaq. Pişiyin daim sərhəddə çatanda divara toxunub geri dönməsini təmin etmək üçün **if on edge, bounce** əmrini dövrün

içərisinə salmaq lazımdır. Bu əmr spraytı sərhəddə çatanda geri dönməyə məcbur edəcək.



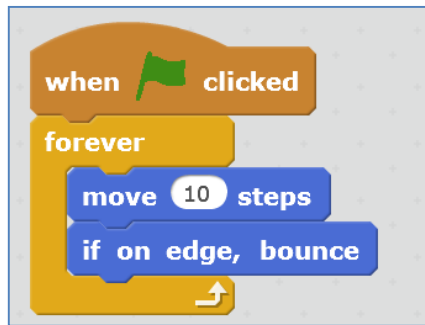
Yenidən bu proqram parçasının üzərinə sıxdıqda pişik sola-sağa doğru hərəkət edəcək.


İndi isə proqramı tam ekranda yerinə yetirməyə çalışaq. Bunun üçün Səhnənin yuxarı sol küncündəki  işarəsini sıxaq. Səhnə tam ekranda görünəndə **5** və **6** sahələri görünməyəcək. Tam ekran rejimində proqramı Səhnənin yuxarı sağ küncündə  qoşa iki düymələri vasitəsilə idarə etmək olar. Yuxarıda dediyimiz kimi, yaşıl bayraq proqramın skriptlərini başlatmaq, qırmızı dairə isə proqramı dayandırmaq üçündür. Proqramda dəyişiklik aparmaq üçün tam ekran rejimindən geri qayıtmaq lazımdır. Bunun üçün Səhnənin yuxarı sol küncündəki  işarəsinə sıxdıqda isə səhnə tam ekran rejimindən adi rejimə qaydır.

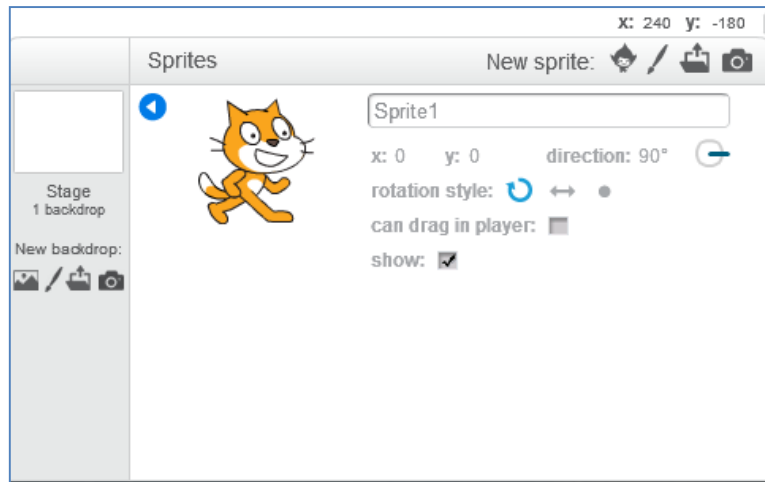
Hadisələr (Events) qrupundan



əmrini proqramın əvvəlinə əlavə edirik.



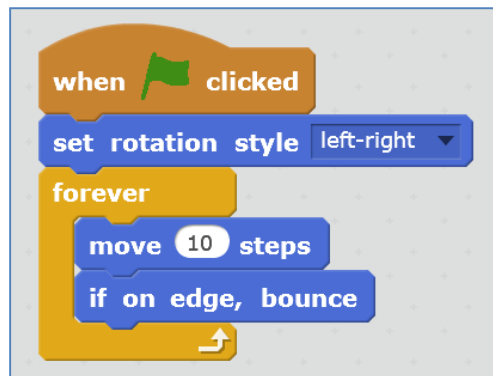
Tam ekran rejiminə keçib proqramı başladırıq. Aha, qəhrəmanımız artıq sola-sağa hərəkət edir. Amma çox vaxt pişik biz istədiyimiz kimi yox, sərhəddə çatdıqda fırlanaraq hərəkət edir. Əlbəttə ki, biz bu halı **4** sahəsində spraytın **info** rejimində sazlaya bilərik. Bunun üçün spraytın nişanının solundakı  düyməsini sıxdıqda ekranda cari sprayt haqqında aşağıdakı məlumat görünəcək.



rotation style: fırlanma stili **can rotate** (fırlana bilər) stilini seçirik. Ancaq yenə də tam ekran rejimində bunu etmək olmur. Scratch proqramlaşdırma dilinin əmrləri bunu proqramın daxilindən idarə etməyə imkan verir. Bunun üçün **Hərəkət**

set rotation style all around ▾

qrupundan əmrini **left-right** (sola-sağa) rejimində proqramın əvvəlinə əlavə etmək lazımdır.



Proqramı başladaq. Artıq pişik istədiyimiz kimi hərəkət edir.

İndi biz pişiyin ayaqlarını tərpedərək qaçmasını təmin etməliyik. Bunun üçün onun **next costume** kostyumlarının növbə ilə görünməsi üçün proqram əlavə etməliyik. Bunun üçün **Görüntü (Looks)** qrupundan əmrini proqrama əlavə edəcəyik.

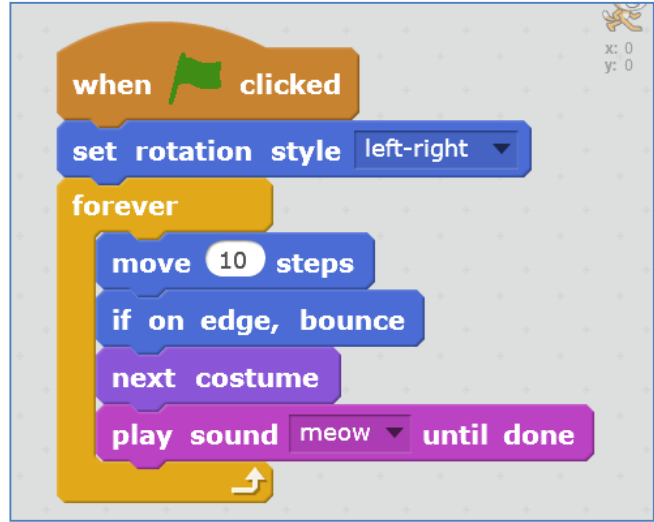
Proqramı başladırıq. Artıq pişik ayaqlarını tərpedərək qaçır.

Çox gözəl. Proqramımızda nə çatmır? Pişik səs çıxartmalıdır (miyoldamalıdır). Yuxarıda dediyimiz kimi, bizim spraytın xassələrinə **meow** adlı səs (pişik səsi) qoşulub. Deməli biz bu səsi səsləndirməliyik. **Səs (Sound)** qrupunda spraytın səsini səsləndirmək üçün 2 əmr var:

play sound pop və **play sound meow until done** əmrləri.

Birinci əmri proqramda dövrün içərisinə salsaq görürük ki, səs tam ifa edilməmiş təkrar-təkrar yenidən başlayır. İkinci əmri proqrama əlavə etdikdə isə pişik ləng hərəkət etməyə başlayacaq. Buna səbəb səs tam səslənməmiş növbəti əmr yerinə yetirilmir. Aşağıdakı proqramı yerinə yetirin və dediklərimizə əmin olun.

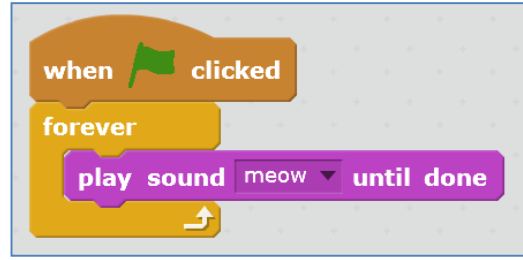
Bəs vəziyyətdən necə çıxmalı.



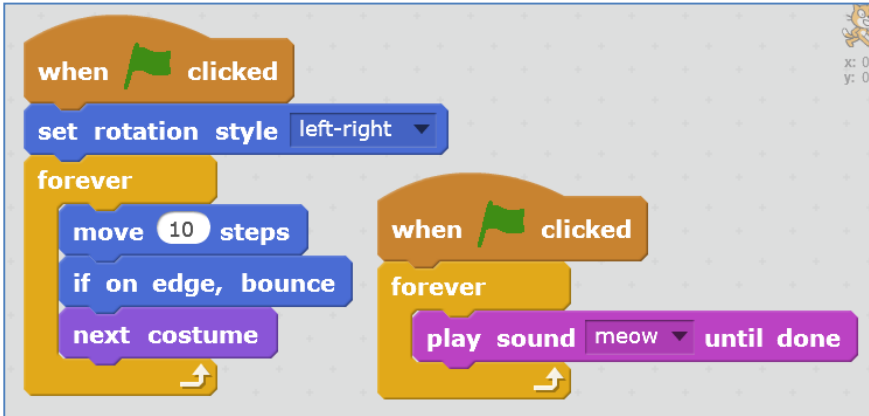
Bildiyimiz kimi insan orqanizmi və onun ayrı-ayrı orqanları bir mərkəzdən (beyindən) idarə edilən və eyni vaxtda işləyə bilən böyük bir sistemdir. Əvvəlcə gəlin həyatda insan orqanizminin necə işləməsini nəzərdən keçirək. Prinsipcə insan hər bir orqanı ayrıca fəaliyyət göstərən bir obyektidir. Bu obyektlər arasında qarşılıqlı bağlılıq da var. Məsələn, insan təkəcə əlini, ya da hansısa barmağını idarə edə bilər. Bu hal bütün orqanlar üçün mövcuddur. Əgər insanın hansısa bir orqanı gözlənilməz vəziyyətlə üzləşsə, bu zaman digər orqanlar da bu vəziyyətə uyğun olaraq fəaliyyətdə olurlar. Belə bir vəziyyəti təsəvvür edək. Azarkeş stadionda qabaq sıralardan birində əyləşərək futbola baxır. Oyunun bir məqamında futbolçulardan birinin güclü zərbəsi nəticəsində top ona sarı gəlir. Bu zaman həmin azarkeşin reaksiyası adətən belə olur: o gözünü yumub əlini üzünə tutaraq yana əyilir və bununla sürətlə gələn topun ona dəyməsindən yayınır. Fikir verirsinizsə, insanın bir neçə orqanı bu vəziyyətdə eyni zamanda, amma hər biri sərbəst hərəkət edir. Bu hərəkətlər ümumi mərkəzdən-beyindən idarə edilir. Sanki hər bir orqan ona verilmiş proqram əsasında işləyir. Əslində insan orqanizmi mükəmməl proqram əsasında işləyən vahid bir sistemdir. Hər bir orqan isə onun “qurğu”larıdır. Təxminən belə qəbul etmək olar ki, insan orqanizminin hər bir orqanı ayrıca proqram əsasında işləyir. Amma bu fərdilik ümumi işə xidmət göstərir. Başqa sözlə, insan orqanizmi paralel hərəkət etməyə qadirdir.



Bizim vəziyyətdə də belədir. Bizim obyekt olan konkret spraytın müxtəlif xassələrini (hərəkətini, kostumları, səsini və s.) ayrıca idarə etməklə ümumi məqsədə nail olmaq olar. Burada da səsi səsləndirmək üçün ayrıca yaşıl bayraqla başladılan proqram bloku qurulmalıdır.

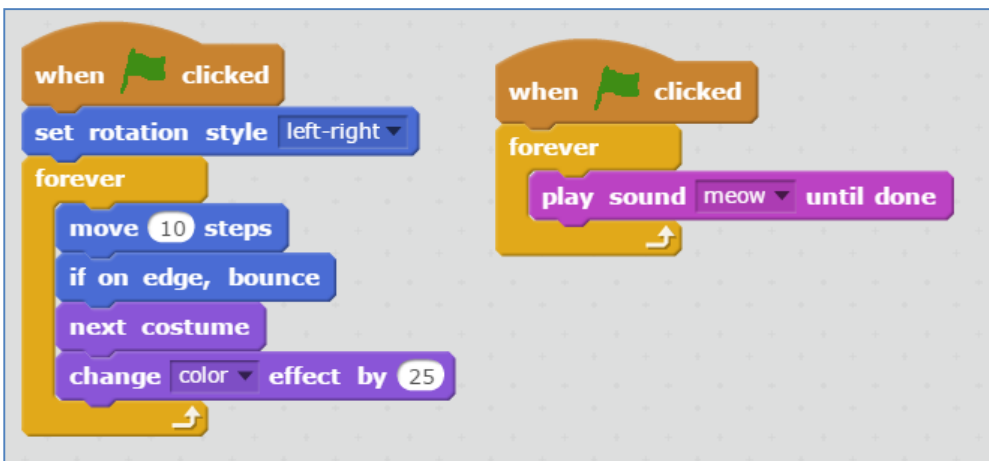


Bu bloku da proqrama əlavə etsək onda tam proqram belə olacaq:



İndi tam ekran rejiminə keçib yaşıl bayraqla proqramı başladın. Əla! Proqram istədiyimiz kim işləyir, pişik ayaqlarını tərpedərək miyoldaya-miyoldaya qaçır.

Əgər pişiyin qaça-qaça rəngini də dəyişməsinə istəyiriksə, onda birinci bloka **Görüntü (Looks)** qrupundan bu əmri parametri ixtiyarı (məsələn, 20 və ya 25) olmaqla əlavə etmək olar.



Proqramı başladın və dediklərimizə əmin olun.

set color effect to 0 əmrinə diqqət etsəniz görərsiniz ki, əmrin **color** yazılan hissəsində alt menyuya giriş var. ▼ düyməsini sıxdıqda effektlərin siyahısı çıxacaq.

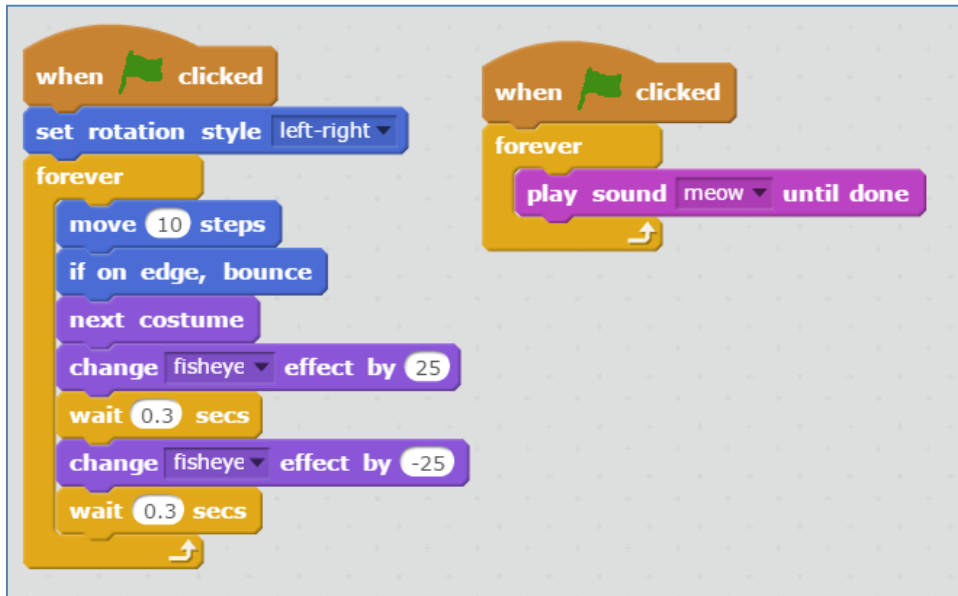
Bunlar **rəng (color)**, **balıq gözü (fisheye)**, **burulğan (whirl)**, **piksel (pixelate)**, **mozaika (mosaic)**, **parlaqlıq (brightness)** və **xəyal (ghost)** effektləridir. 3.2.2. bölməsində də bu haqda qısa məlumat verilib.



Proqramımızda bəzi dəyişiklər edək. **İdarəetmə (Control)** qrupundan proqramımıza **wait 1 secs** **gözlə** əmrini əlavə edək. Bu əmrdəki “1” ədədi neçə saniyə gözləməyi göstərir. Susma halında bu 1 saniyədir. Biz bu gözləməni çox və ya az edə bilərik (2, 3, 4 və sairə). Bizim halda 1 saniyə vaxt versək görərik ki, pişiyin hərəkətində ləngimə baş verir. Bəs görəsən 1 saniyədən az vaxt verə bilərikmi?

Bu vəsaitə əlavədə **Saniyənin hissələri və ya kəsr** adlı bölmədə biz hissə və kəsr haqqında bilgiler vermişik. Həmin bilgiler əsasında biz saniyəni hissələrə bölə bilərik.

Gəlin proqramımıza qayıdaq. Rəngi dəyişmək və gözləmək əmrlərindən istifadə etməklə proqramı aşağıdakı vəziyyətə gətirək.





Görürük ki, proqram istədiyimiz kimi işləməyə başladı. Pişik səs çıxardıqca ağız da tərpənir. Amma...

Amma gözəllik nöqtəyi-nəzərdən qəhrəmanımız ağ yox hansısa maraqlı bir fonda qaçsa daha yaxşı olardı. Bunun üçün **4** sahəsində Səhnəni seçin və səhnəyə hər hansı bir rəngarəng fon əlavə edin. Pişiyin yerini fona uyğun olaraq dəyişin. Bundan sonra proqramı başladın. Əla! Əla! Əla! Bu başqa məsələ. İlk proqramınızın işindən duyduğunuz sevinc hissini biz də duyuruq.



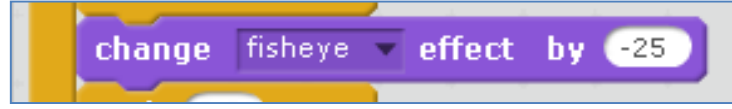
İnamlı ol. Qorxmaq lazım deyil. Heç kim proqramçı doğulmayıb. Sadəcə səndə proqramçı olmaq hissi güclü olmalıdır. Onda hər şeyə nail olacaqsan.

Proqramda istifadə edilən əmrlərdə verilmiş ədədləri, effektləri dəyişin və alınan nəticələri təhlil edin.

Proqramını yaddaşda yadda saxla. Bunu etməsən zəhmətin hədəf gedə bilər.

4.2. Geriyə hərəkət

Proqrama diqqət eləsək görürük ki, orada olan aşağıdakı əmrdə parametri göstərən 25 ədədinin qarşısında “-” işarəsi var.



Yəni yazılış “-25”-dir. Ədədin əvvəlinə “-” simvolu yazmaq nə məqsəd daşıyır?

Gəlin belə bir proqram fragmenti yazaq və yerinə yetirək. Proqramın işinin nəticəsində pişik 20 addım irəli (sağa) gedəcək və 1 saniyə gözlədikdən sonra yenidən 20 addım geriyə gedib ilkin vəziyyətini alacaq.



Bunu belə bir vəziyyətlə müqayisə etmək olar. Əsgərlər sıraya düzülüblər. Komandir onlara əvvəlcə 5 addım irəli addımlamağı əmr edir. Əsgərlər 5 addım irəli gəldikdən sonra komandir indi də onlara 5 addım geriyə addımlamağı əmr edir. Bu dəfə əsgərlər üzvlərini döndərmədən 5 addım geriyə addımlayıb əvvəlki (başladıqları) nöqtəyə qayıdırlar. Scratch proqramlaşdırma dilində geriyə qayıtmaq üçün addımların sayının qarşısında “-” (**minus**) işarəsi qoyulur. Deməli, hər hansı əmrdəki parametrdə ədədin qarşısında minus işarəsi qoyulubsa, onda hərəkət tərsinə, yəni geriyə olacaq.



əmrində də əvvəlki əmrdə 25 dəfə tətbiq edilən balıq gözü effekti bu əmrlə 25 dəfə əks istiqamətdə yerinə yetirilir və nəticədə sanki pişiyin ağzının tərپənməsi effekti alınır.

Qarşısında “**minus**” işarəsi olan ədədə *mənfi ədəd* deyilir.

Bu vəsaitə əlavədə **Mənfi ədədlər** bölməsində aşağıdakı nümunə verilmişdir.

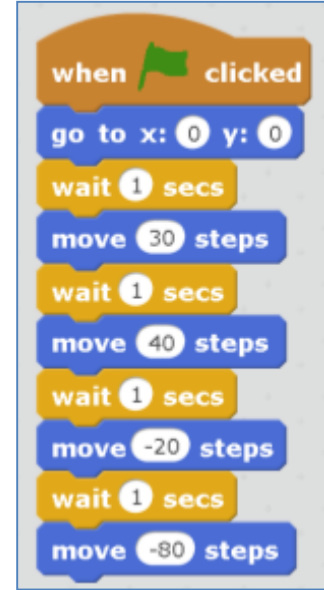
Hesab edək ki, oğlan başlanğıcda “0” nöqtəsində üzünü sağ tərəfə durub. Aşağıda alınan göstəriciləri ədəd oxunda tapın.

- 3 addım irəli gedəndən sonra oğlan 3 nöqtəsində olacaq. $0+30=30$
- 4 addım irəli gedəndən sonra oğlan 7 nöqtəsində olacaq. $30+40=70$
- 2 addım geri gedəndən sonra oğlan -5 nöqtəsində olacaq. $70+(-20)=50$
- 8 addım geri gedəndən sonra oğlan (-3) nöqtəsində olacaq. $50+(-80)=(-30)$



Dediklərimizi Scratch proqramlaşdırma dilinin əmrləri vasitəsilə yazın. Alınan nəticəni gördüyünüz proqramdakı əmrlərlə müqayisə edin.

Proqramdakı ikinci əmr bizi səhnənin mərkəzinə aparır. Bəs həmin əmrdəki x və y hərfləri nəyi göstərir?



Proqrama diqqət eləsək görürük ki, orada olan aşağıdakı əmrdə parametri göstərən



əmr var. Bu əmr spraytı səhnənin mərkəzinə aparır. Əmrdə istifadə edilən x və y işarələmələri nəyi göstərir?

Bu vəsaitə əlavədə **Dekart koordinat sistemi** bölməsində koordinatlar barədə bilgiler vermişik. Üfüqi və şaquli koordinatlarla ifadə edilən koordinat sistemi *dekart koordinat sistemi* adlanır. Əgər diqqətlə baxsaq görürük ki, dekart koordinat sistemi ilə düz səthi (müstəvini) ünvanlamaq olar. Scratch proqramının səhnəsi də bir düz səthdir, yəni müstəvidir.

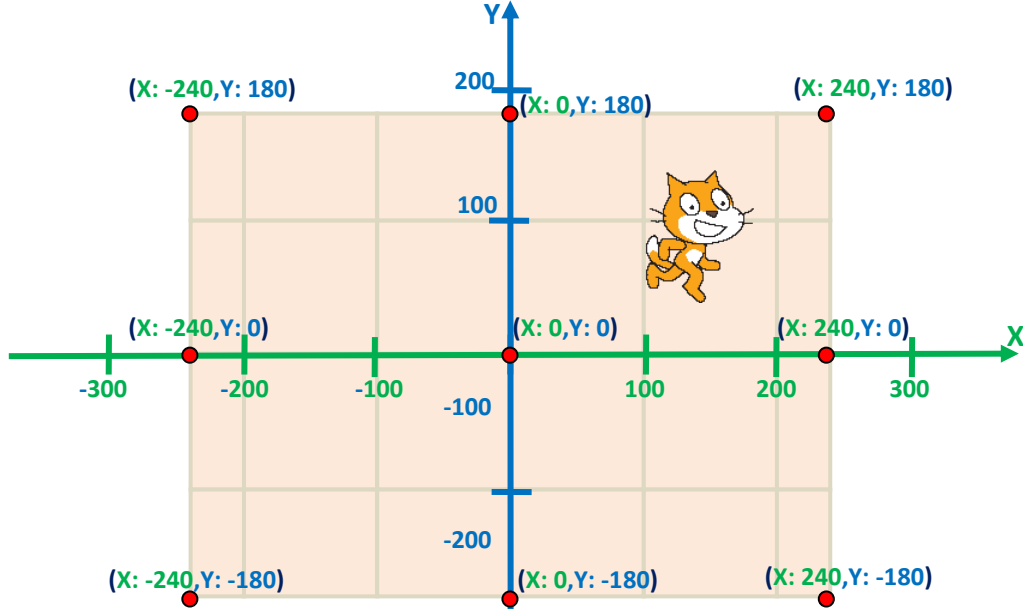
Təkrar da olsa, əlavədəki bilgilerin bəzilərini bir də yada salaq. Dekart koordinat sistemində (bizim səhnədə də) başlanğıc nöqtə mərkəz götürülür.

Aşağıdakı şəkildə səhnənin sxemi verilib. Səhnənin mərkəzində həm üfüqi, həm şaquli koordinatların sıfır başlanğıcı göstərilib. Bu nöqtədən keçən iki (biri üfüqi- X ilə işarə edilir, digəri isə şaquli- Y ilə işarə edilir) ox bir-biri ilə kəsişir. Başlanğıc nöqtənin koordinatı ($X: 0, Y: 0$) ilə işarə edilir.

Biz əvvəlki məşğələlərdə müsbət və mənfi ədədlər barədə danışmışdıq.

Üfüqi X oxunda sıfır nöqtəsindən sağdakı koordinatlar müsbət, soldakılar isə mənfi işarə edilir.

Şaquli **Y** oxunda sıfır nöqtəsindən yuxarıdakı koordinatlar müsbət, aşağıdakılar isə mənfi işarə edilir.




