

5.1.1 Kuchuk va mushuk Salom aytadi¹ - Ketma-ketlik

Oldingi bobda biz Entry blokli kodlash va Pygame Zero-da kodlashning asosiy tuzilish farqini tushunib oldik. Shu asosda, Entry asosiy kitobidan yettita misolni Entry-Python emas, balki haqiqiy Python kodlash orqali yaratib ko'ramiz. Ammo buni amalga oshirish uchun faqat Pygame Zero kutubxonasi funksional jihatdan yetarli emas. Shuning uchun qo'shimcha foydalanuvchi kutubxonasi kerak bo'ladi, va biz ushbu foydalanuvchi pgzhelper kutubxonasi (user library, ya'ni odatda oldindan o'rnatilmagan va ehtiyojga ko'ra foydalanuvchi tomonidan yaratilgan kutubxona) chaqirishdan boshlaymiz.

{% hint style="info" %} pgzhelper birinchi bo'lib **Cort**, Singapurdagi dasturchi, Scratch blokli kodlashni PygameZero-ga o'tkazishda(porting) PygameZero cheklovlari tufayli yuzaga kelgan qiyinchiliklarni(o'xshash funksiyalarni bajarishni) yaxshilash uchun yaratgan. Shunga o'xshash sabablarga ko'ra **Roboticsware** ham PygameZero-ga Entry bloklarini kodga aylantirishni osonlashtirish uchun mavjud kutubxonaning funksiyalarini to'ldirdi va buni ochiq kodlar GitHub sayti (<https://github.com/roboticsware/pgzhelper>) orqali ommaga tarqatmoqda. {% endhint %}

```
from pgzhelper import *
```

Kodning birinchi qatori pgzhelper kutubxonasini chaqiradigan qismidir. Kutubxonani chaqirishning ikki asosiy usuli mavjud. Odatda siz *import pgzhelper* deb ishlatishga odatlangan bo'lsangiz kerak, lekin bu holda kutubxonani yangi sintaksis yordamida chaqirish kerak bo'ladi.