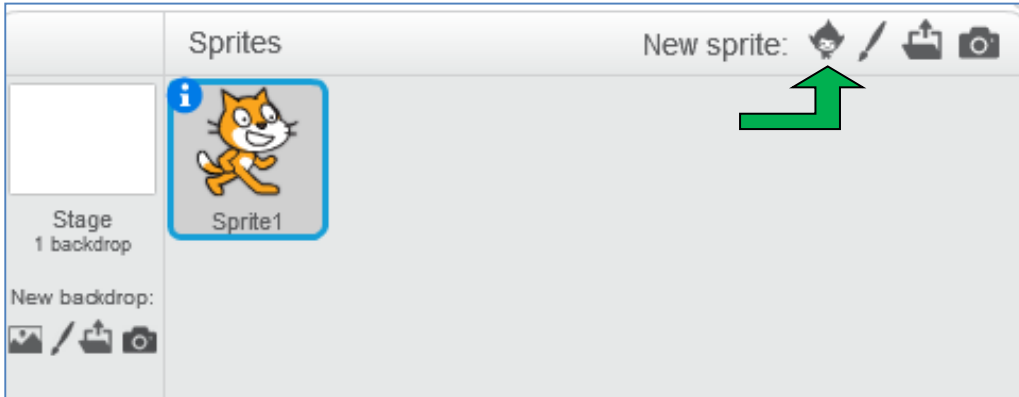
Şəkildə səhnənin bəzi nöqtələrinin koordinatları verilib. Diqqət eləsəniz görərsiniz ki, səhnədə üfüqi 480 (“-240”-la “0” arasında 240 nöqtə və “0”-la “240” arasında 240 nöqtə) koordinat nöqtəsi var. Şaquli isə 360 (“-180”-la “0” arasında 180 nöqtə və “0”-la “180” arasında 180 nöqtə) koordinat nöqtəsi var.



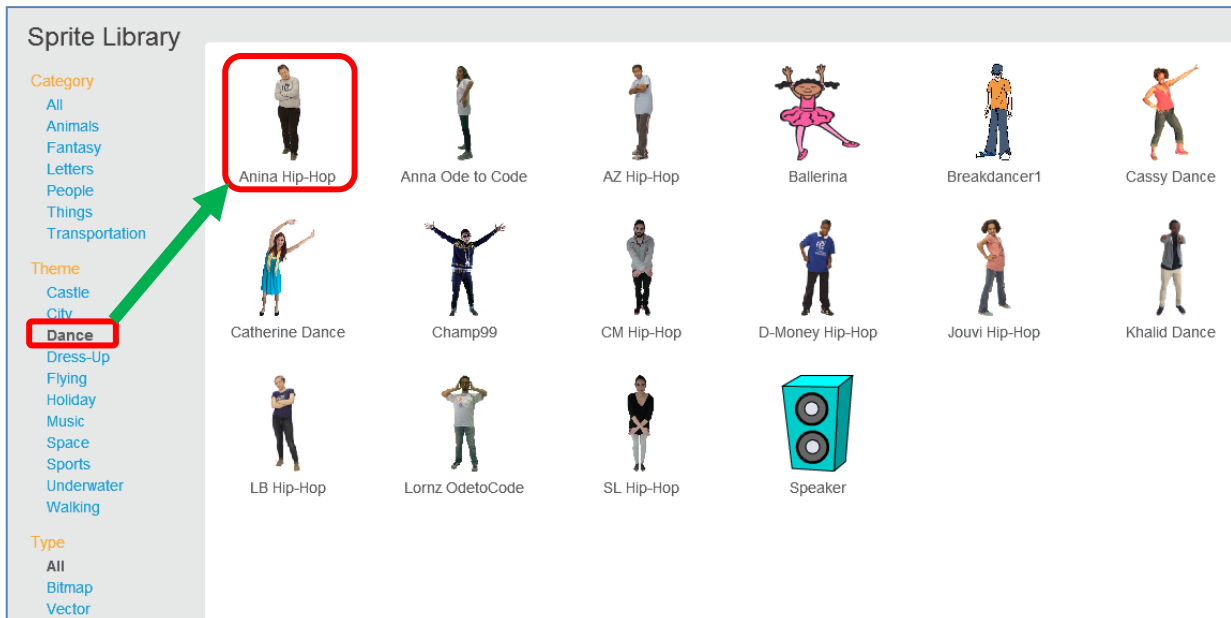
4.3. Rəqs edən uşaq

Gəlin indi maraqlı bir proqram yazaq. Qəhrəmanımız rəqs edəcək.

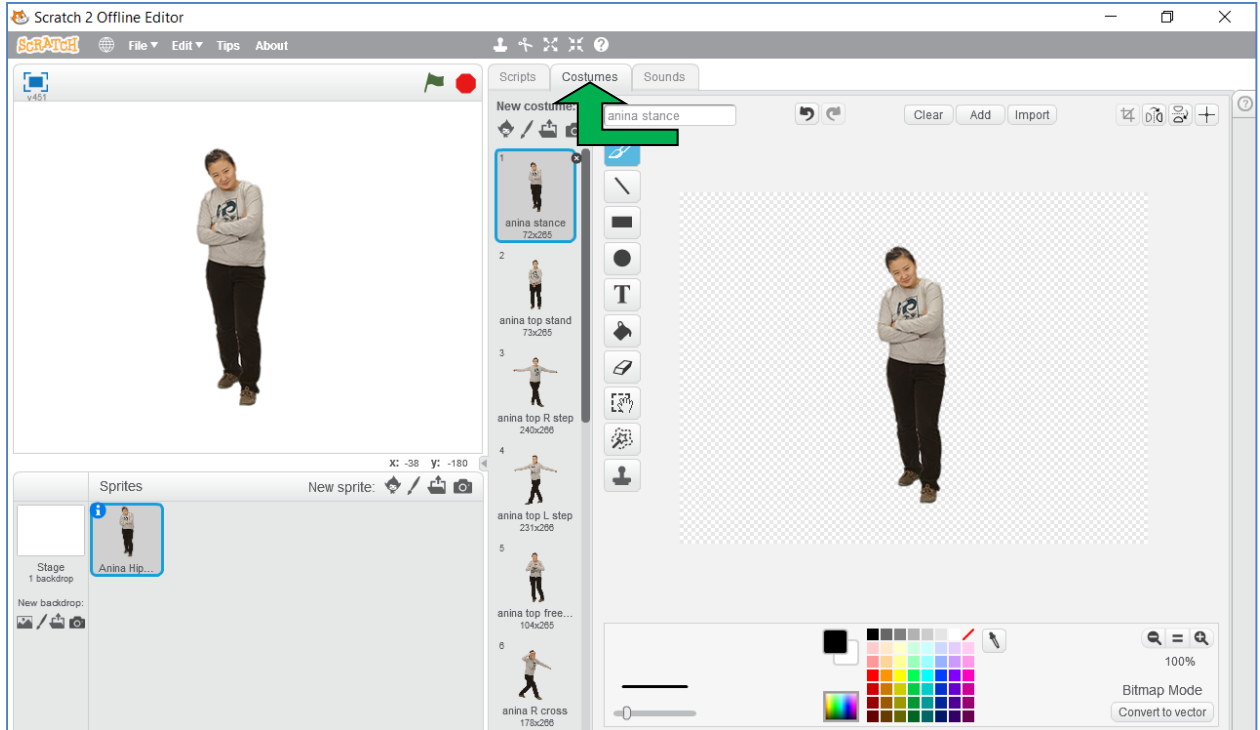
Yeni boş səhifə açırıq. Açılmış səhifədə susma halında olan spraytı (pişiyi) silirik. Bunun üçün əsas interfeysin 4 nömrəli sahəsində spraytın nişanının üzərində klavişin sağ düyməsini sıxıb **Delete** əmrini seçirik. Daha sonra həmin sahədəki **New sprite:** bölməsində  düyməsini sıxırıq.



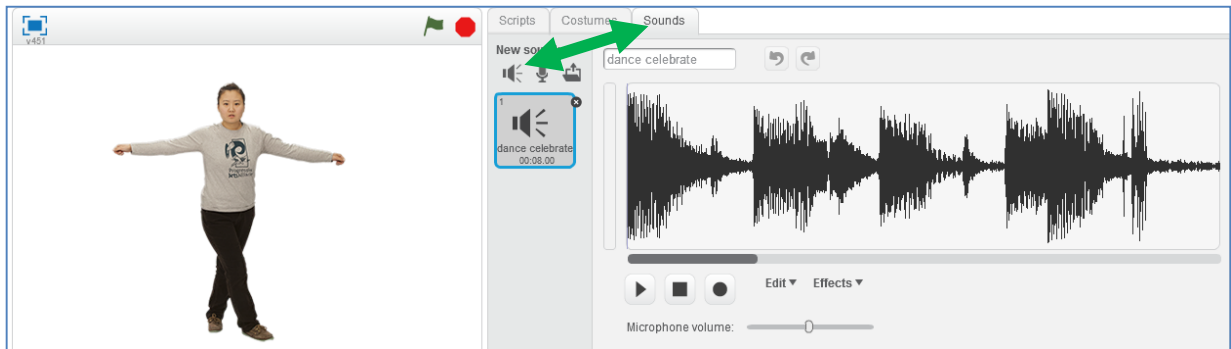
Scratch proqramının kitabxanasından (əgər Sizin öz hazırladığınız şəkil varsa, onu da götürmək olar) yeni sprayt daxil edirik. Məsələn: biz açılan pəncərədə **Dance** qrupundan **Anina Hip-Hop** adlı spraytı seçib proqramımıza personaj kimi daxil edirik.



Əgər əsas interfeysin 7 nömrəli sahəsində **Costumes** düyməsini sıxsaq görərik ki, bizim spraytımızın müxtəlif vəziyyətlərdə 13 ədəd kostyumu var. Əgər biz bu kostyumları növbə ilə səhnədə göstərsək hərəkət alınar.



Eyni sahədə **Sounds** düyməsini sıxsaq görərik ki, bizim spraytımıza səs də əlavə edilib. Əgər spraytımızın səsi yoxdursa, onda biz ona səs əlavə edə bilərik.



Gəlin əsas interfeysin 7 nömrəli sahəsində **Scripts** düyməsini sıxaq və personajımız üçün hərəkət effekti verən sağdakı şəkildə gördüyümüz əmrləri yazaq.

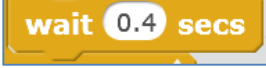
Yaşıl bayraq düyməsini sıxıb programı başlatsaq görərik ki, personajımız hərəkət edir.





Amma hərəkət sürəti çox yüksəkdir. Biz hərəkətin sürətini aşağı salmalıyıq. Bunun üçün proqrama daha bir əlavə edək. Bu  əmridir.

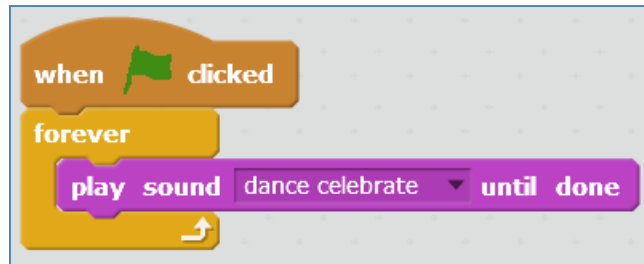


Yaşıl bayraq düyməsini sıxıb proqramı yenidən başlatsaq görərik ki, personajımız artıq normal sürətlə hərəkət edir. Seçəcəyimiz musiqinin tempindən asılı olaraq bu fasilənin müddətini azalda bilərik. Məsələn, biz sprayta əlavə edilmiş musiqinin tempinə uyğun olaraq əmrin parametrini **0.4** götürürük: .



Yaşıl bayraq düyməsini sıxıb proqramı başladaq.

İndi isə musiqini səsləndirək. Əvvəllər qaçan pişiyin səs çıxarması üçün yazdığımız proqramda demişdik ki, eyni bir personaj üçün eyni vaxtda həm hərəkət, həm də səs veririksə, onda onların hər birini ayrıca yaşıl bayraqla idarə etmək lazımdır. Burada da səsi ayrıca əmrlər bloku ilə vermək lazımdır. Aşağıdakı şəkildə həmin əmrlər blokunu görürsünüz:



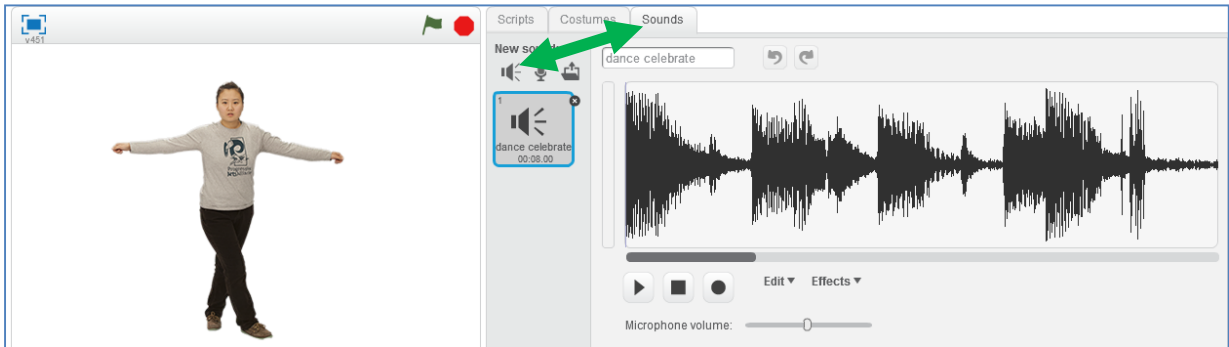
Ümumi proqram aşağıdakı kimi olacaq:




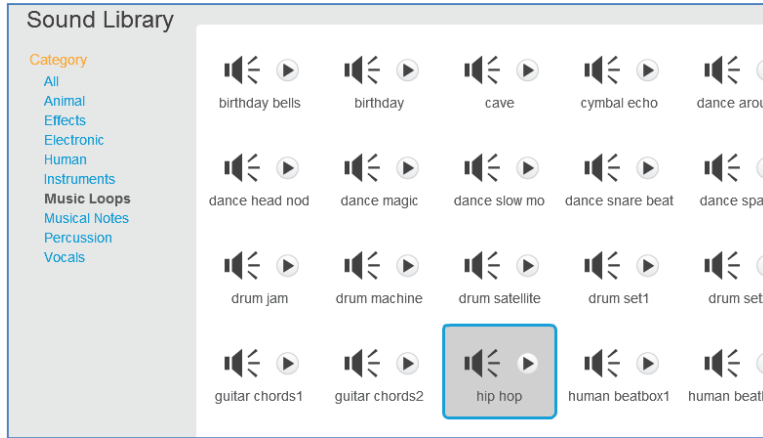
Gəlin yaşıl bayraq düyməsini sıxaq və proqramı başladaq. Aha, artıq uşaq musiqi sədaları altında rəqs edir.

Bəs görəsən rəqsin musiqisini dəyişmək olarmı?

Bunun üçün əsas interfeysin **7** nömrəli sahəsində **Sounds** düyməsini sıxaq və bizim spraytımıza başqa bir səs əlavə edək.

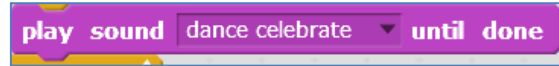


Açılan menyuda  düyməsini sıxanda yeni bir pəncərə açılacaq.

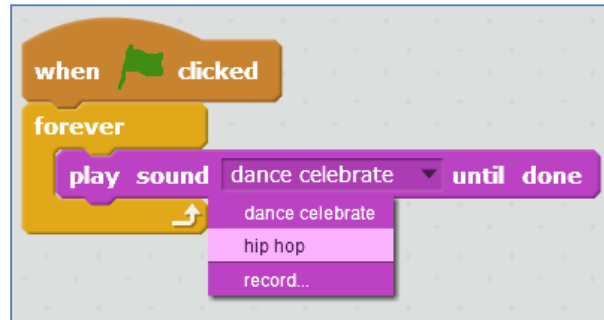


Bu pəncərədə **Music Loops** qrupundan **hip hop** musiqisini götürək və aşağı sağ küncdəki **OK** düyməsini sıxaq. Daha sonra əsas interfeysin **7** nömrəli sahəsində

Scripts düyməsini sıxaq və proqramda



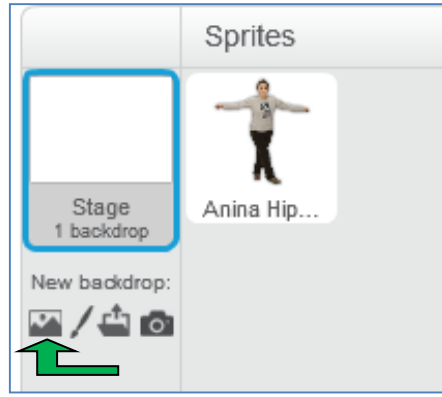
əmrinin parametrlər sahəsindəki ▼ düyməsini sıxıb açılan siyahıdan **hip hop** musiqisini seçək.



Yaşıl bayraqla başlat düyməsini sıxıb proqramı başladaq. Görəcəyik ki, uşaq artıq yeni musiqi sədaları altında rəqs edir. İstədiyimizə nail olduq.

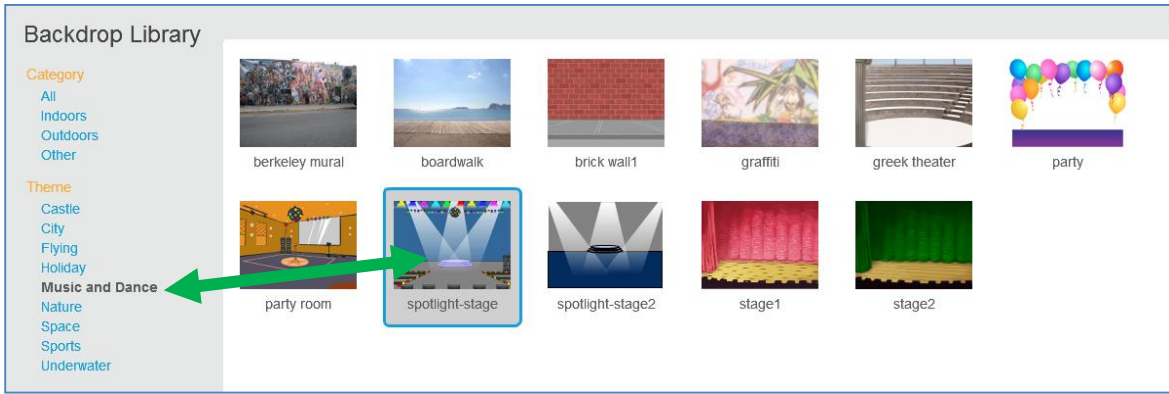
Daha nəyimiz çatmır? Əgər uşaq musiqi sədaları altında rəqs edirsə, bəs niyə bunu diskoteka şəraitində etməsin? Bunun üçün gəlin fonu dəyişək.

Əsas interfeysin **4** nömrəli sahəsində **Səhnəni (Stage)** seçirik. Daha sonra həmin sahədəki **New backdrop:** bölməsində  düyməsini sıxırıq.

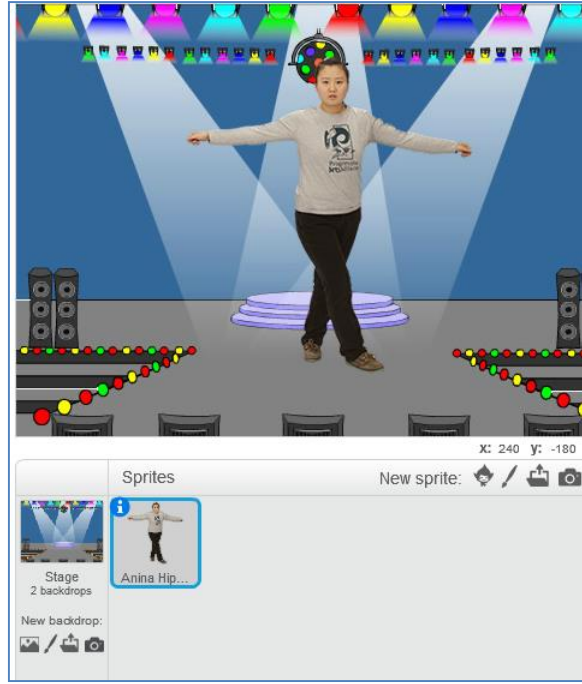



Əsas interfeysin **4** nömrəli sahəsində **Səhnəni (Stage)** seçirik. Daha sonra həmin sahədəki **New backdrop:** bölməsində  düyməsini sıxırıq.

Yeni açılan **Backdrop Library** pəncərəsində **Music and Dance** qrupunda **spotlight-stage** (və ya **spotlight-stage2**) fonunu seçib aşağı sağ küncdəki **OK** düyməsini sıxırıq. Əsas interfeysin **7** nömrəli sahəsində **Scripts** düyməsini sıxaq.




İndi uşaq gözəl bir diskoteka zalındadır. Amma bir nüans işləri korlayır. Uşağın ölçüsü diskoteka zalına nəzərən çox böyükdür.



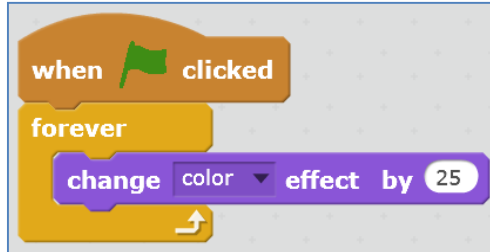
Bu vəziyyəti düzəltmək üçün uşağın ölçüsünü kiçiltmək lazımdır. Əsas interfeysin **4** nömrəli sahəsində **Spraytı (Anina Hip...)** seçirik. Daha sonra əsas interfeysin **2** nömrəli sahəsində  düymələrindən istədiyimizi seçərək **Səhnədə** personajın üzərinə sıxırıq və ölçünü fonun ölçüsünə uyğunlaşdırırıq. Ölçünü seçdikdən sonra personajın yerini dəyişib onu səhnədə yerləşdiririk. Növbəti şəkildə Siz artıq personajın optimal ölçüsünü və yerini görürsünüz.



İndi Səhnənin  düyməsini sıxıb tam ekran rejiminə keçin. Yaşıl bayraq düyməsini sıxıb proqramı başladın. **ƏLA! ALINDI!**

Əgər Sizin kompüterinizin parametrləri güclüdürsə, daha bir əlavə də edə bilərsiniz.

Əsas interfeysin **4** nömrəli sahəsində **Səhnəni (Stage)** seçin. Daha sonra Səhnə üçün aşağıdakı əmrlərdən ibarət blok hazırlayırıq.



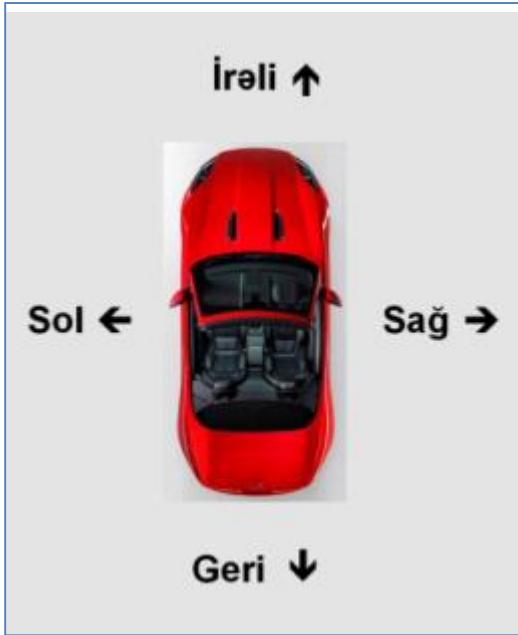
Yaşıl bayraq düyməsini sıxıb proqramı başladın. **Gözəl. Diskoteka zalının işıqları da yanıb - sönməyə başlayır!**

Biz müəlliflər ilk dəfə bu proqramı yazarkən keçirdiyimiz hissləri artıq Siz də keçirirsiniz. Bu gözəl anları gələcəkdə yenə də yaşamanızı arzulayaraq bu proqramla işimizi tamamlayırıq və işimizi davam edirik.

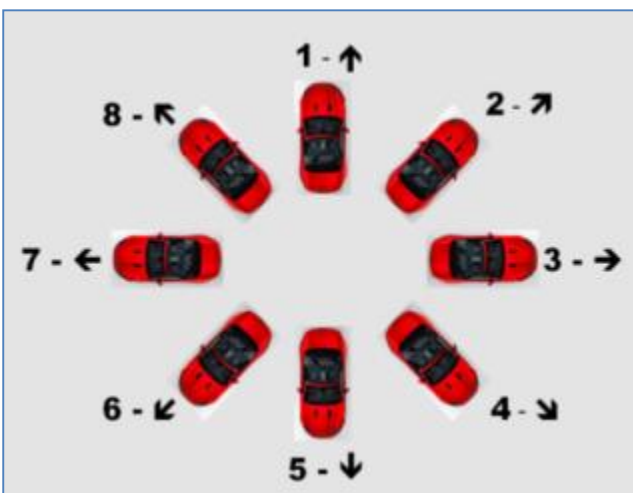


4.4. İstiqamət

Bütün proqramlaşdırma dillərində olduğu kimi Scratch proqramlaşdırma dilində də istiqamətlərlə iş zamanı bəzi detallara fikir vermək lazımdır. Bildiyimiz kimi həyatda insanın harada necə durmasından asılı olmayaraq dörd istiqamət (*irəli, geri, sol və sağ*) əsas götürülür. Şəkildə avtomobil nümunəsində avtomobillə nəzərən bu əsas istiqamətlər göstərilib. Avtomobilin duruşundan asılı olmayaraq avtomobillə nəzərən bu istiqamətlər dəyişmir.



Soldakı şəkildə avtomobilin 8 vəziyyətdə duruşu verilib. Avtomobilin duruşu dəyişsə də avtomobillə nəzərə dörd istiqamət olduğu kimi qalır.



Məsələn, 1 vəziyyətində olan avtomobil müəyyən qədər fırlanıb tutaq ki, 2 vəziyyətini almışsa, onda avtomobillə nəzərən istiqamətlər də eyni qədər dəyişəcək və avtomobildə olan şəxslər üçün bu istiqamətlər (*irəli, geri, sol və sağ*) dəyişməyəcək.

Aşağıdakı cədvəldə avtomobilin 8 vəziyyəti üçün kənarından baxanlar üçün istiqamətlərin yerdəyişməsi göstərilib.

Nö	İrəli	Geri	Sol	Sağ
1	↑	↓	←	→
2	↗	↖	↙	↘
3	→	←	↑	↓
4	↘	↖	↗	↙
5	↓	↑	→	←
6	↙	↗	↘	↖
7	←	→	↓	↓
8	↖	↘	↙	↗

Özünüz şəkillə cədvəldə verilənləri araşdırın və nəticə çıxarın.

İndi isə dediklərimizə konkret proqram nümunələrində baxaq.

Scratch proqramlaşdırma dilində istiqamətləri müəyyən etmək üçün 2 yol var. Birinci yol **4** sahəsindən seçdiyimiz sprayt haqqında informasiya bölməsinə daxil olub spraytın istiqamətini dəyişməkdir. Bu yol proqramın işçi vəziyyətində əl çatandır. Amma proqramı tam ekranda başladanda bu yolu seçmək olmur. Buna görə də ikinci yoldan, proqram daxilində əmrlərdən istifadə etmək daha səmərəlidir. Proqramda belə əmrlərdən biri **Göstərilən istiqamətə dönməli (point in direction)** əmridir. Bu əmrdə parametrin **4** standart qiyməti açılan siyahıda verilib. Bunlar



- **0** - yuxarı,
- **90** - sağa,
- **180** - aşağı,
- **-90** – sola.

Amma biz bucağı ixtiyari də götürə bilərik. Necə ki, bizim misalda biz özümüz həmin sahəyə **45** ədədini yazmışıq. Bu əmr dəqiq istədiyimiz istiqaməti verir.

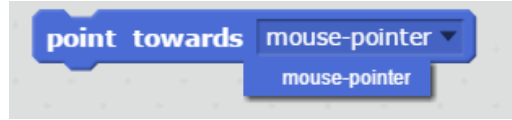
Növbəti əmrlər spraytı istədiyimiz bucaq qədər saat əqrəbi və əksi istiqamətində döndərir. Bu əmrlər əvvəlki istiqaməti verilən bucaq qədər dəyişir. Məsələn, əgər biz sağa **15** dərəcə dönmək əmri vermişiksə və bu əmrə qədər bucaq **90** idisə, onda bu əmrdən sonra bucaq **105** dərəcə olacaq.



Burada bir şeyi də demək yerinə düşər. Əgər biz “**sağa 20 dərəcə dönməli**” əmrinin əvəzinə “**sola -20 dərəcə dönməli**” əmrini yazsaq eyni effekti alacağıq. Yəni mənfi ədəd əks istiqamət deməkdir.

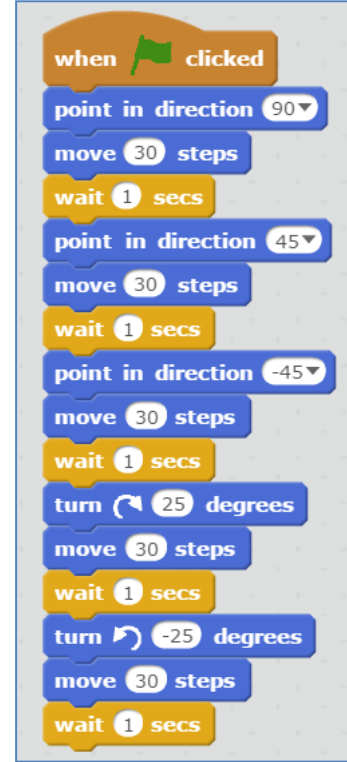


Başqa bir əmr spraytı kursora və ya digər sprayta istiqamətləndirir. Əmr yerinə yetirilən anda həmişə sistemdə reaksiya verilə biləcək mövcud obyektlərin siyahısı görünür. Ən sadə halda heç bir obyekt olmasa, onda siçan qurğusunun göstərdiyi kursora istiqamətləndirir.



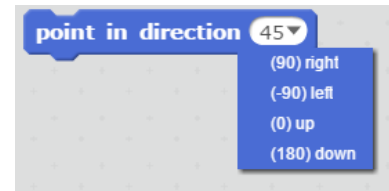
if on edge, bounce

Ekranın kənarında olduqda, geriye təkən verir. Əgər siz öz qəhrəmanınızı nəzərdən qaçıрмаq istəmirsinizsə, onda bu halda faydalı əmrdir. Kənara çatdıqda spraytın hərəkəti əks istiqamətə dəyişir. Sağdakı növbəti proqramda dediklərimizi əyani görə bilərik.



Scratch proqramlaşdırma dilində proqram daxilində əmrlərdən istifadə etməklə istiqamətləri müəyyən etmək olar (daha səmərəlidir). **Göstərilən istiqamətə dönməli (point in direction)** əmrində parametrin 4 standart qiyməti açılan siyahıda verilib. Bunlar

- 0° - yuxarı,
- 90° - sağa,
- 180° - aşağı,
- -90° – sola.



Bu vəsaitin əlavəsində **Bucaqlar** bölməsində biz bucağın olmasını izah etmişik. Dediklərimizi başa düşmək üçün həmin bilgilərə təkrar müraciət etmək məsləhətdir.

Yuxarıda verilmiş nümunələrdən fərqli olaraq biz bucağı ixtiyari də götürə bilərik. Məsələn, 45°. Bu əmr dəqiq istədiyimiz istiqaməti verir.

Əmrlərdə bucağı müsbət göstərəndə saat əqrəbi istiqamətində, mənfi görəndə isə saat əqrəbinin əksi istiqamətində dəyişmə baş verəcək.

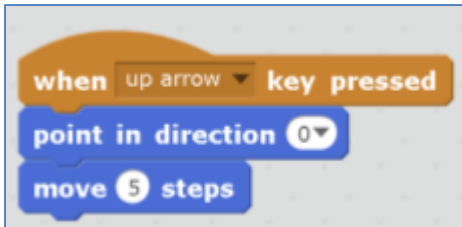
4.5. Klavişlərlə idarəetmə

Bir çox proqramlarda, xüsusilə oyunlarda obyektləri klavişlərlə (ən çox istifadə edilən klavişlər yuxarı, aşağı, sola və sağa ox klavişləridir) idarə edirlər. Məsələn, buna nümunə olaraq labirint oyununu göstərə bilərik. Bu məqsəd üçün Scratch proqramlaşdırma dilində də uyğun əmr mövcuddur.

Tutaq ki, biz yuxarı ox klavişini sıxdıqda sprayt istiqamətini yuxarı götürsün və 5 addım irəli addımlasın.

Hadisələr qrupunda **Probel klavişi sıxılmışsa** əmrinin parametrlər hissəsinə sıxdıqda seçim menyusu alınır. Menyuda probel klavişini, 4 istiqamət oxlarını, ixtiyari (any) klavişi, latın əlifbasının 26 hərfini və 0-dan 9-a qədər rəqəmləri seçmək olar.

Bizə **yuxarı ox (up arrow)** klavişini seçmək lazımdır. Daha sonra istiqaməti yuxarı (0 dərəcə istiqamət) yönəldən və 5 addım irəli gedən əmrlərdən istifadə edəcəyik. Proqram fraqmenti belə olacaq.



İndi isə yuxarı ox klavişini sıxın və spraytın istiqamətini yuxarı dəyişib 5 addım irəli getməsinə əmin olun.

Qeyd: Spraytın ölçüsü səhnəyə nəzərən nisbətən böyük olduğundan onun ölçüsünü kiçiltmək məsləhətdir.

Eyni qayda ilə digər oxlar üçün də proqram fraqmentlərini qurmaq olar. Prinsipcə digər oxlar üçün də eyni əmrlərdən istifadə ediləcək. Sadəcə olaraq seçim menyusundan bizə lazım olan parametrləri seçməliyik. Belə hallarda proqram yazma müddətini qısaltmaq üçün artıq yazılmış proqram fraqmentinin dublikatını çıxarıb yeni proqram fraqmentində parametrləri dəyişmək lazımdır.



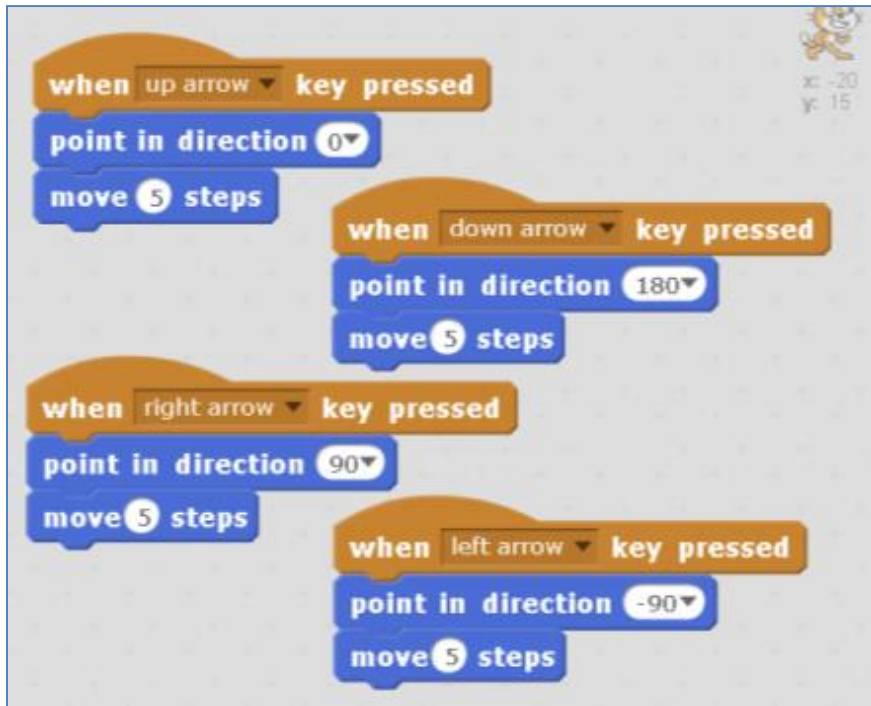
Obyektin dublikatını iki variantda çıxartmaq olar:

- Birinci variantda dublikatını yaratmaq üçün seçdiyimiz proqram fraqmentinin birinci əmrinin üzərində siçanın sağ düyməsini sıxıb **dublikat (dubliate)** sətirini seçməli.
- İkinci variantda Scratch proqramının **2** sahəsində dublikat düyməsini (şəkildə mavi rəngdə verilib) seçib dublikatını yaratmaq üçün seçdiyimiz proqram fraqmentinin birinci əmrinin üzərinə sıxmaq lazımdır.



Burada bir hala da diqqət yetirmək lazımdır. **Probel klavişi sıxılmışsa** əmrini yerinə yetirmək üçün proqramı yaşıl bayraq düyməsi ilə başlatmağa ehtiyac yoxdur. Bu əmr özü sərbəst işləyir. Bu əmrin formasının yuxarıdan yumru olması da bunu bildirir.

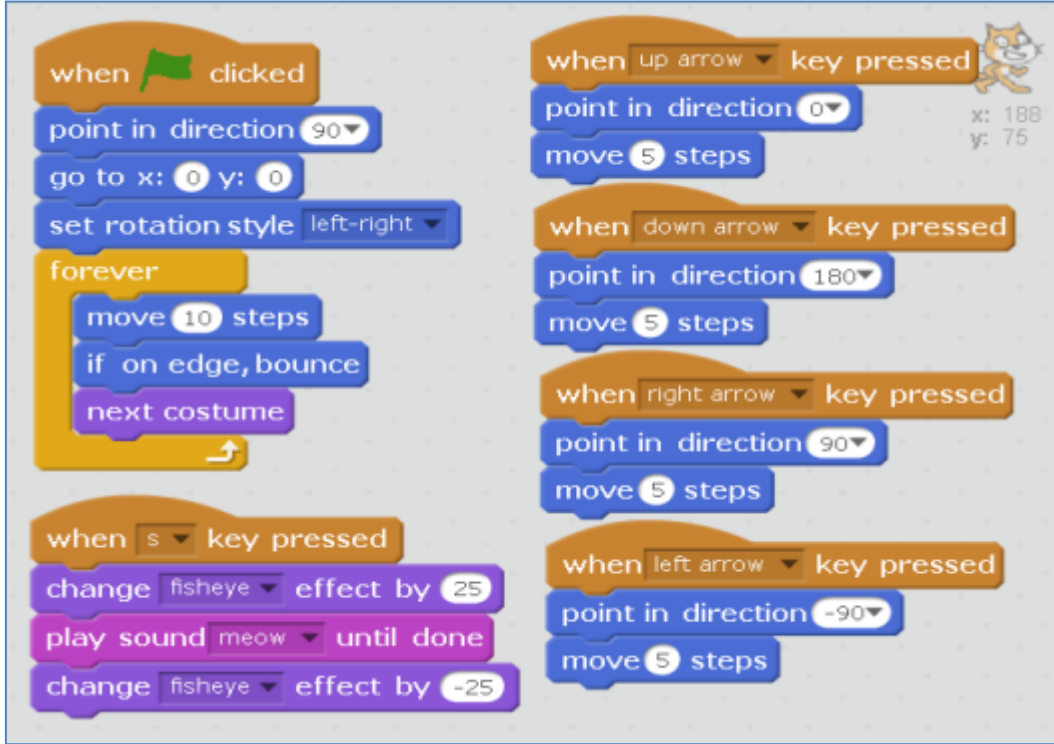
İndi isə 4 klavişlə idarə etməyi bacaran yazdığımız bu proqram aşağıdakı kimi görünəcək:



Proqramın işləməsini yoxlayın. Əla! Biz artıq klavişlər vasitəsilə obyektləri (spraytları) idarə edə bilirik. Gəlin proqramımızı bir qədər təkmilləşdirək, birinci proqramımızdakı əmrləri bu proqrama əlavə edək. Amma birinci proqramdan fərqli olaraq bu proqramda pişiyin daimi yox, yalnız “S” klavişini sıxdıqda səs çıxarmasını proqramlaşdıraq. Burada “S” klavişinin seçəndə ingilis dilində **Sound (səs)** sözünün “S” hərfi ilə başlamasını əsas götürmüşük.

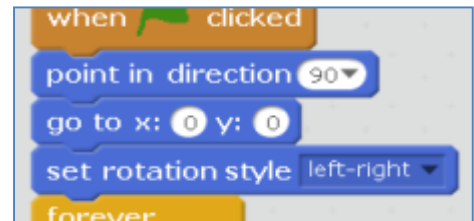
Diqqət! Proqrama klavişlərlə iş əməlləri salınıbsa, onda kompüterdə daxiletmə dilini ingilis dilinə çevirin. Yoxsa bu hərflərlə idarəetmə mümkün olmayacaq.

Proqramın mətni aşağıdakı kimidir.



Yaşıl bayraqla idarə edilən əməllər sırasında birinci 3 əməl aşağıdakı işləri görür:

- Spraytın istiqamətini sağa doğru yönəldir.
- Spraytı səhnənin ortasına aparır.
- Fırlanma stili kimi sola və sağa hərəkəti seçir.



“S” klavişi ilə idarə olunan əməllər blokunda əvvəlcə balıq gözü effekti spraytın mərkəzinə nəzərən bir qədər formasını dəyişir. Daha sonra pişiyin səsi gəlir. Üçüncü əməl isə balıq gözü effekti ilə obyektə edilmiş forma dəyişikliyini geri qaytarır.

Proqramı tam ekran rejiminə gətirin və yaşıl bayraqla proqramı başladın. Bu zaman pişik adı qaydada sola-sağa qaçmağa başlayacaq. İstiqamət oxları ilə pişiyin hərəkət 4 istiqamətdə dəyişmək olar. “S” klavişini sıxdıqda isə pişik ağızını tərpedərək miyoldayacaq.

Çox gözəl! Artıq getdikcə istədiyimizə nail oluruq.

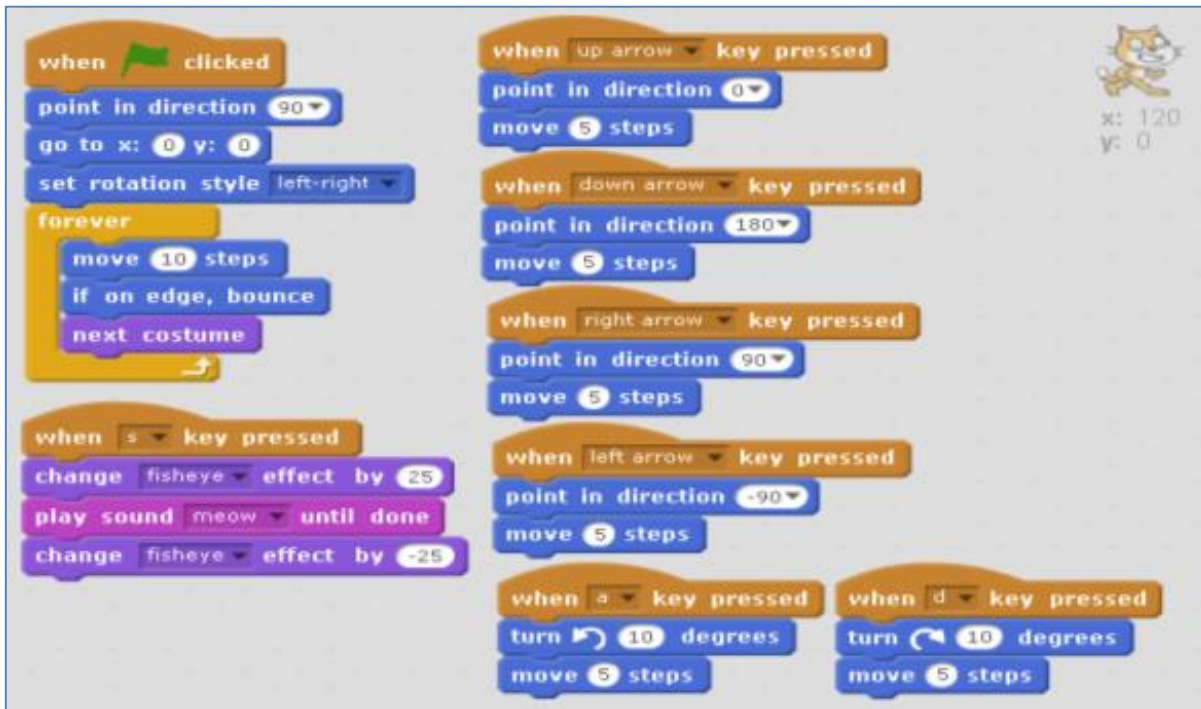


Gəlin proqramımıza bir detal da əlavə edək. Klavişləri sıxmaqla obyektimizin (spraytımızın) istiqamətini bir qədər sola və ya sağa yönəldək. “S” klavişindən soldakı və sağdakı klavişləri götürək. Bunlar “A” klavişi istiqamətə sola, “D” klavişi isə sağa döndərəcək.



Burada biz parametri mənfi etsək, yəni “10” əvəzinə “-10” yazsaq, onda sağ əvəzinə sola, sol əvəzinə isə sağa dönmə baş verəcək. Bunun səbəbini isə “Geriyə hərəkət və ya mənfi ədəd” bölməsində izah eləmişdik.

Dəyişiklikdən sonra proqram belə olacaq.



4.6. Dəyişənlər

Biz artıq proqramlaşdırmada ilk uğurlarımızı qazanmağa başlayırıq. Gəlin sonuncu proqramımıza bir də nəzər salaq. Bu proqramı daha da necə təkmilləşdirmək olar?

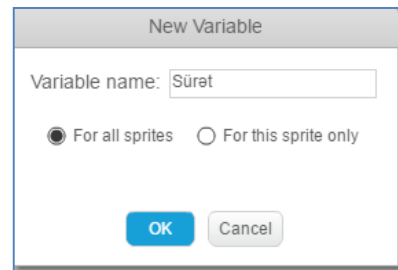
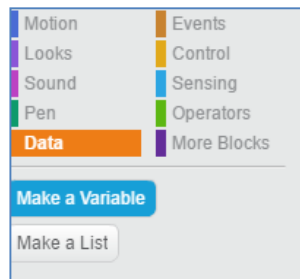
İlk növbədə proqramımızın ağ səhnəsini maraqlı bir fonla əvəz edə bilərik. Bu prosesi necə etməyi artıq bildiyimizdən bunu bir daha izah etməyəcəyik. İstədiyiniz fonu səhnəyə qoya bilərsiniz. Hətta istəsəniz səhnəyə bir neçə fon qoyub səhnə üçün proqram yazaraq onları növbə ilə göstərməklə maraqlı effekt ala bilərsiniz.

Proqramımızdakı pişiyin hərəkət sürəti 10 addım nəzərdə tutulub. Klavişlərlə istiqaməti dəyişərkən isə sürət 5 addım götürülüb. Növbəti təkmilləşdirmə kimi iki ayrı klavişi sıxmaqla obyektimizin hərəkət sürətini artırıb-azalda bilərik.

Amma burada bir şeyi nəzərdən qaçıрмаq olmaz. Biz istənilən hal üçün sürəti yadda saxlamalıyıq. Yadda saxladığımız sürəti dəyişəndən sonra yeni sürət əvvəlki yerdə saxlanılmalıdır. Hərəkət əmrində isə həmin sürəti göstərməliyik. Bütün proqramlaşdırma dillərində olduğu kimi, Scratch proqramlaşdırma dilində də məlumatları yadda saxlayıb gələcəkdə ehtiyac olduqda onu dəyişə bilmək üçün **dəyişən** və **siyahı** elementlərindən istifadə edilir. **Dəyişən** bir məlumatı yadda saxlayıb sonradan onu dəyişməyə imkan verən yaddaşdır. **Siyahı** isə birdən çox məlumatı yadda saxlayıb sonradan onları istədiyimiz vaxt sıra nömrəsinə görə dəyişməyə imkan verən yaddaşdır. Siyahıya bir çox proqramlaşdırma dilində massiv deyilir. Bu dərsimizdə biz hələ ki, yalnız dəyişənlə işləməyi öyrənəcəyik.

Beləliklə işimizə başlayaq. İlkin şərtimiz belədir. Biz artıq “A”, “S” və “D” klavişlərindən səs, sola və sağa fırlanma üçün istifadə edirik. Bu klavişlərə ən yaxın olan “Z” və “X” klavişlərindən sürəti azaltmaq və artırmaq üçün istifadə edə bilərik. Cari sürəti yadda saxlamaq üçün isə **Sürət** adlı dəyişən yaradacağıq. Bunun üçün aşağıdakı əməliyyatları aparmaq lazımdır:

Verilənlər qrupu → **Data** → **Variable name:** **Sürət** → **For all sprites** → **OK**



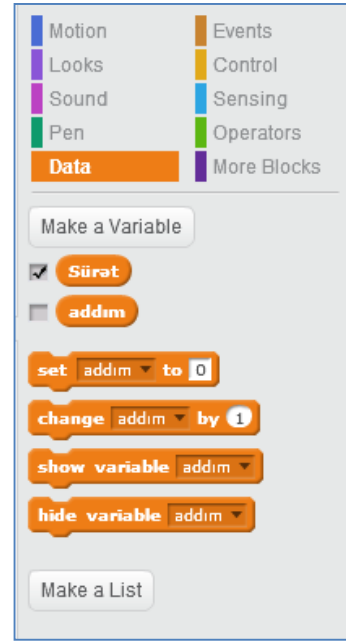
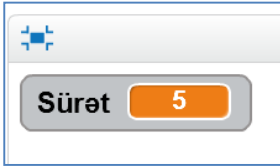


Bu zaman verilənlər qrupunda əlavə əmr sətirləri əmələ gələcək. Biz oxşar qaydada əlavə olaraq **addım** adlı dəyişən də yaratsaq, onda həmin dəyişənin də adı siyahıya düşəcək. Amma digər əmrlərin sayı dəyişməyəcək.

Dediklərimiz sağdakı şəkildə öz əksini tapır.

Dəyişənlərin adlarının qarşısındakı xanaları işarələdikdə səhnədə həmin dəyişənlərin adları və qiymətləri görünəcək. İşarələməni götürdükdə isə görüntü olmayacaq. Bizim nümunədə biz yalnız **Sürət** dəyişənini

nişanladığımızdan səhnədə yalnız bu dəyişən haqqında məlumat görünəcək.



Eyni əməliyyatı proqramın daxilindən etmək üçün show **variable** və **hide** əmrlərindən istifadə etmək olar.

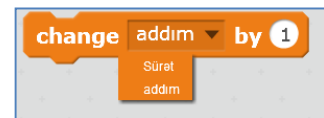


set **to** əmri verilmiş dəyişənin əvvəlki qiymətini silib ona verdiyimiz qiyməti mənimsədir. Məsələn, tutaq ki, solda gördüyümüz əmrdə əgər əmrə qədər **addım** dəyişənin qiyməti 15 olmuşsa, əmrdən sonra onun qiyməti sıfır olacaq.



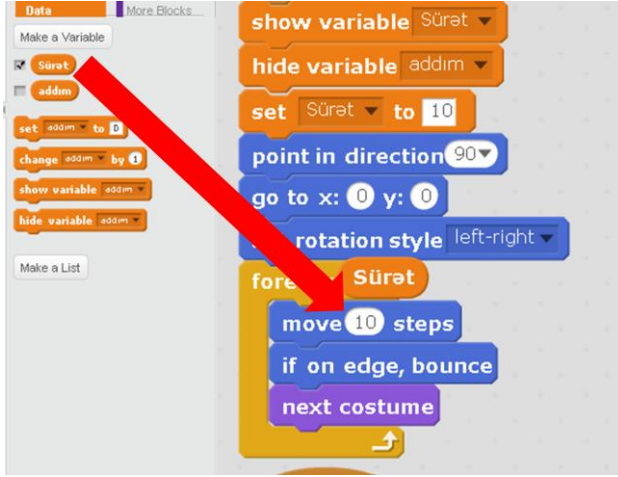
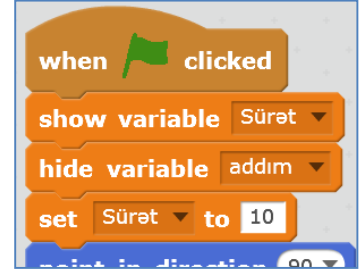
change **by** əmri verilmiş dəyişənin əvvəlki qiymətini verdiyimiz qiymət qədər dəyişdirir (müsbətdirsə artırır, mənfidirsə azaldır). Məsələn, tutaq solda gördüyümüz əmrdə əgər əmrə qədər **addım** dəyişənin qiyməti 15 olmuşsa, əmrdən sonra onun qiyməti 16 olacaq.

Bu əmrləri digər dəyişənə tətbiq etmək üçün dəyişənin adı olan hissənin sağ tərəfindəki ▼ işarəsinə sıxmaqla dəyişənlərin adları olan siyahını açmalı və həmin siyahıdan istədiyimiz dəyişənin adını seçmək olar.

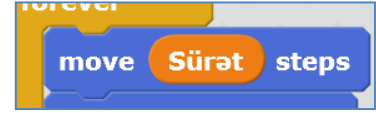


Bizə hələ ki, bir dəyişən **Sürət** dəyişəni lazımdır.

Proqram başlayanda sprayta başlanğıc sürət olaraq **10** qiyməti verək. Bundan başqa **Sürət** dəyişəninin qiymətini səhnədə göstərək, **addım** dəyişəninin qiymətini isə yox. Sağdakı şəkildə bu əmrləri görürsünüz.

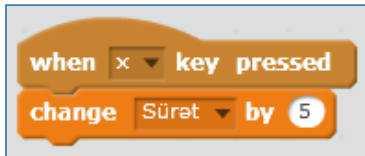


Daha sonra **Verilənlər qrupundan** dəyişənlərin adları olan sahədən **Sürət** dəyişəninin adını daşayıb **move 10 steps** əmrində 10 qiyməti olan sahəyə yerləşdirməli. Onda əmrin görünüşü belə olacaq:

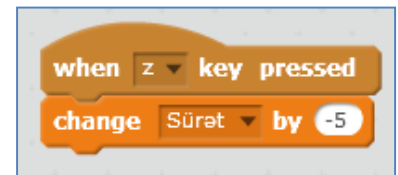


Bu vəziyyətdə proqramı başlatdıqda yenə proqram əvvəlki kimi işləyəcək.

Sürəti dəyişmək üçün “Z” və “X” klavişləri ilə idarə olunan proqram blokları yazmaq lazımdır. Biz sürəti hər dəfə 5 vahid dəyişəcəyik. Əvvəlcə sürəti artırmaq üçün proqram yazaq. Bu halda **change** **by** əmrindən istifadə edəcəyik. Solda əmrlərin yazılışını görürsünüz.



Oxşar qaydada sürəti azaltmaq üçün əmrləri yaza bilərik. Sağdakı şəkildə də bu əmrlərin yazılışını görürsünüz.



Proqramı tam pəncərə rejiminə keçirin. İndi isə yaşıl bayraq düyməsini sıxmaqla proqramı başladın.

Əvvəlki klavişlərdən savayı “Z” və “X” klavişləri vasitəsilə sürəti də idarə etmək olar. Sürətin dəyişməsinə səhnədə sol küncdəki indikatora görmək olar.

Proqramın ümumi görünüşünü aşağıdakı şəkildə görürsünüz.



```
when clicked clicked
show variable Sürət
hide variable addım
set Sürət to 10
point in direction 90
go to x: 0 y: 0
set rotation style left-right
forever
move Sürət steps
if on edge, bounce
next costume
when s key pressed
change fisheye effect by 25
play sound meow until done
change fisheye effect by -25
when up arrow key pressed
point in direction 0
move 5 steps
when down arrow key pressed
point in direction 180
move 5 steps
when right arrow key pressed
point in direction 90
move 5 steps
when left arrow key pressed
point in direction -90
move 5 steps
when a key pressed
turn 10 degrees
move 5 steps
when d key pressed
turn 10 degrees
move 5 steps
when z key pressed
change Sürət by -5
when x key pressed
change Sürət by 5
```

Sizin nə hiss etdiyinizi təsəvvür edirik. Çünki, vaxtilə həmin hissləri biz də keçirmişik.

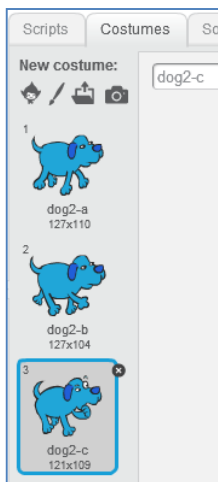
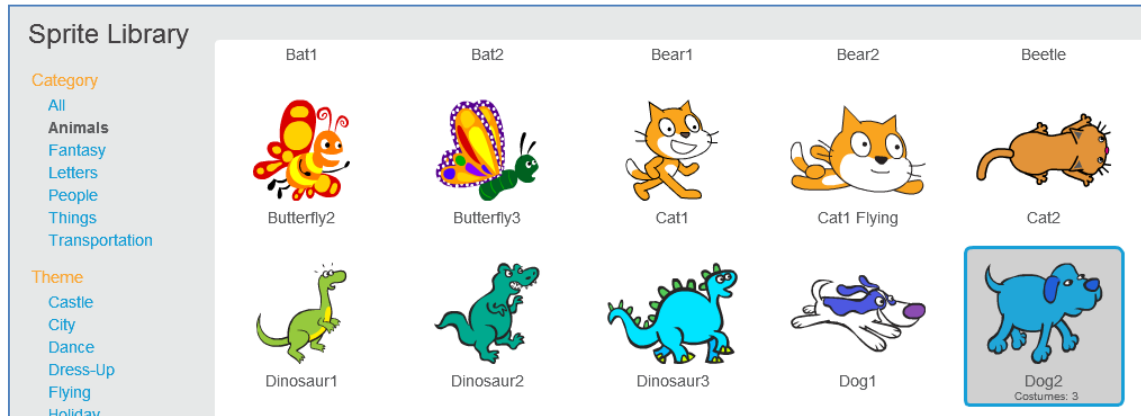
4.7. İt və pişik

Həyatda belə bir səhnəni çox görmüşük. İt hürə-hürə pişiyi qovur, pişik isə qaçaraq ondan qurtulmaq istəyir. Əgər it pişiyə çatsa onu dişləmək istəyəcək və pişik isə miyoldayaraq itdən qaçmağa çalışacaq.

İndi isə gəlin bu situasiyaya uyğun bir oyun quraq. İt hürə-hürə pişiyi qovur. Pişik isə qaçıb aradan çıxmaq istəyir. Əgər it pişiyə çatırsa, pişik çığırır və yoxa çıxır. Bu zaman it qaçmağına bir müddət fasilə verir və yenidən qaçmağa başlayır.

Pişik bir müddətdən sonra səhnənin mərkəzində peyda olur və yenə qaçmağa başlayır. Əgər it pişiyi 10 dəfə tuta bilsə, onda mükafat qazanılır.

Yeni proqram yazmağa başlayarkən, pişik personajı artıq proqrama daxil edilmiş olur. Əlavə yeni bir it personajı əlavə edək. Bunun üçün Scratch proqramının 4 sahəsindən yeni sprayt daxil etmək təlimatını yerinə yetirərək aşağıdakı it personajını proqrama daxil edək.

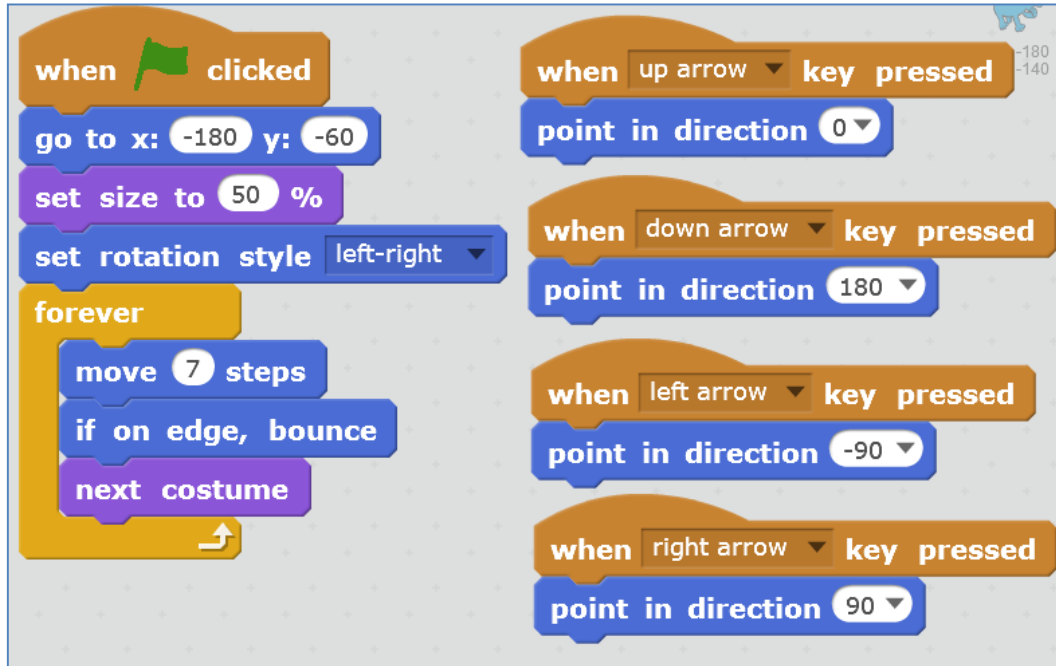


Spraytlar kitabxanasının **Animals** qrupundakı **Dog2** spraytı bizim məqsədlərə uyğun gəlir. Çünki bu spraytın 3 kostyumu var. Bu kostyumların köməyilə biz itə hərəkət verə bilərik.

Daha bir vacib əməliyyatı da yerinə yetirək. Scratch proqramının 4 sahəsindən istifadə edərək spraytların adını uyğun olaraq “İt” və “Pişik” adları ilə dəyişək.



Gəlin əvvəlcə it üçün aşağıdakı şəkildə gördüyünüz əmrlər blokunu yazaq.



Şəkildən görüldüyü kimi, əmrlər bloku 2 hissədən ibarətdir:

- Yaşıl bayrağı sıxmaqla idarə edilən əmrlər bloku;
- İstiqamətləri göstərən klavişləri sıxmaqla idarə edilən əmrlər bloku. Burada 4 istiqamətli (yuxarı, aşağı, sola və sağa) əmrlər verilib.

İkinci qrupa daxil olan əmrlər haqqında əvvəlki məşğələlərdə danışmışıq. Ona görə də həmin əmrlərə şərh verməyəcəyik.

Yaşıl bayraqla idarə edilən əmrlərin təyinatı ilə isə tanış olaq.

```
when green flag clicked
  go to x: -180 y: -60
  set size to 50 %
  set rotation style left-right
  forever
    move 7 steps
    if on edge, bounce
    next costume
```

Yaşıl bayrağa sıxdıqda əmrinin təyinatı proqramı başlatmaqdır.

əmri personajımızı, yəni iti proqram başlayan kimi Səhnənin (-180: -60) nöqtəsinə aparır.

əmri personajın (itin) ölçüsünü 50% həcminə qədər kiçildir.

əmri iti sola-sağa hərəkət etməyə istiqamətləndirir.

Növbəti **forever** əmrinin daxilində qruplaşmış əmlər bizə əvvəlki proqramlardan məlum olan personajı hərəkət etdirmək üçün nəzərdə tutulan əmlərdir. Sadəcə olaraq əmrində addım 7 götürülüb.

Yaşıl bayraq düyməsini sıxmaqla bu blokun işləməsini yoxlamaq olar. İstiqamət oxları ilə itin hərəkət istiqamətini dəyişmək olar.

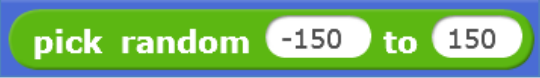
İndi isə pişiyin hərəkəti üçün əmlər blokunu yazacaq.


```
when green flag clicked
  go to x: 0 y: 0
  set size to 50 %
  set rotation style left-right
  forever
    turn pick random -150 to 150 degrees
    move pick random 15 to 45 steps
    if on edge, bounce
    next costume
```




Bu blokdakı 4 əmr itin əmrlərinə oxşayır. Sadəcə pişiyi ilkin daşıma koordinatı Səhnədə (0:0) nöqtəsidir.

Əsas dəyişiklik növbəti **forever** əmrinin daxilində qruplaşmış əmrlərdədir.

Diqqəti çəkən bir məqamı qeyd etmək istərdik. Burada iki əmr daxilində **pick random** əmrindən istifadə edir.  Bu əmr verilmiş intervalda ixtiyari ədəd verir. Məsələn: verdiyimiz nümunədə **forever** dövr əmrinin hər addımında əmr **-150** ilə **150** arasında ixtiyari ədəd götürür.

Hərəkət sürətini bildiren  əmrində addım dəyişəndir. Bu əmr hər dəfə sürət üçün **pick random** əmrinin köməyiylə 15-ə 25 arasında ixtiyari bir ədədi götürür. Bu isə sürətin hər dəfə fərqli olmasına gətirib çıxarır.

Növbəti yeni əmr  əmridir. Bu əmr də yuxarıda dediyimiz kimi **pick random** əmrinin köməyiylə **-150** və **150** arasında ixtiyari ədədi götürür və bu ədədə uyğun dönmə bucağı seçir.

Burada bir şeyi etməyə dəyər. Aşağıdakı iki əmrlər blokunun işinin nəticəsi eyni olacaq. Çünki bir istiqamətdə mənfi bucaq dönməsi əks istiqamətdə müsbət bucaq dönməsinə bərabərdir:



Bu əmrlər ona görə bu formada seçilir ki, pişiyin hərəkətinin istiqamətini və sürətini əvvəlcədən müəyyənləşdirmək mümkün olsun.

Bəs görəsən it pişiyi tuta bilərmə? Bunun üçün itin hərəkət istiqamətini, yuxarıda dediyimiz kimi, oxlarla idarə etmək olar.

Bəs itlə pişik hansısa nöqtədə üz-üzə gələndə nə baş verməlidir? Yuxarıda şərtləşdiyimiz kimi, əgər it pişiyə çatırsa, pişik çığırır və yoxa çıxır. Bu zaman it qaçmağına bir müddət fasilə verir və yenidən qaçmağa başlayır. Gəlin bu şərtləri ödəyən nəticəni almaq üçün uyğun əmrləri yazaq.

Əvvəlcə pişik üçün bu vəziyyətin əmrlər blokunu yazaq.

```
if touching It ? then
  repeat 3
    play sound meow until done
  hide
  go to x: 0 y: 0
  wait 5 secs
  show
```

Məntiqlə biz belə bir şərti yoxlamalıyıq: **Əgər pişik itlə təmasdadırsa onda müəyyən əməlləri yerinə yetiririk.**

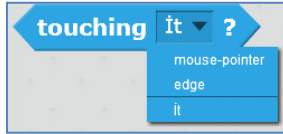
Bir şərt əmri kimi **İdarəetmə (Control)** qrupundakı aşağıdakı əmri götürürük:



Pişiyin itlə təmasda olması halını yoxlamaq



üçün **Sensor (Sensing)** qrupundakı istifadə edəcəyik. Bu əmrin parametrlər hissəsində qara üçbucağı sıxaraq açılan siyahıdan **İt** sözünü



seçirik.

əmrindən

Daha sonra Pişik 3 dəfə miyoldayıb, yoxa çıxacaq. Görünməz vəziyyətdə Səhnənin mərkəzinə daşınacaq. **5** saniyə gözlədikdən görünəcək.

```
when clicked
  go to x: 0 y: 0
  set size to 50 %
  set rotation style left-right
  forever
    turn pick random -150 to 150 degrees
    move pick random 15 to 45 steps
    if on edge, bounce
    next costume
    if touching It ? then
      repeat 3
        play sound meow until done
      hide
      go to x: 0 y: 0
      wait 5 secs
      show
```

Yeni yazdığımız bloku proqramımıza əlavə edirik. Proqramın son görünüşü soldakı kimi olacaq.

Tam ekran rejiminə keçin. **Yaşıl bayraq** düyməsini sıxıb proqramın düzgün işləməsini yoxlayın. Hansısa halda bu iki personaj toqquşacaq. Bu halda yeni yazdığımız proqram bloku işləməlidir. Əgər işləməsə, nədə səhv etdiyinizi araşdırın. **Qırmızı dairə** düyməsini sıxıb proqramı durdurun, adi ekrana keçin.

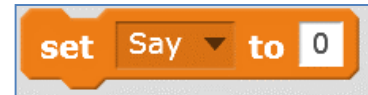


İndi isə vəziyyətin İt personajı üçün hissəsinin proqram blokunu yazaq. Bildiyimiz kimi biz İtlə Pişiyin toqquşmalarının sayını aparmalıyıq. Bunun üçün biz **Say** adlı bir dəyişən götürməliyik. Bunun üçün **Verilənlər (Data)** qrupunda **Make a Variable** düyməsini sıxmaqla **Say** adlı dəyişən yaradıırıq.

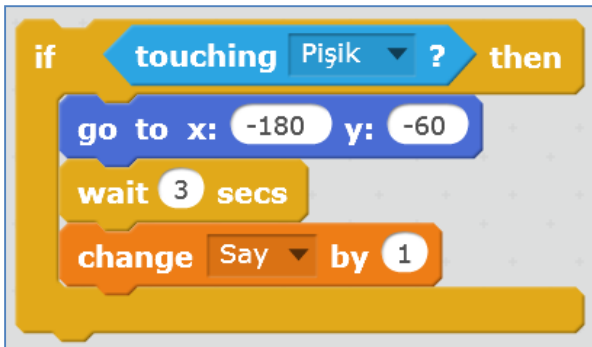
Bu halda həmin qrupda aşağıdakı əmrlər əmələ gələcək:



Toqquşmaların sayını apararkən başlanğıcda **Say** dəyişəninə “0” qiyməti



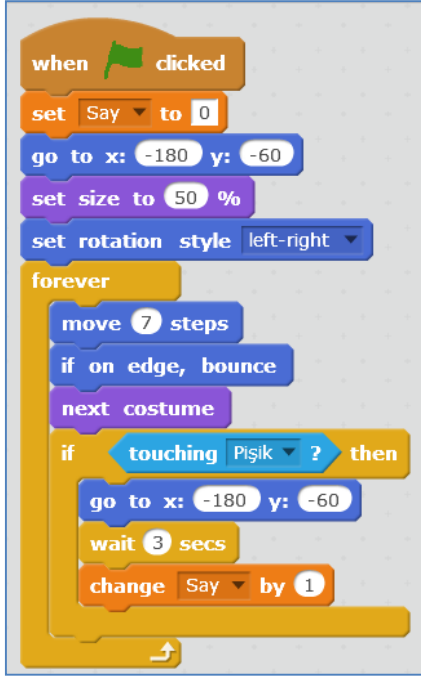
mənimsətmək lazımdır. Bunun üçün proqramın əvvəlində əmrini yerləşdirmək lazımdır.



İki personajın toqquşması zamanı situasiyanın İt üçün olan proqram hissəsi soldakı şəkildə verilib. Prinsipcə bu Pişik üçün olan proqram blokuna bənzəyir. Sadəcə olaraq bu halda **İti** gizlədib yenidən göstərməyə ehtiyac yoxdur. Personaj da (-180 : -60) daşınır. Əvəzində **Say** dəyişənini bir vahid artıran



əmr salınıb. Bu ona görədir ki, toqquşmaların sayını hesablaya bilək.



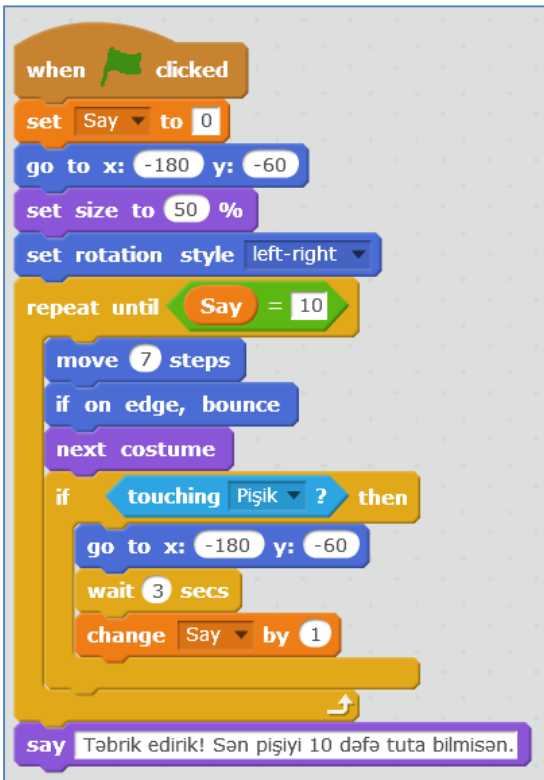
Yazdığımız yeni əmrlər blokunu proqramımıza əlavə edək. Soldakı şəkildə həmin proqram hissəsi verilib.

Səhnəni tam ekran rejiminə keçirək və **Yaşıl bayraq** düyməsini sıxıb proqramı başladaq. Görəcəyik ki, proqram işləyir, say hesablanır. Amma say qiyməti durmadan artır. Bizim proqram isə say **10**-a bərabər olanda durmalıdır. Səbəb nədir?



Məsələ ondadır ki, biz proqramımızda sonsuz dövr operatoru olan **forever** əmrindən istifadə etmişik. Bizim halda bu əmr uyğun gəlmir. Ona görə də başqa dövr əmrindən istifadə etmək lazımdır.



əmrini bizim tələbləri ödəyir. Bu əmrin daxilindəki əmrlər parametrdə verilmiş şərt ödənməyə qədər işləyəcək. Bizim qurduğumuz oyunun proqramı toqquşmaların sayı **10**-a bərabər olanda işini dayandırmalıdır. Deməli biz **forever** əmrinin əvəzinə **repeat until** əmrindən istifadə etməliyik.




Soldakı şəkildə əmrlər bloku göstərilib. **forever** əmrinin yerinə **repeat until** əmrindən istifadə edilmişdir. Bu əmrin parametrisə hissəsində

Operatorlar (Operators) qrupundan  operatorunu və **Verilənlər (Data)** qrupundan  dəyişənini götürüb bizə gerek olan

repeat əmrini üçün şərtini qururuq.



Daha bir yeni əmr  əmridir. Bu əmrin parametrisə hissəsinə “**Təbrik edirik! Sən pişiyi 10 dəfə tuta bilmisən.**” Mətnini yazırıq. Nəticədə proqram bitəndə həmin



mətn **Səhnədə** görünəcək.

Oxşar düzəlişi (**forever** əmrini **repeat until** əmri ilə əvəz etməyi) **Pişik** personajı üçün də edirik. Bəzən elə ola bilər ki, biz proqramı saxladan zaman **Pişik** görünməz vəziyyətdə ola bilər. Bu halda proqramı təkrar başladanda **Pişik**

personajı görünməz olacaq. Ona görə də proqramın lap əvvəlinə **show** əmrini salmaq lazımdır ki, proqram başladılan zaman heç bir vəziyyətdən asılı olmayaraq **Pişik** personajı **Səhnədə** görünsün.

Aşağıda **İt** və **Pişik** personajları üçün yazılmış proqramların görüntüsü verilib.

```

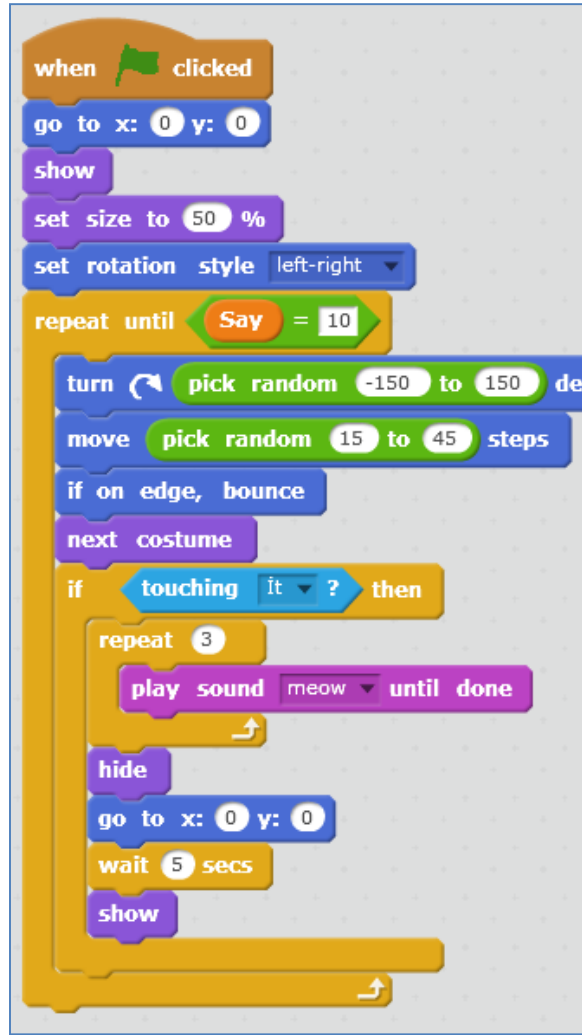
when green flag clicked
  set Say to 0
  go to x: -180 y: -60
  set size to 50 %
  set rotation style left-right
  repeat until Say = 10
    move 7 steps
    if on edge, bounce
    next costume
    if touching Pişik? then
      go to x: -180 y: -60
      wait 3 secs
      change Say by 1
  say Təbrik edirik! Sən pişiyi 10 dəfə tuta bilmisən.

when up arrow key pressed
  point in direction 0

when down arrow key pressed
  point in direction 180

when left arrow key pressed
  point in direction -90

when right arrow key pressed
  point in direction 90
  
```



Son dəfə bu proqram üçün **Səhnəni** böyüdü və proqramı başladın. ƏLA! Proqramımız işlədi. Amma gözəl bir fon olmasa proqramın görünüşü pis olur. Səhnəyə istədiyiniz fonu quraşdırın və yeni yazdığınız oyundan həzz alın.



4.8. Proqramda riyaziyyat

Artıq yavaş-yavaş proqramlaşdırmada ilk uğurlarımızı qazanmağa başlayırıq. Gəlin sonuncu proqramımıza bir də nəzər salaq. Bu proqramı daha da necə təkmilləşdirmək olar?

Gəlin proqramı yenidən başladın və dediyimiz ardıcılığı yerinə yetirin.

- İki dəfə “Z” klavişini sıxaraq sürəti sıfıra endirin. Bu bizim pişik nə irəli, nə də geri hərəkət etməyəcək. Yalnız yerində sayacaq.
- İki dəfə “Z” klavişini yenidən sıxaraq sürəti “-20” qiymətinə düşürün. Pişik səhnənin sol divarına qədər hərəkət edib yerində saymağa başlayacaq.
- İndi sol ox klavişini sıxın. Pişik sağa hərəkət edib divara çatan kimi yenə yerində sayacaq.
- Eynilə yuxarı və aşağı ox klavişlərini sıxın və eyni mənzərəni görün.

Nə baş verir. Biz proqramda nəzərdə tutduğumuzun əksinə olaraq pişik sol klavişi sıxanda sağa, sağ klavişi sıxanda sola, yuxarı klavişi sıxanda aşağı və aşağı klavişi sıxanda isə yuxarı hərəkət edəcək. Bundan başqa divara çatan kimi yerində sayacaq. Buna səbəb sürətin mənfi qiymət almasıdır. Bəs elə etmək olarmı ki, sürət mənfi qiymət almasın?

Bunun üçün biz sürəti aşağı salmaq üçün olan proqram blokunda məhdudiyyət qoymalıyıq. Bu məhdudiyyət nədən ibarətdir. **Sürət** dəyişəni mənfi qiymət alan kimi onun qiymətini sıfır edəcəyik.

Siz yəqin ki, riyaziyyatdan “<”, “=” və “>” münasibətləri ilə tanışsınız. Bunlar kiçikdir, bərabərdir və böyükdür münasibətlərini göstərir. Məsələn, $3 < 6$, $8 > 5$, $3 + 4 = 7$, $-6 < 2$ və s. Ədəd oxunda solda yerləşən ədəd sağdakından kiçikdir. Mənfi ədəd həmişə ədəd oxunda sıfırdan və istənilən müsbət ədəddən solda yerləşdiyindən, alınmış sürətin sıfırdan kiçik olmasını yoxlamaq kifayətdir.



Şəkildə mənfi ədədlər göy, müsbət ədədlər qırmızı və sıfır isə qara nöqtə ilə verilib. Göründüyü bütün göy rəngli nöqtələr qara və qırmızı rəngli nöqtələrə nəzərən solda yerləşir.

Proqramda riyazi münasibətlərdən istifadə etmək üçün **Operatorlar qrupu** mövcuddur.

Bu qrupa əsas riyazi əməllər (+, -, *, /), münasibət əməlləri (<, =, >), riyazi məntiq əməlləri (**and**, **or**, **not**) daxildir.

Bundan başqa bu qrupda randomizasiya, qalıq (mod), yuvarlaqlaşdırma (round) əməlləri var.

Lap aşağıda isə daha bir əmr də var. Həmin əmrin parametr olan hissənin sağ tərəfindəki ▼ işarəsinə sıxmaqla əlavə riyazi funksiyaların adları olan siyahını açıb həmin siyahıdan istədiyimiz funksiyanı seçə bilərik. Aşağıda soldakı şəkildə həmin funksiyaların siyahısı verilib.

abs
floor
ceiling
sqrt
sin
cos
tan
asin
acos
atan
ln
log
e ^
10 ^

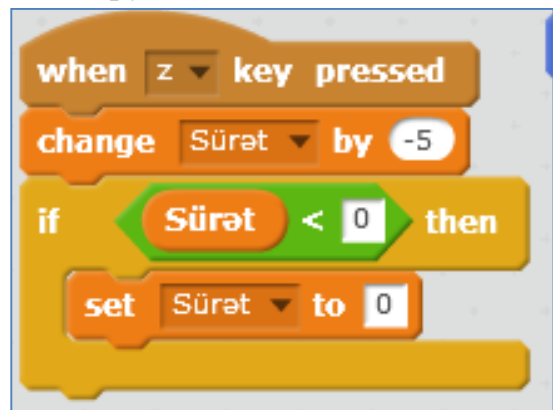
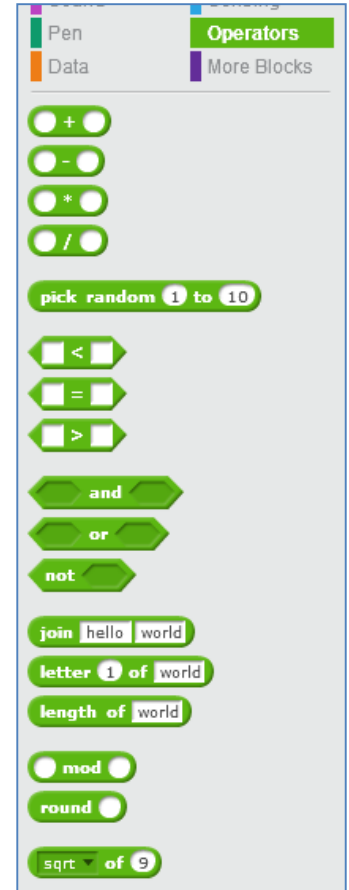
Bu funksiyalar haqqında yuxarı siniflərdə riyaziyyat fənnində məlumat verilir. Biz də gələcək dərslərimizdə bu funksiyalardan istifadə etdikcə onların izahını verəcəyik.

İndi isə bizə yalnız kiçikdir (“<”) əməli lazımdır.

Bir şeyə də diqqət yetirmənizi istərdik. Bu qrupdakı əmrlərin heç biri pəzl formalı deyil. Onlar yalnız digər qruplarda parametrləri olan əmrlərinin daxilində istifadə edilir.

İdarəetmə qrupunda şərt əmri var. Biz ondan istifadə edəcəyik. Şərtimiz isə belə olacaq: əgər **Sürət** dəyişəninin qiyməti sıfırdan kiçikdirsə, onda **Sürət** dəyişəninin qiymətini sıfıra bərabər etmək.

Sağdakı proqram blokunda həmin dediklərimiz göstərilib. “Z” klavişini sıxdıqda **Sürət** dəyişəninin qiyməti 5 vahid azaldılır. **Sürət** dəyişəninin qiyməti mənfi olan kimi ona **sıfır** mənimsədilir.





Proqramın tam mətni aşağıda verilib.

```
when clicked
  show variable Sürət
  hide variable addım
  set Sürət to 10
  point in direction 90
  go to x: 0 y: 0
  set rotation style left-right
  forever
    move Sürət steps
    if on edge, bounce
    next costume
  when up arrow key pressed
    point in direction 0
    move 5 steps
  when down arrow key pressed
    point in direction 180
    move 5 steps
  when right arrow key pressed
    point in direction 90
    move 5 steps
  when left arrow key pressed
    point in direction -90
    move 5 steps
  when a key pressed
    turn 10 degrees
    move 5 steps
  when d key pressed
    turn 10 degrees
    move 5 steps
  when z key pressed
    change Sürət by -5
    if Sürət < 0 then
      set Sürət to 0
  when s key pressed
    change fisheye effect by 25
    play sound meow until done
    change fisheye effect by -25
  when x key pressed
    change Sürət by 5
```

5. Müxtəlif proqram nümunələri

Bu ana kimi biz proqramımızda yalnız bir spraytdan istifadə etmişik. Amma daha səviyyəli proqram yazmaq istəsək, onda çoxlu sayda spraytlardan istifadə etməliyik.

Öncə yeni yazmaq istədiyimiz proqramın nə edəcəyini aydınlaşdırmaq lazımdır.

Yəqin ki, hamınız kompüterdə oyun oynamısınız. Amma heç kim təsəvvürünə gətirmir ki, bir oyundan ötrü proqram yazmaq üçün neçə insan nə qədər zəhmət sərf edir. Bu qədər insanın işini bir məqsədə yönəltməkdən ötrü əvvəlcə oyunun məqsədləri tam müəyyən edilməlidir. Bu mərhələ *problemin qoyuluşu* və ya *məsələnin qoyuluşu* adlanır.

Sonra onun epizodları detallarına qədər aydınlaşdırılmalıdır. Oyunun bütün detalları sözlə ifadə edilməlidir. Bu mərhələ *layihənin hazırlanması* adlanır.

Sonda isə həmin oyun üçün hansısa proqramlaşdırma dilində proqram yazılmalıdır. Bu mərhələ *kodlaşdırma* adlanır.

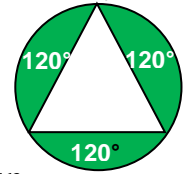


5.1. Üçbucaq, kvadrat və digər çoxbucaqlı fiqurların çəkilməsi

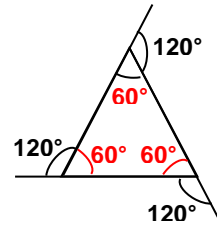
Üçbucaq, kvadrat, beşbucaqlı və s. Bütün bunlar hamısı çoxbucaqlı adlanır.

5.1.1. Üçbucağın çəkilməsi

Bizə üçbucaq çəkən proqram yazmaq lazımdır. Bu vəsaitə əlavədə verdiyimiz bilgilərdə bildiyimiz kimi tam dairə 360 dərəcədir. Əgər dairəni 3 bərabər hissəyə ayırsaq, bu zaman dairəni əhatə edən çevrə üzərində bir-birindən bərabər məsafədə yerləşən 3 nöqtə alacağıq. Bu nöqtələrin birləşməsi bizə bərabər tərəfli üçbucaq verəcək. Üçbucağın hər tərəfinin dairədən ayırdığı bucaq 120 dərəcə olacaq.



Biz sağdakı şəkildən görürük ki, 120 dərəcə üçbucağın xarici bucaqlarıdır. Onun daxili bucaqları isə 60 dərəcəyə bərabərdir.



Aşağıda solda ekranda üçbucaq çəkmək üçün proqramın skriptləri verilmişdir.

```

when green flag clicked
  show
  clear
  pen up
  go to x: 0 y: 0
  set pen size to 2
  set pen color to red
  pen down
  repeat 3
    move 100 steps
    turn 120 degrees
  pen up
  hide
  
```

Gəlin bu proqramın əmrlərini araşdıraq. Adəti üzrə proqram **yaşıl bayrağa sıxdıqda** əmri ilə başlayır. Bizim proqramda hansı sprayt götürülmüşsə, qrafik şəkli də o çəkəcək. Adətən yeni proqram yazmağa başlayanda susma halında pişik götürülür. Amma Siz pişiyin əvəzinə, qələm və ya ixtiyarı digər sprayt götürə bilərsiniz.

show əmri həmin spraytı səhnədə göstərəcək. Bu əmr ona görədir ki, proqramın sonunda **hide** əmri vasitəsilə sprayt gizlədilir. Əgər proqramın əvvəlində sprayt göstərilməsə, onda biz ekranda nəyin şəkil çəkdiyini görməyəcəyik.

Daha sonra **clear** əmri ekranda olan görüntünü silir. Əgər proqram bir neçə dəfə ardıcıl buraxılırsa, bu zaman ekranda əvvəlki görüntü qalır və proqramın yeni başladılmasının işinin nəticəsi görünür. Bu problemi **clear** əmri aradan götürür.

Adətən biz yazı yazarkən yeni cümləni yazmağa başlamaq üçün qələm olan əlimizi qaldırıb qələmi yazını yazmaq istədiyimiz nöqtəyə gətiririk və əlimizi endirərək

qələmi kağızın üzərinə toxundurub yazını yazmağa başlayırıq. Scratch proqramlaşdırma dilində bu funksiyaları **pen up (qələmi qaldır)**, **pen down (qələmi endir)** və **goto x: 0 y: 0 (get x: 0 y: 0)** əmrləri yerinə yetirir.

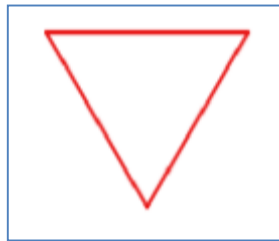
Burada daha 2 əmri qeyd etmək istərdik. **set pen size to 2 (qələmin ölçüsünü 2 müəyyənləşdir)** və **set pen color to (qələmin rəngini müəyyənləşdir)** əmrləri uyğun olaraq qələmin ölçünü 2 müəyyən edir, digəri isə onun rəngini qırmızı edir.

Növbəti əmr **repeat 3** əmridir. Bu əmr onun daxilindəki əmrləri **3** dəfə təkrar edir. **repeat** əmrinin daxilində isə **move 100 steps (100 addım irəli get)** və **turn 120 degrees (saat əqrəbi istiqamətin 120 dərəcə döndər)** əmrləri yazılıb.

İndi isə gəlin yazdığımız proqramı tam oxuyaq.

- Yaşıl bayrağı sıxaraq proqramı başladırıq.
- Spraytı göstəririk.
- Qələmi qaldırıq.
- Səhnənin mərkəz nöqtəsinə (x: 0 və y: 0) daşırıq (keçirik).
- Qələmin ölçüsünü (qalınlığını) 2 müəyyənləşdiririk.
- Qələmin rəngini qırmızı seçirik. Bunun üçün kursoru əmrdəki rəngi göstərən hissəsinə gətirib, sonra pipetkanın köməyi ilə ekrandakı ixtiyari rəngi (bizim halda **qırmızı**) seçirik.
- Qələmi endirik.
- 3 dəfə aşağıdakı əmrləri yerinə yetiririk:
 - 100 addım irəli gedirik
 - Saat əqrəbi istiqamətində 120 dərəcə döndürük
- Qələmi qaldırıq.
- Spraytı gizlədirik.

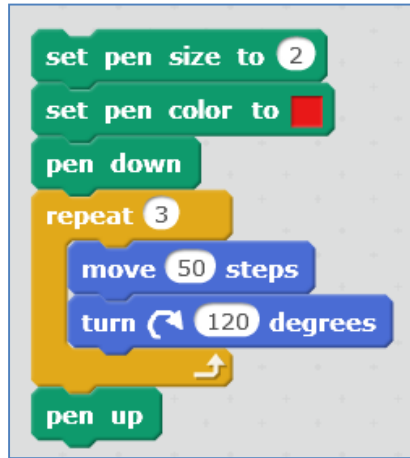
Verilmiş proqramı yazın və onu başladın. Səhnədə aşağıdakı görüntünü alacağıq:



Gəlin proqramımızı bir qədər təkmilləşdirək. Bizə səhnənin ortasında bir-birindən eyni məsafədə yan-yanı duran üç ədəd üçbucaq çəkmək lazımdır. Bunun üçün



aşağıdakı əmrləri **go to x:-100 y:0**, **go to x:0 y:0** və **go to x:100 y:0** əmrləri ilə uyğun nöqtələrə daşınmaqla 3 dəfə yerinə yetirmək lazımdır:



goto əmri daxil olmaqla bu əmrlər qrupu 3 dəfə təkrarlananda cəmi 24 əmr yazmış oluruq. Əgər üçbucaqların sayı çox olsa onda əmrlərin də sayı çox olur və bu proqramın böyüməsinə səbəb olur. Belə vəziyyətlərdə dövr əmrlərindən istifadə etmək olar. **İdarəetmə (Control)** qrupunda aşağıdakı dövr əmrləri məhz bu məqsədlər üçün nəzərdə tutulub:



repeat əmri göstərilən say qədər onun daxilindəki əmrləri təkrar edir.

forever əmri daxilindəki əmrləri sonsuz sayda təkrar edir. Bu əmrdən biz artıq əvvəllər istifadə etmişik.

Repeat until əmri göstərilən şərt yerinə yetirilməyə qədər onun daxilindəki əmrləri təkrar edir.

Bizim hal üçün **repeat** əmri **3** qiyməti ilə uyğun gəlir. Sadəcə olaraq **x** dəyişəninin qiymətini dövr daxilində dəyişərək ona “-100”, “0” və “100” mənimsətmək lazım gələcək. Bunun üçün aşağıdakı **Hərəkət (Motion)** qrupundakı



əmrlərindən istifadə edəcəyik. **set x to** əmri **x** üçün başlanğıc qiyməti (-100) müəyyənləşdirəcək. **change x by** əmri isə dövr daxilində **x**-in qiymətini 100 vahid dəyişəcək.

```
when clicked
  show
  clear
  pen up
  go to x: -100 y: 0
  set pen size to 2
  set pen color to red
  repeat 3
    pen down
    repeat 3
      move 50 steps
      turn 120 degrees
    pen up
    change x by 100
  hide
```

İndi isə proqramımızın mətni və işləmə prinsipi ilə tanış olaq. Birinci dörd əmrin təyinatı ilə artıq tanış olmuşuq.

əmrini spraytı (x:-100 y:0) nöqtəsinə daşıyır.

Sağda gördüyünüz şəkildə növbəti əmrlər verilmişdir.

repeat 3 əmrini onun daxilindəki əmrləri üç dəfə təkrar edir.

Dövrün daxilindəki üç əmr bizə artıq tanışdır.

pen down əmrini qələmi

endirir. Daxili **repeat 3** əmrini **move 50 steps** və **turn 120 degrees** əmrlərini 3 dəfə təkrar edir. Hər dəfə

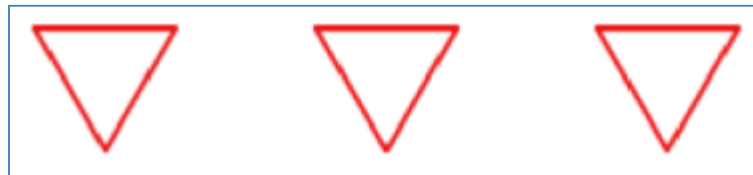
daxili dövr bitdikdə **pen down** əmrini qələmi qaldırır. Əgər diqqət edirsinizsə, **move 50 steps** əmrində addım 50 götürülüb. Bu da ona görədir ki, üçbucaqların sayı çoxalanda onların səhnədə sərbəst yerləşməsi üçün yer çatmaya bilər. İstəsəniz addımın qiymətini 100 və ya başqa götürün və alınan nəticəni müqayisə edin.

```
repeat 3
  pen down
  repeat 3
    move 50 steps
    turn 120 degrees
  pen up
  change x by 100
```

Burada yeni olan **change x by 100** əmridir. Bu əmr x-in qiymətini 100 addım sağa dəyişir (başqa sözlə əvvəlki qiymətin üzərinə 100 artırır.).

Proqramın sonunda **hide** əmrini vasitəsilə sprayt gizlədilir.

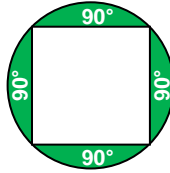
İndi yaşıl bayrağı sıxmaqla proqramı başladın. Nəticədə aşağıdakı görüntü alınacaq.



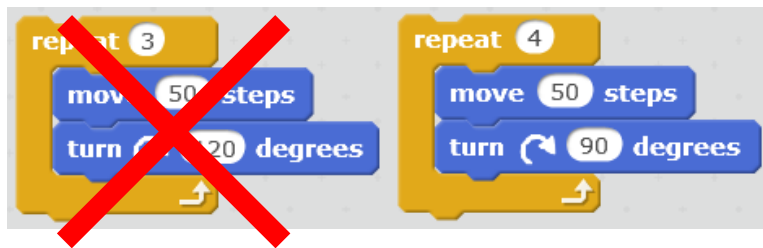


5.1.2. Kvadratın çəkilməsi

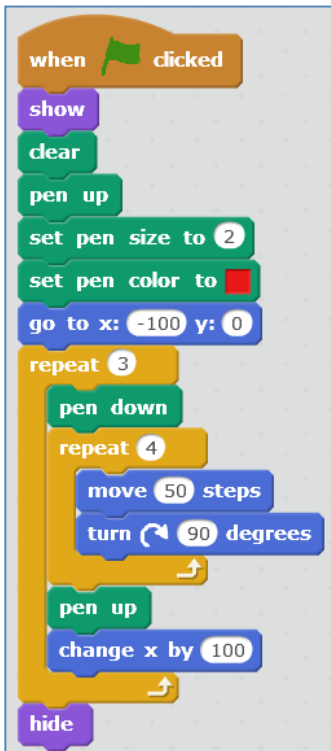
İndi isə kvadrat (bərabər tərəfli düzbucaqlı) çəkən proqram yazaq. Yenə də bu vəsaitə əlavədə verdiyimiz bilgilərə əsasən tam dairə 360 dərəcə olmasını əsas götürsək və dairəni 4 bərabər hissəyə ayırsaq, bu zaman dairəni əhatə edən çevrə üzərində bir-birindən bərabər məsafədə yerləşən 4 nöqtə alacağıq. Bu nöqtələrin birləşməsi bizə bərabərtərəfli düzbucaqlı, yəni kvadrat verəcək. Kvadratın hər tərəfinin dairədən ayırdığı bucaq 90 dərəcə olacaq.



Kvadratı çəkmək proqramı prinsipcə üçbucağı çəkmək proqramına oxşayır. Sadəcə olaraq proqramda daxilə üçbucaq çəkən əmrlərin yerinə kvadrat çəkən əmrlər daxil etmək lazımdır.



İndi isə proqramın tam mətni ilə tanış olaq:



Prinsipcə daxilə kvadrat çəkmək əmrlərindən savayı bütün əmrlər eynidir.

Sadəcə olaraq proqram daxilində konkret fiquru (üçbucaq, kvadrat və s.) çəkən proqram bloku fərqli olur.

5.1.3. İxtiyari çoxbucaqlının çəkilməsi

Üçbucaq və kvadratın çəkilməsi proqramlarının yaradılması zamanı dediklərimizi ümumiləşdirsək görürük ki, bizə birinci növbədə çoxbucaqlı çəkmək üçün tərəflərin sayını bilmək lazımdır. Bəs konkret çoxbucaqlı çəkən zaman dönmə bucağını necə müəyyənləşdirmək olar?

Bildiyimiz kimi tam dairə **360** dərəcədir.

Üçbucaqda tərəflərin sayı **3**, dönmə bucağı isə **120** dərəcədir. **360: 3 = 120**

Kvadratda tərəflərin sayı **4**, dönmə bucağı isə **90** dərəcədir. **360: 4 = 90**

Onda beşbucaqlı üçün dönmə bucağı **72** dərəcə olacaq. **360: 5 = 72**

Altıbucaqlı üçün dönmə bucağı **60** dərəcə olacaq. **360: 6 = 60**

Səkkizbucaqlı üçün dönmə bucağı **45** dərəcə olacaq. **360: 8 = 45**

Beləliklə, deyilənlərdən aydın olur ki, ixtiyari çoxbucaqlı çəkmək üçün proqram yazanda **360** dərəcəni çoxbucaqlının tərəflərinin sayına bölməklə dönmə bucağını tapmaq olar. Məsələn: onbucaqlı fiquru çəkəndə dönmə bucağı belə hesablanacaq:

$$360 : 10 = 36$$

On beş bucaqlı üçün düstur belə olacaq:

$$360 : 15 = 24$$

Dediklərimizi ümumiləşdirsək deyə bilərik ki, tərəflərinin sayı **n** olan çoxbucaqlını çəkərkən dönmə bucağını hesablamaq üçün 360 ədədini çoxbucaqlının sayına bölmək lazımdır.

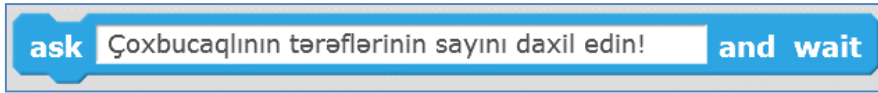
$$\text{Dönmə bucağı} = 360 : \langle \text{tərəflərin sayı} \rangle$$

Qeyd: Çoxbucaqlının tərəflərinin sayı maksimum **360** ola bilər. Tərəflərinin sayı çox olanda alınan fiqur daha çox dairəyə oxşayacaq. Ona görə də tərəflərinin sayını **20**-dən çox götürməyin. Tərəflərin sayı çox olduqca addımların sayını azaldın. Bu proqramın daha effektiv işləməsinə gətirib çıxaracaq.

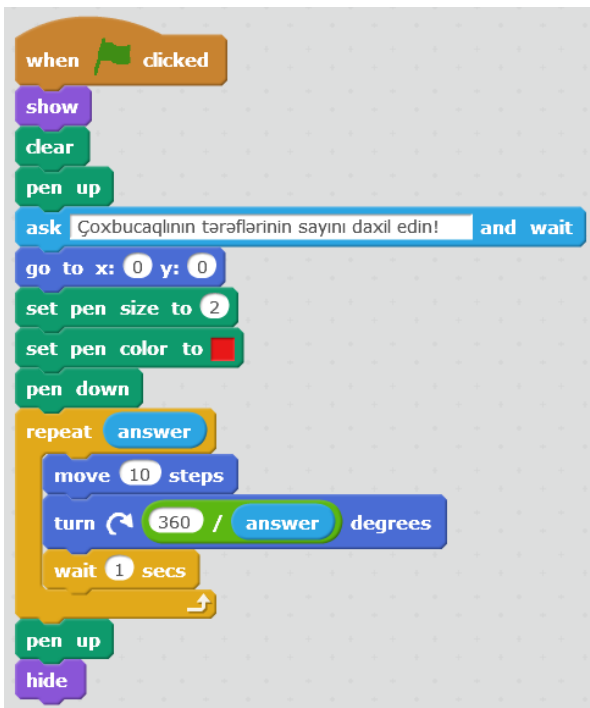
İndi isə gəlin belə bir proqram yazaq. Proqram Sizdən çoxbucaqlının tərəflərinin sayını daxil etməyi soruşacaq. Siz sayı daxil edəndən sonra proqram tərəflərinin sayı həmin sayı uyğun çoxbucaqlını çəkəcək.



Gəlin dediyimiz proqramı yazmaq üçün lazım olan əmrləri seçək. Əmrlərin əksəriyyəti üçbucağın çəkilməsi zamanı istifadə edilən əmrlərdir. Burada yeni istifadə ediləsi iki əmr **Sensor (Sensing)** qrupundakı **ask** və **answer** əmrləridir. Bu iki əmr birgə işləyirlər, daha doğrusu bir-birini tamamlayırlar. **ask** əmrinin daxilində verilmiş mətn səhnədə əks olunur. Bizim nümunədə bu “**Çoxbucaqlının tərəflərinin sayını daxil edin!**” mətnidir.



Bu mətnin əks edilməsi ilə eyni zamanda səhnənin aşağısında mavi rəngli konturla əhatə edilmiş sahə əmələ gələcək. Daxil etmək istədiyimiz informasiya bu sahədən daxil ediləcək. Bəs görəsən daxil edilən yeni informasiya hara yerləşdirilir, daha doğrusu harada yadda saxlanılır? Əlbəttə ki, **answer** adlı sahədə. Bu sahə istifadəçi ilə proqram arasında mübadilə rolunu oynayan aralıq bir yaddaşdır. Biz **answer** adlı yaddaşdan (sahədən) proqramımızda bir dəyişən kimi istifadə edə bilərik. Aşağıda bu proses bir sxem kimi verilib.




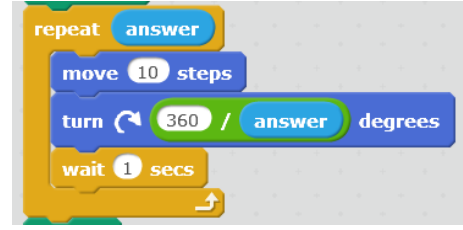
Soldakı şəkildə proqramın mətni verilib.




Birinci dörd əmr bizə əvvəlki proqramlardan məlumdur. Beşinci əmr **ask** əmridir. Bu əmr səhnədə “**Çoxbucaqlının tərəflərinin sayını daxil edin!**” mətnini çap edir və klaviaturdan sayın daxil edilməsini gözləyir. Biz klaviaturdan tərəflərin sayını daxil edəndən sonra növbəti **go to x:0 y:0** əmri spraytı (x:0 y:0) nöqtəsinə daşıyır.

7-9-cu əmrlər qələmin qalınlığı və rəngini seçir və qələmi endirir.


Bizə maraqlı olan hissə proqramın daxili

dövrüdür. Bu dövrdə iki yerdə  əmrindən istifadə edilib. Birincisi **repeat** əmrinin dövrün sayını göstərən hissədir. Üçbucaq çəkəndə bu hissədə **3**, kvadrat çəkəndə isə **4** yazmışdıq.



İndi isə daxil etdiyimiz və  sahəsində yerləşdirilmiş sayı proqramın mətninə salırıq. **move** əmrində addımı **50** götürmüşük.  əmrinin istifadə edildiyi ikinci yer **turn** əmridir. Bu əmrdə dönmə bucağını tapmaq üçün  düsturundan istifadə etmişik.

Bu düstur bizə **Dönmə bucağı = 360 : <tərəflərin sayı>** düsturunun proqramda ifadəsidir.


Biz proqrama daha bir yeni əmr əlavə etmişik. Bu  əmridir. Məsələ bundadır ki, proqram çoxbucaqlının tərəflərini çox sürətlə çəkir. Bu əmr hər tərəfi çəkəndən sonra bir saniyə gözləyir. Nəticədə biz tərəflərin çəkilişini amla görürük.


Sonuncu iki əmr isə qələmi qaldırır və spraytı gizlədir.

İndi Siz bu proqramı yazıb yerinə yetirin və dediklərimizin düzgünlüyünə əmin olun.

Siz sonda proqramın bəzi parametrlərini dəyişin və alınan fərqli nəticələri müqayisə edin.

Məsələn, çoxbucaqlının tərəflərinin sayını **15** və ondan artıq götürəndə görəcəyik ki, çoxbucaqlı istədiyimiz kimi alınmır.

Bu halda  əmrinin parametrini **50** yox, **10** və ya **5** götürmək məsləhətdir.

İndi Siz hətta **60** tərəfli çoxbucaqlı da çəkə biləcəksiniz. Sadəcə olaraq proqramın sürətli işləməsi üçün bu halda  əmrini silin.

Maraqlıdır! 60 tərəfli çoxbucaqlı daha çox dairəyə oxşayır. Bu artıq növbəti dərsemizin mövzusudur.



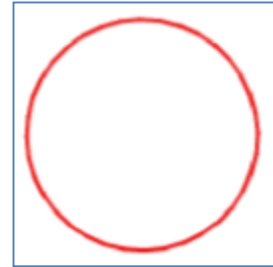
5.2. Dairələr

Əvvəlcə gəlin aydınlaşdıraq görək dairə nədir. Aşağıdakı şəkildə 3, 4, 6, 8, 10, 15, 24 və 36 bucaqlı fiqurların şəkli verilmişdir.



Şəkillərə diqqət etsəniz görərsiniz ki, bucaqların sayı çoxaldıqca alınan fiqurun tərəfinin ölçüsü kiçilir və faktiki tərəf görünməz olur. Bu vəsaitə əlavədə bucaqlar haqqında bilgiler verilib. Həmin məlumata əsaslanaraq deyə bilərik ki, 360 dərəcə bizə tam bir dairəni verir. Deməli, **360** bucaqlı fiqurun tərəfinin ölçüsü çox kiçik olacaq və müəyyən məsafədən baxdıqda onun ölçüsü faktiki görünməyəcək.

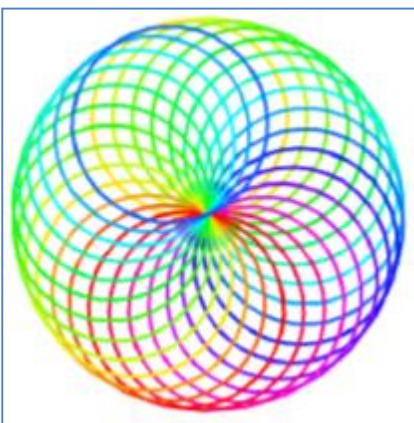
Bizim nümunədə artıq **36** bucaqlı çoxbucaqlı fiquru faktiki dairəyə oxşayır. Deməli dairə çəkmək üçün **36** tərəfli çoxbucaqlı çəkmək kifayətdir.



Sağdakı şəkildə əvvəlki dərstdəki izah etdiyimiz proqram tərəfindən çəkilmiş **36** bucaqlı çoxbucaqlının görünüşü verilib.

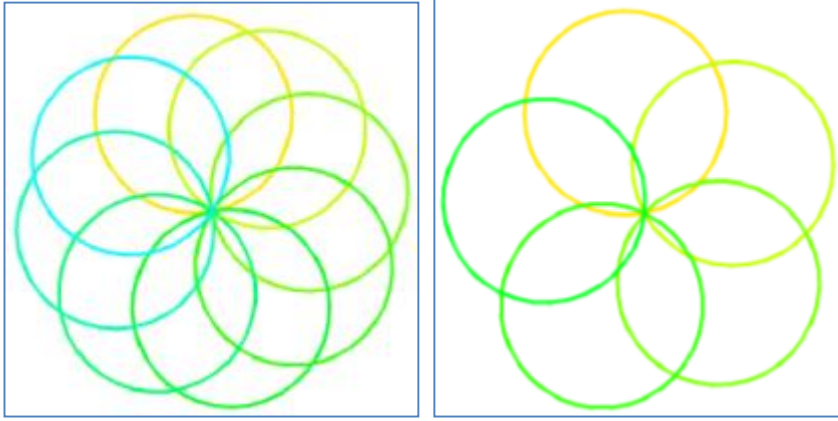
Deməli növbəti dərsimizdə dairələrlə bağlı olan proqramın əsas hissəsi **36** bucaqlı çoxbucaqlının çəkilməsidir.

Bəs görəsən aşağıdakı görüntüdəki fiquru çəkən proqramı necə yazmaq olar?



İlk baxışda çətin görünsə də diqqətlə baxsaq görürük ki, hamısı eyni bir nöqtədə kəsişən müəyyən sayda dairələrin kəsişməsidir.

Aşağıdakı nümunədə səkkiz və dörd ədəd dairədən alınan oxşar fiqura baxaq:



Şekildən görüldüyü kimi bu dairələrin hamısı səhnənin eyni bir nöqtəsində kəsişir. Bu xüsusilə beşbucaqlı fiqurda daha aydın görünür. Kəsişmə üçün ən optimal nöqtə səhnənin mərkəz, yəni $(x:0 y:0)$ nöqtəsidir.

Qeyd ediləsi ikinci fakt odur ki, hər dairənin hüdudları (çevrəsi) müxtəlif rəngdədir. Bildiyimiz kimi **Qələm (Pen)** qrupunda Çoxbucaqlının rənglərini dəyişmək üçün əvvəlcə konkret hansısa rəngi götürmək olar. Bunun üçün

set pen color to

set pen color to 0

əmərlərindən birindən istifadə etmək olar.

Birinci variantda biz bu vəsaitə əlavədə verilmiş təlimata uyğun olaraq istədiyimiz rəngi seçə bilərik. İkinci variantda biz ilkin seçmək istədiyimiz rəngin nömrəsini (kodunu) əmrin daxilində yazsaq bilərik. Əmrin seçimi Sizin öhdənizə düşür. Biz Sizə aşağıda təqdim etdiyimiz proqramda birinci variantı seçmişik.

Sonra dövr daxilində hər dəfə rəngin cari qiymətini dəyişə bilərik. Bunun üçün

change pen color by 10

əmərinə istifadə edə bilərik. Susma halında rəng dəyişmə

addımı 10 götürülür. Amma Siz bu addımı istədiyiniz kimi götürə bilərsiniz.

Sonuncu qeyd isə hər bir çəkilən dairənin ortaq başlanğıc nöqtəsinə nəzərən müxtəlif istiqamətə yönəlməsidir. Bu istiqaməti də aşağıdakı əmr vasitəsilə təyin etmək olar.

turn 360 / answer degrees

Burada **360** tam dövrənin bucaqlarının sayını, **answer** isə klaviaturadan daxil edilən tərəflərin sayını göstərir. Bu ədədləri bir-birinə böləndə istiqaməti bildiren dönmə bucağını alırıq.

İndi isə Sizə təqdim etdiyimiz proqramı araşdıraraq:



```

when green flag clicked
  clear
  pen up
  ask "Dairələrin sayını daxil et!" and wait
  go to x: 0 y: 0
  set pen size to 2
  set pen color to orange
  repeat (answer)
    change pen color by 10
    pen down
    repeat 36
      move 10 steps
      turn 10 degrees
    turn (360 / answer) degrees
    pen up
  
```

Birinci üç əmr əvvəlki proqramlardan bizə məlumdur: Yaşıl bayrağı sıxmaqla proqram başladılır, ekranın köhnə görüntüsü silinir və qələm qaldırılır.

4-cü əmrdə **ask** əmri vasitəsilə dairələrin sayı daxil edilir. Yenə də əvvəlki proqramlardan bildiyimiz kimi, daxil edilən məlumat (bizim halda say) **answer** sahəsində saxlanılır.

go to x:0 y:0 əmri spraytı səhnənin mərkəzinə daşıyır.

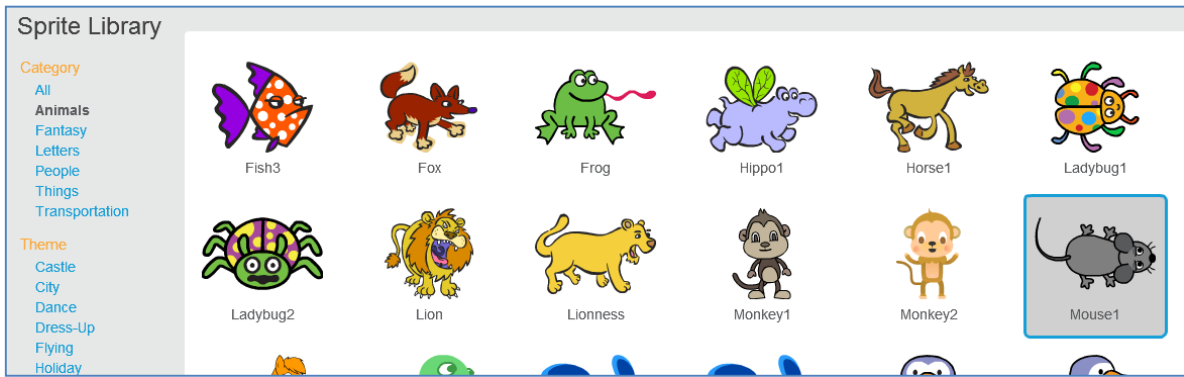
set pen size to 2 və **set pen color by** əmrləri qələmin ölçüsünü və başlanğıc rəngini müəyyənləşdirir.

5.3. Mouse qurğusu ilə manipulyasiya

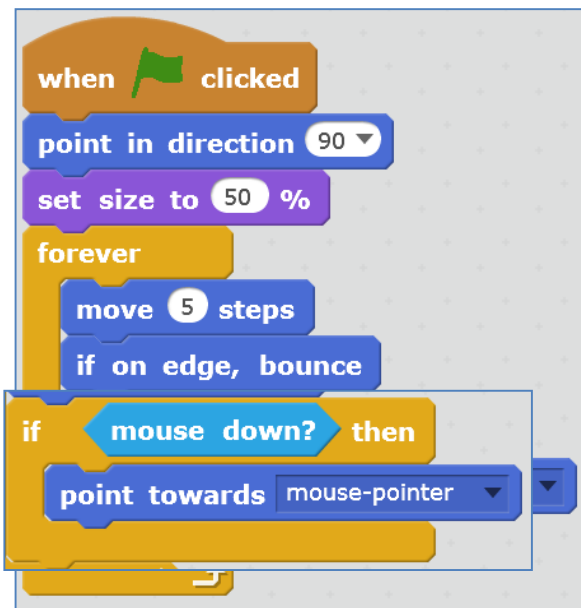
İndi isə gəlin əyləncəli bir proqram yazaq. Təsəvvür edin ki, **Səhnədə** müxtəlif istiqamətlərdə 4 siçan qaçır. Siz mouse (siçan) qurğusunun sol düyməsini **Səhnənin** istənilən yerində sıxanda səs eşidilir və siçanlar həmin istiqamətə qaçırlar.

Belə bir ssenarinin proqramını yazmaq üçün biz aşağıdakı ardıcılığı yerinə yetiririk:

- Pişik personajını silirik.
- Scratch proqramının əsas menyusunun **4** sahəsindən yeni bir personaj (siçan) daxil edirik. Bunun Spraytlar kitabxanasından (Sprite Library) **Animals** qrupundan **Mouse1** spraytını seçə bilərik.



- Mouse1 personajı (spraytı) üçün aşağıdakı proqramı yazırıq:



Birinci əmr **Yaşıl bayrağı sıxanda** əmri proqramı başladır.

İkinci əmr personajın istiqamətini (bizim halda sağa) müəyyənləşdirir.

Üçüncü əmr personajın ölçüsünü 50% (2 dəfə azaltma) edir.

Dördüncü **forever (daima)** əmri öz daxilində 3 əmri idarə edir. Bunlardan birincisi hərəkət sürətini 5 müəyyən edir. İkincisi ekranın kənarı olduqda geriye təkən verir. Üçüncü əmr isə şərt əmri olan

əmdir. Bu əmrin şərt hissəsində **Siçanın sol düyməsi sıxılıbsa**

mouse down?



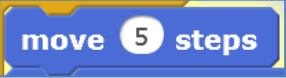
şərt operatoru verilib. Bu o deməkdir ki, Mouse qurğusunun sol düyməsi sıxılında şərt əmrinin daxilindəki əmr yerinə yetiriləcək. Şərt əmrinin daxilindəki əmr isə

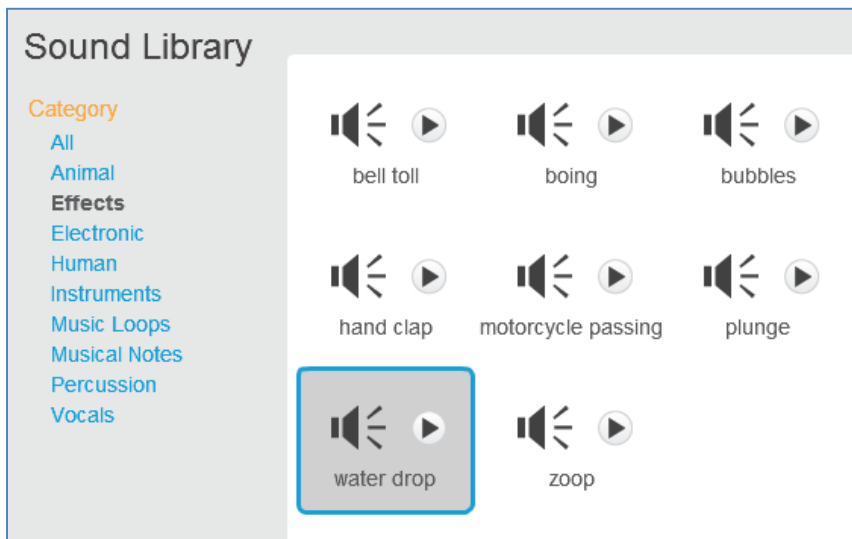


Spraytı kursora yönəldir əmridir. Əgər şərt ödənirsə, onda siçan hərəkət istiqamətini dəyişərək siçanın kursoru istiqamətində hərəkət etməyə başlayır. Aha, istədiyimiz vəziyyət alındı.

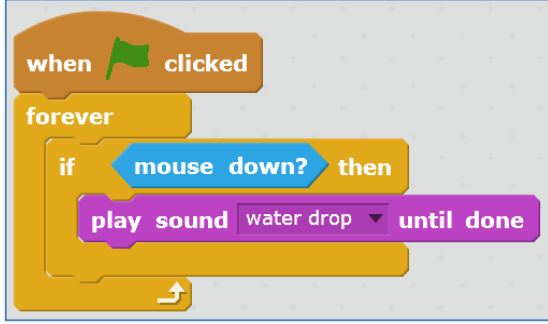
- İndi isə mövcud personajımızı (spraytımızı) 3 ədəd çoxaldırıq. Bunun üçün Scratch proqramının əsas menyusunun **4** sahəsində **Mouse1** spraytının üzərində sağ düyməni sıxıb **duplicate** sətirini seçirik. Bu halda **Mouse2** adlı ikinci bir sprayt yaranacaq. Bu əməliyyatın əsas üstünlüyü ondadır ki, yeni sprayt yaranan zaman əsas spraytın bütün əmrlərini özündə saxlayır.
- Əvvəlki addımı daha 2 dəfə yerinə yetirərək **Mouse3** və **Mouse4** adlı daha iki sprayt yaradırıq.



- Növbəti addımda yeni yaranan 3 spraytda  əmrlərinin parametrlərini uyğun olaraq dəyişirik. Birinci əmr üçün **-90, 0 və 180** dərəcə, ikinci əmr üçün isə **7, 10 və 12** addım qiymətləri müəyyənəndiririk. Hansı sprayt üçün hansı parametrin seçilməsinin fərqi yoxdur.
- Səhnəyə keçirik.
- Səhnə üçün səs effekti əlavə edirik. Məsələn: **Səs Kitabxanasından (Sound Library) Effect** qrupundan **water drop** səsini götürmək olar. Aşağıdakı şəkildə bu proses göstərilib:



- Səhnə üçün aşağıdakı proqramı yazın:



Yaşıl bayrağı sıxanda əmridir.

Növbəti **forever (təkrar et)** əmrinin daxilində şərt əmri gəlir. Bu şərtə görə əgər Mouse (siçan) qurğusunun sol düyməsi sıxılıbsa, onda **water down?** səs faylı səslənəcək.

- Sonda tam ekran rejiminə keçib **Yaşıl Bayraq** düyməsini sıxın və proqramı başladın. Mouse (siçan) qurğusunun sol düyməsini Səhnənin müxtəlif nöqtələrində sıxanda siçanların necə hərəkət edəcəyinə baxın.



6. Səs və musiqi

Bildiyimiz kimi bizi əhatə edən mühit müxtəlif səslərlə boldur. Təbiətin özünün səsi (yağış, axar çay, şələlə, dəniz, külək, ildırım və s.); heyvanların, quşların və digər canlıların səsi, nəqliyyatın səsi, avadanlıqların səsi), müxtəlif proseslərin (xörək bişəndə çıxan səs, çaynik qaynayanda çıxan səs və s.), musiqi alətlərinin səsi, nəğmələr və s. Bu səsləri öz yaranma səbəblərinə görə təbii və süni yaranan səslərə bölə bilərik.

Kompüterdə alınan səslər süni səslərdir. Ancaq təbii səsləri kompüterin yaddaşına yazanda o qədər keyfiyyətlə yazırlar ki, onu təbii səsdən ayırmaq adətən çətin olur.

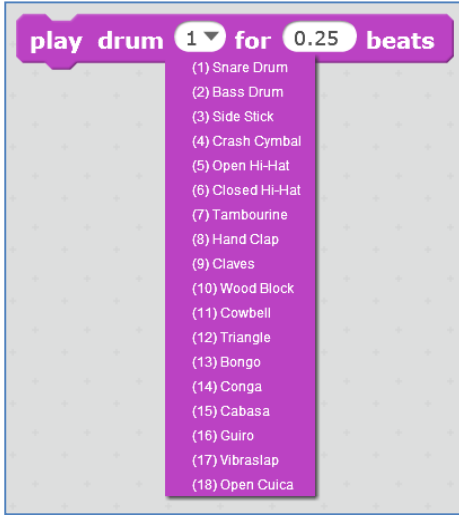
Scratch proqramının səs və musiqi ilə işləmək üçün çox gözəl əmrləri var. Bu əmrlər Səs qrupunda cəmləşiblər. Biz artıq sağda gördüyünüz iki əmrlə tanışıq. Bu əmrlər **play sound** və **play sound** **until done** əmrləridir. İlk proqramımızı yazarkən bu əmrlərdən istifadə etmişik.



Gəlin indi növbəti əmrlərlə tanış olaq.

6.1. Səs və musiqi

İlk tanış olacağımız əmr **stop all sounds** əmridir. Bu əmr işləyəndə hal-hazırda proqramda səsləndirilən bütün səsləri kəsir.



Solda gördüyünüz **play drum** **for** **beats** əmri vasitəsilə 18 növ zərb aləti seçib ifa edilən musiqini istədiyimiz taktı ifa edə bilərik. Əlbəttə ki, azca musiqi təhsili olan şəxs üçün bu terminlər adidir. Bu kitabın əlavəsində belə terminlərin qısa izahı verilib.

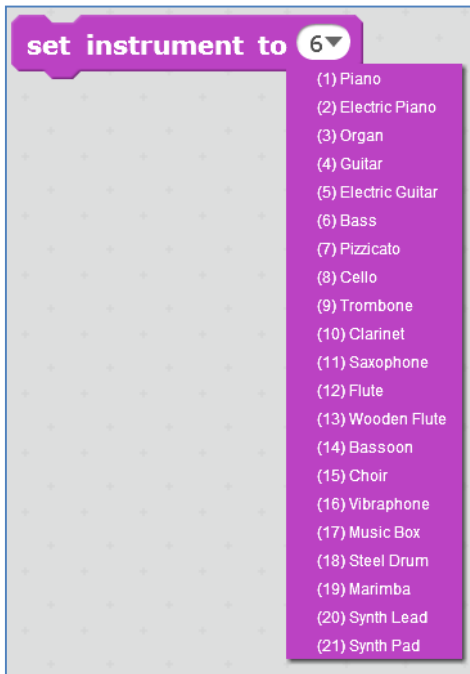


rest for **beats** əmri vasitəsilə musiqinin ifası zamanı göstərilən taktların sayı qədər pauza verir.



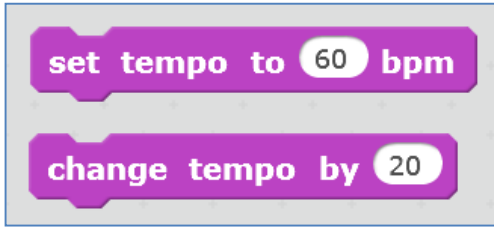
play note **for** **beats** əmri vasitəsilə 2 oktava və 1 not (cəmi 15 not) həcmində piano klavişindən istədiyimiz notu seçib verdiyimiz takt

qədər səsləndirə bilərik.



set instrument to əmri vasitəsilə siyahıda verilmiş 21 alətdən istədiyimizi seçib musiqini həmin alətin səsi ilə səsləndirə bilərik.

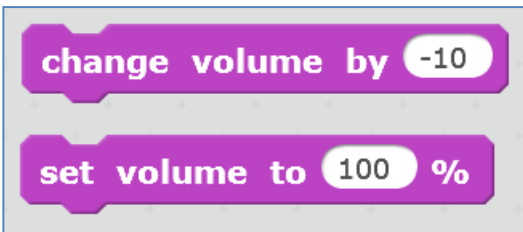
Burada bir şeyi də qeyd edirdik ki, biz bir neçə musiqi alətini seçib onları eyni vaxtda səsləndirərək orkestr, ansambl və ya qrup formada ifa ala bilərik.



Sol tərəfdə gördüyünüz bu iki taktın qiymətini dəyişmək üçündür. Sadəcə olaraq birinci əmrin cari qiymətini verdiyim qiymətlə əvəz edir. İkinci əmr isə tempin cari qiymətini verdiyimiz qiymət qədər dəyişir. Qiymət müsbət olanda artırır, mənfi olanda isə azaldır. Məsələn, tutaq ki, taktın cari qiyməti **30**-dur. Birinci əmr yerinə yetdikdən sonra takt **60** olacaq. İkinci əmr yerinə yetdikdə isə takt **50** ($30+20=50$) olacaq.

6.2. Səsin gücünün dəyişdirilməsi

Biz yazdığımız proqramda elə edə bilərik ki, müəyyən klavişlərin və ya işarələrin (şəkillərin) köməyi ilə səsin gücünü artırıb-azalda bilərik. Gəlin daha iki əmrlə tanış olaq.



Birinci olan **change volume by** əmri vasitəsilə səsin ucalığını artırıb (ədəd müsbət olsa)-azalda (ədəd mənfi olsa) bilərik.

İkinci olan **set volume to** əmri vasitəsilə isə səsin ucalığını konkret faiz həcmində

müəyyən edə bilərik.



Soldakı iki əmr bizə səsin gücünü və musiqinin tempini proqramda müəyyən məqsədlər üçün istifadə etməyə kömək edir. Məsələn, səsin gücü **100%**-dən çox və **0%**-dən az ola bilməz. Biz səsi ucaltmaq və ya azaltmaq üçün istifadə etdiyimiz düyməni və ya klavişi sıxdıqca səsi dəyişmək üçün istifadə edilən qiymət artıb-azalacaq. Müəyyən bir halda bu qiymət **100**-ü aşı və ya **0**-dan aşağı ola bilər. Bu isə real vəziyyətdə ola bilməz. Ona görə də səsin ucalığını düzgün idarə etmək üçün biz bu əmrlərdən istifadə etməklə səsin ucalığını nəzarətdə saxlaya bilərik.

Gələcək misallarda bu dediklərimizin əyani istifadəsini görəcəyik.

6.3. Səslə iş. Misallar

Əvvəlki məşğələlərdə biz artıq **Səs (Sound)** qrupuna daxil olan əmrlərlə tanış olmuşuq. Məsələn, pişik personajı üçün proqram quranda onu həm də səsləndirə bildik. Gələcəkdə biz səs və musiqi ilə bağlı müxtəlif proqram nümunələrinə baxacağıq. Amma növbəti iki məşğələdə notlardan istifadə ilə bağlı proqramları yaratmaqla məşğul olacağıq.

6.4. DO-RE-Mİ – dən “CÜCƏLƏRİM”-ə qədər

Bu vəsaitin əlavəsində musiqi nəzəriyyəsi azacıq məlumat vermişik. İndi gəlin birinci oktavanın notlarını səsləndirən proqram yazaq.



Həmişə olduğu kimi proqramı Yaşıl bayraq düyməsini sıxmaqla başladırıq.

Daha sonra SƏS (Sound) qrupundakı



əmrindən istifadə etməklə uyğun notları səsləndiririk. Soldakı şəkildən görüldüyü kimi biz birinci oktavaya aid **48, 50, 52, 53, 55, 57 və 59** nömrəli notlardan istifadə edəcəyik.

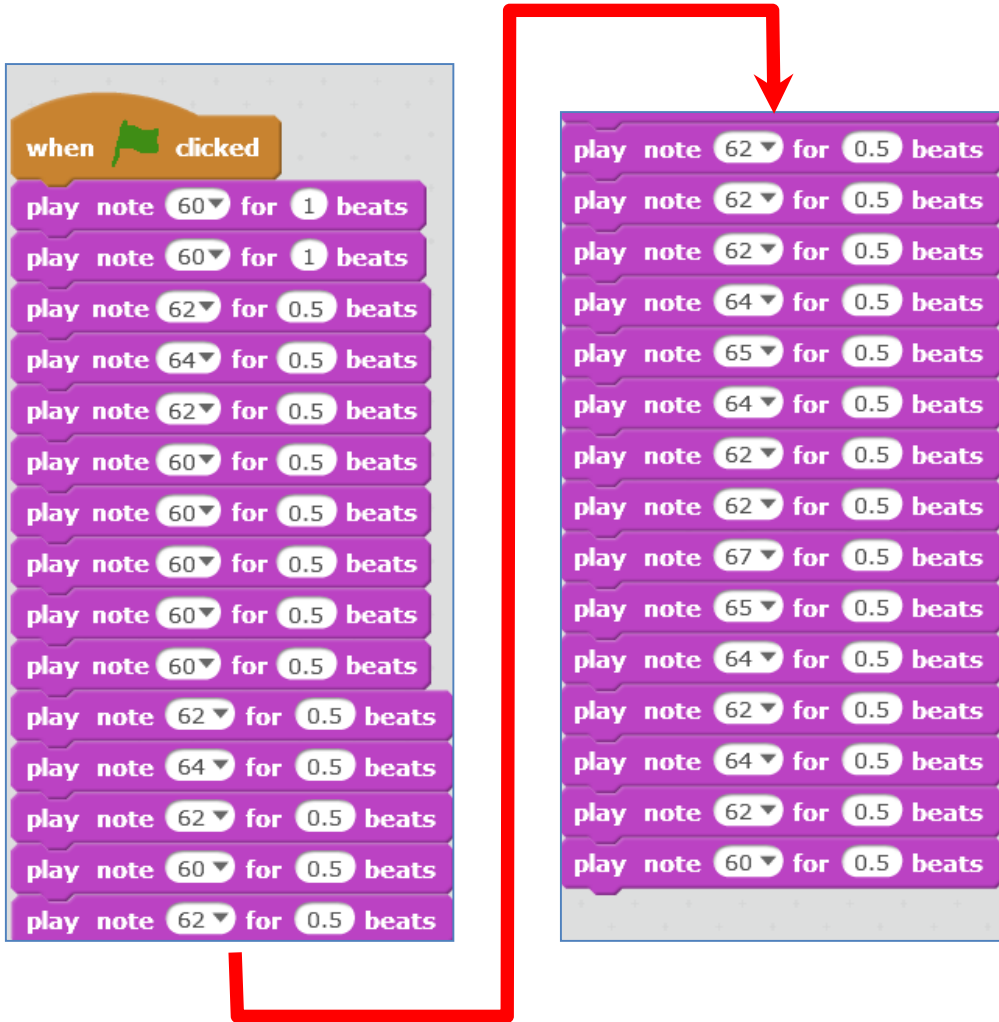
Yaşıl bayraq düyməsini sıxın proqramınızı başladın. Proqram birinci oktavanın yeddi notunu səsləndirəcək.

Bəs görəsən biz məşhur bəstəkarımız Qəmbər Hüseynlinin Dünya uşaq musiqisi albomuna düşmüş “Cücələrım” mahnısının (sözləri istedadlı şairimiz Tofiq Mütəllibova məxsusdur) musiqisindən bir parçanı səsləndirə bilərikmi?





Aşağıdakı şəkildə “Cücələrım” mahnısının musiqisini ifa edən proqramın kodu verilib. Proqramın uzunluğu böyük olduğundan onu iki hissəyə bölmüşük. Amma Siz proqramı yazarkən şəkildəki iki parçanı birləşdirib yazın.



İndi isə **Yaşıl bayraq** düyməsini sıxaraq bu gözəl musiqinin bir parçasını dinləyin.

Belə gözəl musiqini yaradan gözəl sənətkarımıza **ALLAH RƏHMƏT ELƏSİN!**

6.5. “A” = DO “S” = RE “D” = Mİ və sairə

Görəsən klaviaturanın klavişlərini pianonun klavişləri kimi istifadə etmək olarmı?

Scratch-in **Hadisələr (Events)** qrupundakı

when space key pressed

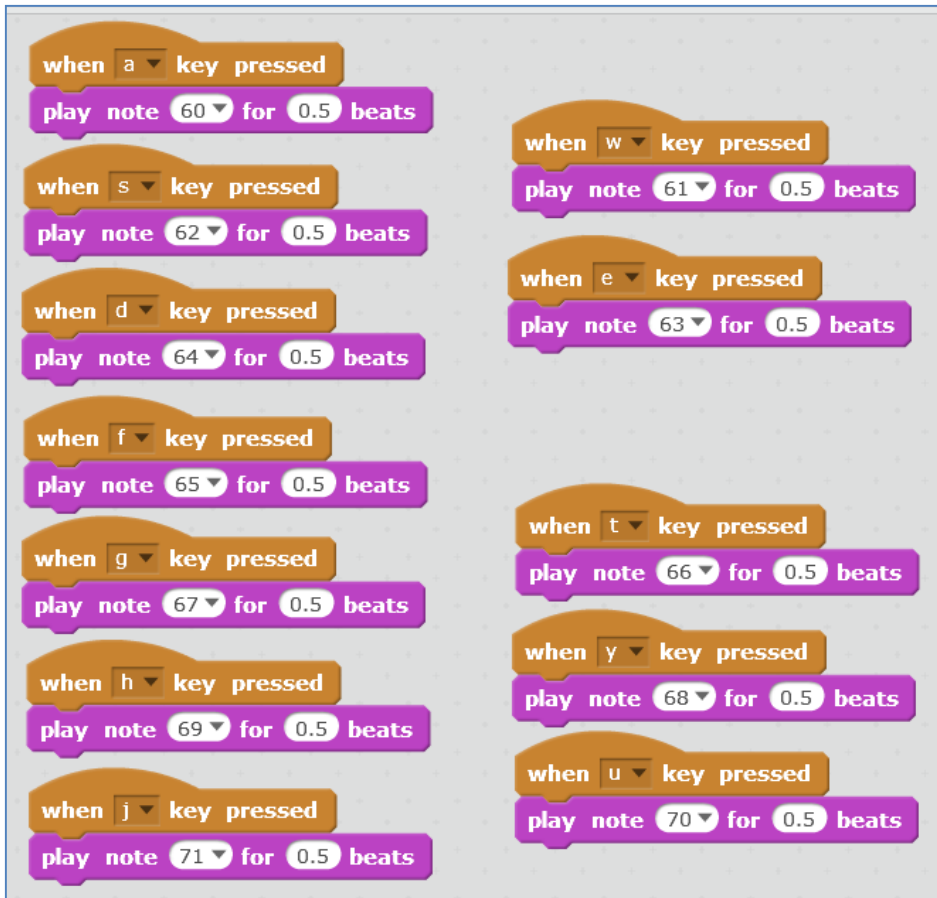
əmrinə və

Səs (Sound) qrupundakı

play note 60 for 0.5 beats

əmrini məqsədimizə çatmaqda

bizə yardımçı olacaq. Artıq bizim bu iki əmrlə işləmək təcrübəmiz olduğundan aşağıdakı şəkildə verilmiş proqramı başa düşmək Sizə çətin olmayacaq.



Şəkildən görüldüyün kimi klaviaturanın klavişləri ilə notlar arasında müəyyən əlaqə qurulmuşdur. Bu əlaqə aşağıdakı cədvəldə öz əksini tapıb:

Klaviş	A	S	D	F	G	H	J	W	E	T	Y	U
Not	DO	RE	Mİ	FA	SOL	LYA	Sİ	DO#	RE#	FA#	SOL#	LYA#
Kod	60	62	64	65	67	69	71	61	63	66	68	70

Bu cədvəldə klaviş sətirində klaviaturanın klavişi, Not sətirində notun adı və Kod sətirində **play note** əmrində notun ikinci oktavadakı kodu göstərilib.



Bu proqramı başlatmaq üçün **Yaşıl bayraq** düyməsini sıxmağa ehtiyac yoxdur. Çünki hər notu səsləndirmək üçün ayrıca klaviş təyin olunub.

İndi Siz klaviaturanın klavişlərindən istifadə etməklə “Cücələrim” mahnısının musiqisini yenidən səsləndirə bilərsiniz.

Əgər Sizin musiqi təhsiliniz varsa, özünüzü gənc bəstəkar kimi də sınağa bilərsiniz.

7. Son söz əvəzi

Əziz oxucumuz, artıq **SCRATCH** proqramı ilə ilk tanışlığın sona çatdı. Sən artıq öz xəyallarını daha yaxşı formada proqram kimi gerçəkləşdirə bilərsən. Sənin uğurun bizim də uğurumuzdur. Biz Sənin uğurlarına Səndən də çox sevinirik.

Bu kitabı oxuyub sona çatanda Sən hiss edəcəksən ki, proqramlaşdırmada ilk uğurlu addımlar atmağa başlamısan. Bizim məqsədimiz də proqramlaşdırmanın Sənin həyatında xüsusi yer tutmasına nail olmaq idi. Sən artıq layihə qura bilərsən, problemi həll edirsən, az da olsa, özünü yaradıcı şəxs kimi ifadə edirsən.

SCRATCH proqramı ilə işləməyi bacarından sonra Sən seçim qarşısında qalırsan. Bu halda ortaya sual çıxır:

GÖRƏSƏN BU QƏDƏR KİFAYƏT EDƏRMİ?

Sən artıq proqramlaşdırmada ilk addımlarını atmısan və bundan sonra növbəti addım atmağa hazırsansa, onda Sənə növbəti kitabımızı təklif edirik. Bu kitab “**Scratch 2 proqramlaşdırma dili**” üzrə kitabın II hissəsidir. Kitabın ikinci hissəsində Sizə daha çətin və maraqlı proqramlar təqdim ediləcək. Kitabı əldə etmək istəyənlər aşağıdakı elektron poçt və ya telefon nömrəsi vasitəsilə bizimlə əlaqə saxlasınlar.

e-mail: abdullaqehreman@gmail.com

Telefon:(+99450) 3503920



8. ƏLAVƏLƏR

8.1. Saniyənin hissələri və ya kəsir

Əgər Siz adi xətkəşi gözünüzün qarşısına gətirsəniz görərsiniz ki, orada böyük bölgülərlə santimetrler göstərilib. Santimetrin bölgüsü də öz növbəsində 10 bərabər hissəyə bölünüb və bu kiçik bölgülər də millimetr adlanır.

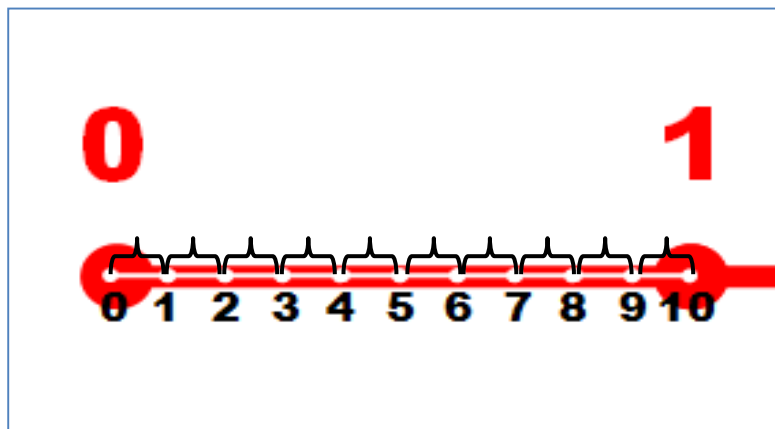


Bunu əsas tutaraq fərz edək ki, biz saniyələri göstərən bir *zaman oxu* çəkmişik.



Bölgülərdə olan qırmızı rəngli böyük ədədlər saniyələri göstərir. Həmin saniyələr arasındakı məsafələri də 10 bərabər hissəyə bölək. Onların da hər birini qara hərflərlə işarələyək. Bizim misalda biz yalnız 1 saniyənin bölgülərini işarələmişik.

Saniyənin bölgüləri olan hissəni böyüdüüb diqqətlə baxaq.



Cəmi 10 hissə var. 10 hissənin hamısını götürsək bu bütöv 1 saniyə edər. Başqa sözlə, on hissə tam bir saniyə edər. Bunu belə işarə edirlər:

1.0

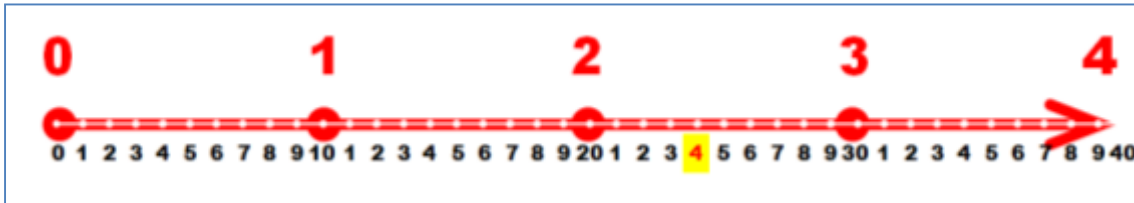
Bu yazılışda “1” ədədi *tam* adlanır. “0” ədədi hissə *kəsr* adlanır. “.” işarəsi tam hissə ilə kəsr hissəni ayırır.

Əgər biz saniyələrin bölgülərindən yalnız 5 hissəni götürsək, bu “*sıfır tam onda 5*” adlanacaq və belə işarə ediləcək: **0.5**

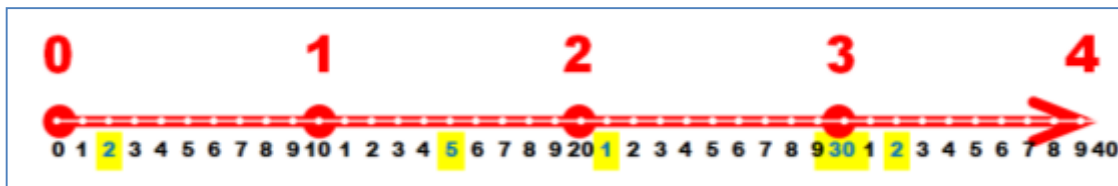
Çalışın bu yazıları oxuyun:



Bəs görəsən **2.4** yazılışı nəyi göstərir və necə oxunacaq? Yazılışdan aydın olur ki, burada tam hissə **2**, kəsr hissə isə **4**-dür. Oxunuş belə olacaq, “*iki tam onda 4*”. Yəni iki tam saniyə və üçüncü saniyənin dörd hissəsi. *Zaman oxunda* bu belə görünəcək (sarı fonda qırmızı yazılış):



İndi isə *zaman oxunda* sarı fonda yaşıl rənglə verilmiş zamanları oxuyun:



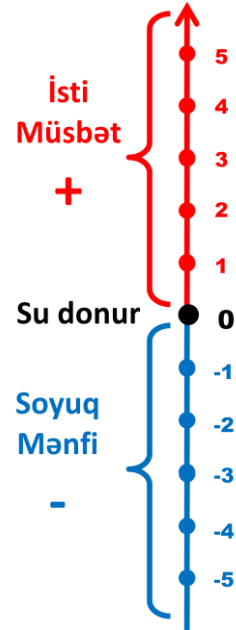
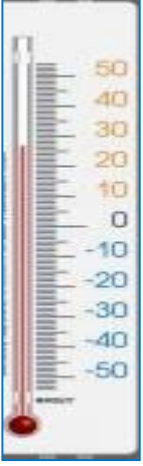
Afərin! Yəqin ki, bir problem olmadı.



8.2. Mənfi ədəd

Riyaziyyatda mənfi ədədlər yuxarı siniflərdə öyrədilir.

Amma real həyatımızda bu ədədlərlə tez-tez rastlaşırıq. Buna ən yaxşı nümunə termometrdir. Termometrin şkalasında orta “0” (sıfır) göstəricisi var. Həmin göstərici termometrin başlanğıc göstəricisidir. Ondan yuxarıda olan ədədlərin qarşısında heçbir işarə yoxdur, aşağıdakıların qarşısında isə minus işarəsi var. Minus işarəsi olmayan göstərici “*müsbət*”, minus işarəsi olan göstərici isə “*mənfi*” göstərici adlanır. Konkret termometr halında müsbət göstərici havanın isti, mənfi göstərici isə soyuq olmasını bildir. Bəs “0” göstərici bunlardan hansına aiddir? Sıfır göstərici bunlardan heç birinə aid deyil, isti ilə soyuq arasında sərhəddir.

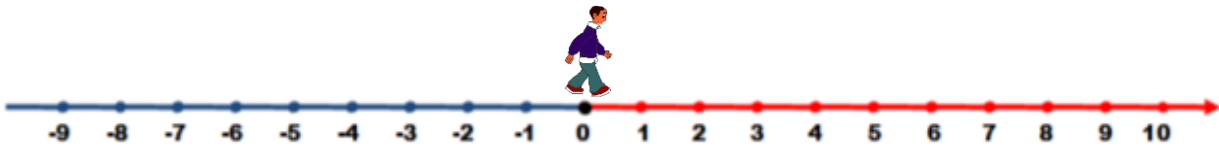


Qarşısında “**minus**” işarəsi olan ədədə *mənfi ədəd* deyilir.





Havanın temperaturu **dərəcə** ilə ölçülür. Termometrdeki göstəricilər dərəcələri göstərir.

Temperatur **sıfır dərəcə** olanda *su donur*.

İndi isə müsbət və mənfi ədədlərlə nümunələrə baxaq. Tutaq ki, ədəd oxunun bir hissəsi verilmişdir. Əlbəttə ki, ədədlərin hamısını ədəd oxunda göstərmək istəsək bunu edə bilmərik. Çünki ədədlərin sayı sonsuzdur. Onlar 1000000, 200, -300, -300000000 ola bilər. Onların hamısını bir vərəqdə göstərə bilmərik. Ona görə də ədəd oxunun hissəsini götürmüşük.



Təsəvvür edək ki, bu ədəd oxunun üzəri ilə üzü sağ tərəfə olan oğlan hərəkət edir. Başqa sözlə oğlan üçün irəli dedikdə sağ tərəf başa düşülür. Əgər oğlan üzünü sol tərəfə çevirsə, onun üçün irəli sol tərəf, yuxarı çevirsə, yuxarı, aşağı çevirsə, aşağı olacaq. Dediklərimiz aşağıdakı şəkildə əyani verilib.

			
→	←	↑	↓

Ədəd oxunda ədədləri toplayanda mənfi ədədləri mütərizədə göstərilir.

Məsələn: (-5).

Toplanan ədədlərin ikisi də müsbət olarsa, onları toplayıb cəmini yazırlar. $5+4=9$

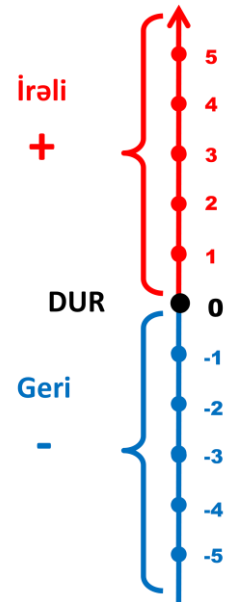
Toplanan ədədlərin ikisi də mənfi olarsa, onları toplayıb qarşısına minus işarəsi qoyub cəmini yazırlar. $(-5) + (-4) = (-9)$

Toplanılardan biri mənfi, digəri isə müsbət olarsa, böyük ədəddən kiçik ədədi çıxıb nəticəni yazırlar. Əgər toplanan ədədlərdən böyüyü müsbətdirsə, qarşısında heç nə yazılmır, mənfidirsə minus işarəsi yazılır.

$$5 + (-3) = 2. \quad 3 + (-5) = -2.$$

Hesab edək ki, oğlan başlanğıcda “0” nöqtəsində üzü sağ tərəfə durub. Aşağıda alınan göstəriciləri ədəd oxunda tapın.

- 3 addım irəli gedəndən sonra oğlan 3 nöqtəsində olacaq.
 $0+30=30$
- 4 addım irəli gedəndən sonra oğlan 7 nöqtəsində olacaq.
 $30+40=70$
- 2 addım geri gedəndən sonra oğlan -5 nöqtəsində olacaq.
 $70+(-20)=50$
- 8 addım geri gedəndən sonra oğlan (-3) nöqtəsində olacaq. **$50+(-80)=(-30)$**





8.3. Dekart koordinat sistemi

Gəlin həyatdan bir nümunəyə baxaq. Tutaq ki, 7 bloklu və 5 mərtəbəli bina mövcuddur. Bu binanın hər blokunda hər mərtəbədə bir mənzil var. Hər mənzildə də bir nəfər yaşayır. Gəlin blokları x , mərtəbələri y ilə işarə edək.

5	Arif	Fərhad	Sona	Xədicə	Səidə	Könül	Zakir
4	Samir	Nigar	Şəhla	Rəşad	Elşad	Aysu	Araz
3	Arifə	Ömər	Leyla	Məhluqə	Zaur	Sərvər	Çiçək
2	Gülər	Vəfa	Mina	Tələt	Qafur	Adilə	Fatma
1	İradə	Sevda	Mehparə	Məryəm	Aynur	Aslan	Mehin
y	x	1	2	3	4	5	6

Şəkildən görüldüyü kimi **Aysu** 6-cı blok, 4-cü mərtəbədə yaşayır. Bunu belə yazıla bilər: $(x: 6, y: 4)$. Burada x blokun nömrəsini, y isə mərtəbənin nömrəsini göstərir. Yəni **Aysu** 6-cı sütunla (blokla) 4-cü sətirin (mərtəbənin) kəsişməsində yaşayır.

Ömərin ünvanı $(x: 2, y: 3)$ olacaq. Yəni 2-ci blok, 3-cü mərtəbə.

3-cü blok 3-cü mərtəbədə Leyla, 5-ci blok 2-ci mərtəbə isə Qafur yaşayır. Uyğun olaraq onların ünvanı $(x: 3, y: 3)$ və $(x: 5, y: 2)$ olacaq. Deməli x və y cütlüyü (blok-mərtəbə) binanın sakinlərinin ünvanlarını işarələməyə imkan verir. x və y dəyişənlərinin hər biri koordinat adlanır. Biri üfüqi (soldan-sağa blokları), digəri isə şaquli (aşağıdan-yuxarı mərtəbələri) koordinatı göstərir.

	Üfqi →					↑
4						Ş a q u l i
3						
2			$(x:3, y:2)$			
1						
y	x	1	2	3	4	

Üfüqi və şaquli koordinatlarla ifadə edilən koordinat sistemi **dekart koordinat sistemi** adlanır. Dekart koordinat sistemi riyaziyyatda geniş istifadə edilir. Əgər diqqətlə baxsaq görərik ki, dekart koordinat sistemi ilə düz səthi (müstəvini) ünvanlamaq olar. Scratch proqramının səhnəsi də bir düz səthdir, yəni müstəvidir.

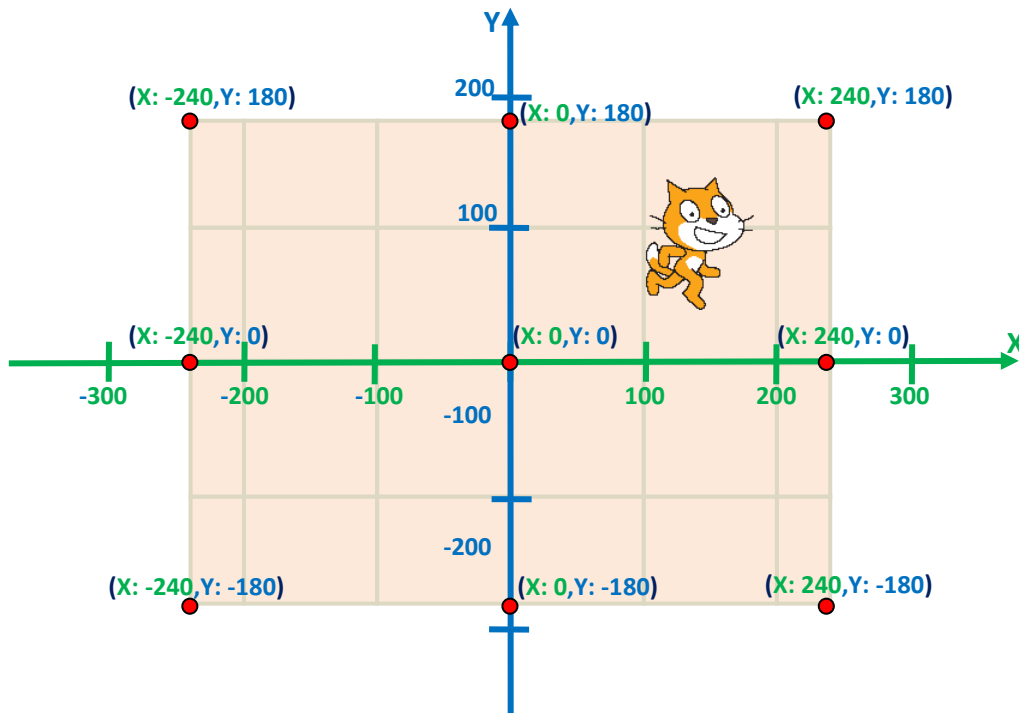
Dekart koordinat sistemində (bizim səhnədə də) başlanğıc nöqtə mərkəz götürülür.

Aşağıdakı şəkildə səhnənin sxemi verilib. Səhnənin mərkəzində həm üfüqi, həm şaquli koordinatların sıfır başlanğıcı göstərilib. Bu nöqtədən keçən iki (biri üfüqi-**X** ilə işarə edilir, digəri isə şaquli-**Y** ilə işarə edilir) ox bir-biri ilə kəsişir. Başlanğıc nöqtənin koordinatı (**X: 0, Y: 0**) ilə işarə edilir.

Biz əvvəlki məşğələlərdə müsbət və mənfi ədədlər barədə danışmışdıq.

Üfüqi **X** oxunda sıfır nöqtəsindən sağdakı koordinatlar müsbət, soldakılar isə mənfi işarə edilir.

Şaquli **Y** oxunda sıfır nöqtəsindən yuxarıdakı koordinatlar müsbət, aşağıdakılar isə mənfi işarə edilir.

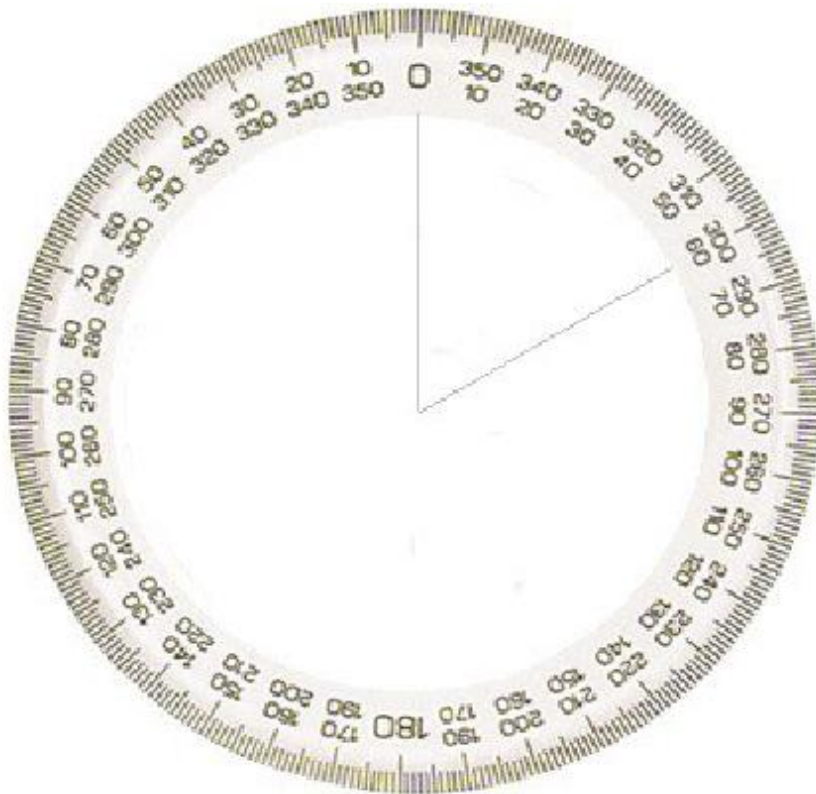


Şəkildə səhnənin bəzi nöqtələrinin koordinatları verilib. Diqqət eləsəniz görərsiniz ki, səhnədə üfüqi 480 (“-240”-la “0” arasında 240 nöqtə və “0”-la “240” arasında 240 nöqtə) koordinat nöqtəsi var. Şaquli isə 360 (“-180”-la “0” arasında 180 nöqtə və “0”-la “180” arasında 180 nöqtə) koordinat nöqtəsi var.



8.4. Bucaq

Yəqin ki, hamımız saat nədir bilirik. Saat dairə formasındadır. Onun ortasında iki əsas əqrəb var: iri və xırda. İri əqrəb dəqiqələri, xırda əqrəb isə saat bildirir. Əqrəblərin yeri müxtəlif olur. Saatın dairəsi 12 bərabər hissəyə bölünüb. Bu da 12 saati bildirir. Hər bir saat arasındakı hissə də öz növbəsində 5 bərabər hissəyə bölünüb. Belə bölgülərin sayı isə 60-dır. Bu isə 1 saatin 60 dəqiqəyə bərabər olmasını göstərir. Əgər bu əqrəbləri bir nöqtədən çıxan işıq şüası kimi qəbul etsək, onda bu iki əqrəb (şüa) mərkəzi dairənin ortasında olan bucaq əmələ gətirir. Bəs görəsən iki şüa arasında müxtəlif bucaqlar nə qədər ola bilər?



Riyaziyyatda dairə 360 bərabər hissəyə bölünür. Hər bir belə hissə *dərəcə* adlanır. Dərəcə “°” ilə işarə edilir. Məsələn: 50° , 35° və s. Beləliklə, dediklərimizdən aydın olur ki, tam dairə 360° (oxunuş: 360 dərəcə) olacaq. Dairənin yarısı 360° olacaq.

Bütöv dairə tam bucaq, dairənin yarısı isə açıq bucaq adlanır. Açıq bucağında yatışı düz bucaq adlanır.



Aşağıdaki şəkildə bucaqların necə işarə edilməsi göstərilib. Bucağın şüaları arasında qövs çəkilir, şüaların başlanma nöqtəsi hərflə (çox vaxt “o”) işarə edilir və qövsün yanında bucağın qiyməti yazılır.



Düz bucaqdan kiçik olan bucağa iti bucaq, düz bucaqdan böyük açıq bucaqdan kiçik olan bucağa kor bucaq deyilir.

Bucaqlar haqqında bu qədər riyazi bilgilərimiz işimizi davam etməyə kifayətdir.



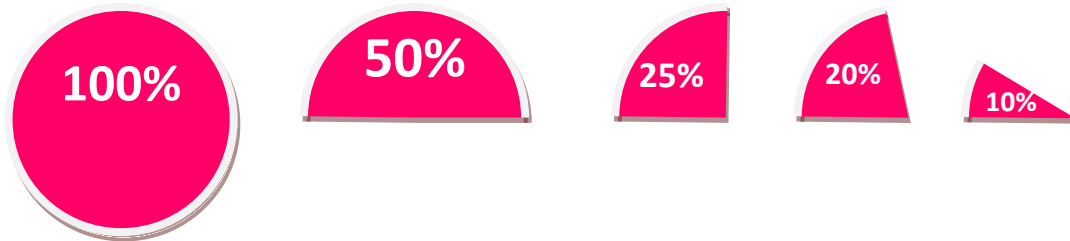


8.5. Faiz nədir?

Əziz dostumuz, Sənin yaşın hələ azdır və faizin nə olduğunu riyaziyyatda keçməmişən. Amma gündəlik həyatda faiz sözünə artıq çox rast gəlmisən. Məsələn, insanlar bankdan pul krediti alırlar və buna görə faiz ödəyirlər. Ya da şagirdlərin cəmi 80 faizi dərstdə iştirak edir. İmtahandan kəsilənlərin sayı 30 faizdir. Belə misalların sayını çoxaltmaq olar. Bəs faiz nədir? Sadə formada bunu necə başa düşmək lazımdır.

Tutaq ki, sinifdəki şagirdlərin hamısı imtahanda iştirak ediblərsə, onda deyirlər ki, bu sinfin şagirdləri imtahanda yüz faiz iştirak ediblər. Bunu belə yazırlar: **100%**. “%” işarəsi “**faiz**” sözünün işarələnməsidir. Dediymizdən aydın olur ki, ilkin nəzərdə tutulanan **100%** yerinə yetirilməsi bu bir tamlıqdır. Onda bütöv tamın yarısı **50%** edəcək. **50%**-in də yarısı **25%** edər. Əgər tamı **5** bərabər hissəyə bölsək, onda **20%** alarıq. Yox, tamı **10** bərabər hissəyə bölsək alınan nəticə **10%** olacaq.

Gəlin dediklərimizə şəkildə baxaq.



Şəkildən aydın olur ki, faiz özü də hissələrə bölmədir. **1%** tamın **100** bərabər hissəyə bölünməsindən alınan hissədir. Beləliklə, faizin nə olmasını bildik.

8.6. Musiqi və notlar

Əziz dostumuz, yəqin ki, piano (royal, fortepiano da eyni quruluşdadır) və onun üzərindəki klavişləri görmüsən. Pianonun 88 klavişi var. Bunlardan 52-si ağ və 36-sı isə qara rəngdədir. Bu klavişlərdən hər birinin üzərinə sıxdıqda müəyyən səs çıxarır. Bu səslər not adlanır. Hər bir klavişin səsi fərqlidir. Soldan sağa getdikcə səslər alçaqdan (bəmdən) yüksəyə (zilə) doğru dəyişir. Yəni iki qonşu klavişlərdən sağdakının səsi daha zildir.



Şəkildən görüldüyü kimi ağ klavişlər heç bir fərq qoyulmadan ardıcıl, qara klavişlər isə iki-iki və ya üç-üç qruplarda yerləşirlər. Klavişlər qruplaşdırılır. Hər qrupda 12 klaviş var. Həmin qruplara **oktava** deyilir.



Şəkildən görüldüyü kimi pianoda cəmi 8 (6 tam və 2 natamam) oktava var. Birinci və sonuncu oktavalar natamamdır. Birinci oktavada 3, sonuncu oktavada isə 1 klaviş var.

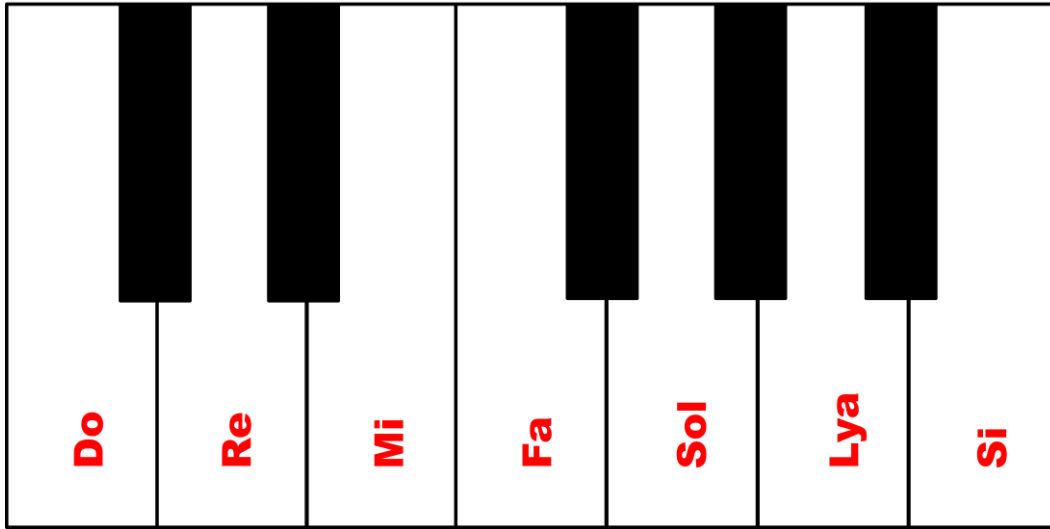
Pianodakı oktavalar soldan sağa ardıcıl olaraq belə adlanırlar:

- Subkontroktava (3 not)
- Kontroktava (12 not)
- Böyük oktava (12 not)
- Kiçik oktava (12 not)
- Birinci oktava (12 not)
- İkinci oktava (12 not)
- Üçüncü oktava (12 not)
- Dördüncü oktava (12 not)
- Beşinci oktava (1 not)



Hər bir oktavada olan ağ klavişlərə uyğun notlar aşağıdakı kimi adlanır:

O K T A V A



Deməli oktava bir **Do** notundan başlayıb digər **Do** notuna kimi olan klavişlərin ardıcılığıdır. Yuxarıda dediyimiz kimi, iki qonşu oktavada yerləşən eyni adlı notlardan (məsələn, **Do** notu) sağdakının səsi daha zildir. Bu **tonallıq**, konkret səsin vəziyyəti **ton** adlanır.

Bəs qara rəngli klavişlərə uyğun notlar necə adlanır? Şəkildən görüldüyü kimi, qara klavişlər iki ağ klavişlərin arasında yerləşir. Bu yerləşmənin özü həmin klavişin nə üçün istifadə edilməsini bildirir.

Alterasiya - Latın sözü olan "**alterare**" - "**dəyişiklik**" deməkdir. Musiqi aləmində alterasiya səsin **zilləşdirilməsinə** və ya **bəmləşdirilməsinə** deyilir. İstənilən səsi yarım ton və ya bir ton zilləşdirmək, həmçinin yarım ton və ya bir ton bəmləşdirmək istədikdə alterasiya işarələrindən istifadə edilir:

- **diyez** - səsi yarım ton yüksəldir (zilləşdirir);

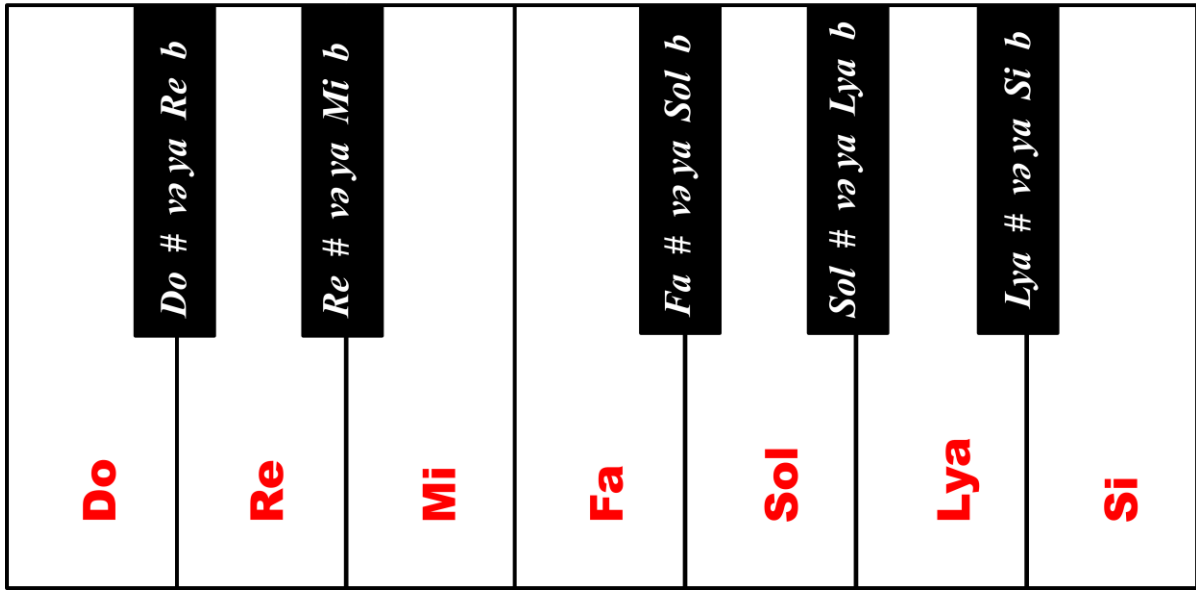
b - **dubl-diyez** - səsi bir ton yüksəldir;

b - **bemol** - səsi yarım ton alçaldır (bəmləşdirir);

b - **dubl - bemol** - səsi bir ton alçaldır.

əgər alterasiyaya uğramış səsdən sonra səsin öz natural halına qayıdırsa, onda bekar - n (və ya n n - dubl - bekar) işarəsi işlənir. Bekar digər alterasiya işarələrini ləğv edərək əsas səsi bərpa edir. (mənbə: <http://www.musigi-dunya.az/magazine1/articles/Lugat/LugPage3.html> Web-səhifəsi).

O K T A V A



Allegro - İtalyan sözü olan "Allegro" - "tez, "iti", "canlı" kimi mənalarda işlənir. Musiqidə isə Allegro - ən geniş yayılmış tez templərdən biridir. XVIII-XIX əsrlərdə bəstəkarlar adətən, sonata və simfoniyaaların birinci hissəsini bu tempdə bəstələyirdilər. Belə hissələrin yazıldığı forma da elə "sonata-allegrosu" adı almışdır. Məsələn, Haydn, Motsart, Bethovenin sonata və simfoniyaalarının birinci hissələri sonata-allegrosu formasında yazılmışdır.

Alt - Bəlkə də artıq sizə məlumdur ki, qalın səslənən oğlan və ya qız səsləri "**alt**" adlanır. Onun diapozonu kiçik oktavanın "**lya**" səmindən ikinci oktavanın "**mi b**" səsinədək uzanır. Ən qalın qadın səsi olan **kontralto** əvvəllər "**alt**" adlanırdı. Sonralar isə bu adı simli alətlər qrupundan olan alət - **alt** (və ya "viola" deyilir) daşımağa başladı. Bəlkə də bu aləti ona görə belə adlandırmışlar ki, onun çox yumşaq, ifadəli insan səsinə çox yaxın olan tembri (səs çaları) var.

Alterasiya işarələri açardan sonra yazılırsa, ona "açar işarələri", ayrıca bir notun qarşısında yazıldıqda isə "təsadüfi işarələr" deyilir.



İstinadlar

1. Golikov Denis & Golikov Artem. Scratch 2.0 Programming. Making games and cartoons. Detailed step by step guide for children. Copyright Golikov Denis & Golikov Artem, 2015
2. Голиков Денис и Голиков Артём. Программирование на Scratch 2. Часть 1.
3. Голиков Денис и Голиков Артём. Программирование на Scratch 2. Часть 2.
4. Голиков Денис и Голиков Артём. Книга юных программистов на Scratch 1.4. Вторая редакция. Copyright Голиков Денис и Голиков Артём, 2014.
5. Еремин Е.А. Газета «Информатика». Среда Scratch – первое знакомство. – М.: Первое сентября, 2008 – №20 (573) – С. 16–28.
6. <https://scratch.mit.edu/> Scratch proqramının rəsmi saytı
7. <http://letopisi.ru/index.php/Скретч> - Скретч в Летописи.ру
8. <http://megogo.net/ru/news/view/1562261-kogda-mozhno-nachinat-obucheniyu-programmirovaniyu.html>
9. <http://muz-teoretik.ru/pauzy-v-muzyke/>
10. <http://muz-teoretik.ru/raspolozhenie-not-na-klaviature-pianino/>
11. <http://nbspace.ru/math/>
12. <http://scratch.ucoz.net>
13. <http://scratch.uvk6.info/> Scratch proqramı üzrə sayt
14. <http://scratch4russia.com/store/>
15. <http://setilab.ru/scratch/category/commun> - Scratch öyrənin.
16. <http://www.7not.ru/theory/01.phtml>
17. http://www.iteach.ru/exp/articles.php?mpt_id_text=115
18. <http://www.webplanet.ru/review/entertainment/2008/01/22/scratch21.html>
19. http://zarapina.blogspot.com/p/scratch_2116.html
20. https://wiki.scratch.mit.edu/wiki/Scratch_2.0

Abdulla Qəhrəmanov

Abdulla Qəhrəmanov 1978-ci ildə BDU-nun Tətbiqi riyaziyyat fakültəsini bitirmişdir. Dövlət Statistika Komitəsində, Maliyyə Nazirliyi, Az. DETK Bakı ETM-də proqramlaşdırma və tədris şöbələrinə rəhbərlik etmişdir. Hal-hazırda informatika fənnindən dərs deyib. Kurikulum üzrə təlimçidir. 10 vəsaitin 200-dən artıq elektron resursun, 3 səmərələşdirici təklifin müəllifidir.

Sevda Sadıqova

Sevda Sadıqova 1994-cü ildə BDU-nun Tətbiqi riyaziyyat fakültəsini bitirmişdir. Bakı ş. 167 №-li tam orta məktəbdə informatika fənnini tədris edir. Son illər İT sahəsində araşdırmalar aparır. Avstriyada, Türkiyədə keçirilən beynəlxalq konfranslarda öz məqalələri ilə iştirak etmişdir.

Dilarə Hacıyeva

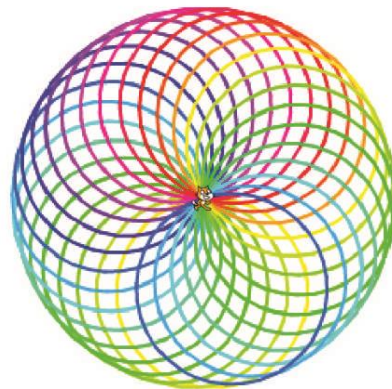
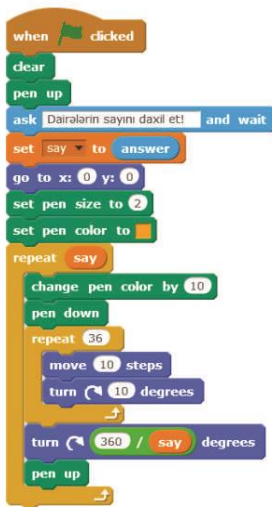
Dilarə Hacıyeva 1997-ci ildə ADPU-ni bitirmişdir. Həmin ildən Bakı ş. 32 №-li məktəbdə ibtidai sinif müəllimi işləyir. 2 dəfə Elektron təhsil müsabiqəsinin nominantı olmuşdur. Kiçik yaşlı məktəblilərin proqramlaşdırma ilə məşğul olması üzrə məktəb səviyyəsində araşdırmalar aparır.

Könül Rəcəbova

Könül Rəcəbova 20 ildən çoxdur ki, İnformatika fənnini tədris edir. Alqoritmləşdirmə və proqramlaşdırma ilə xüsusilə maraqlanır. Çoxlu sayda elektron vəsaitin müəllifidir. Bir çox təhsil sərgilərinin iştirakçısıdır.

İlahə Cəfərova

İlahə Cəfərova 2003-cü ildə ADNA-nın Cihazqayırma fakültəsini İnformasiya-ölçmə texnikası və texnologiyası ixtisası üzrə bitirmişdir. 2008-ci ildən Bakı ş. 23 №-li məktəbdə İnformatika fənnini tədris edir. “Python proqramlaşdırma dili” kitabının (2015-ci il) həmmüəllifi, 5 vəsaitin, 30-dan çox elektron resursun müəllifidir. Bir çox təhsil sərgilərinin iştirakçısıdır.



Qiyməti razılaşma yolu ilə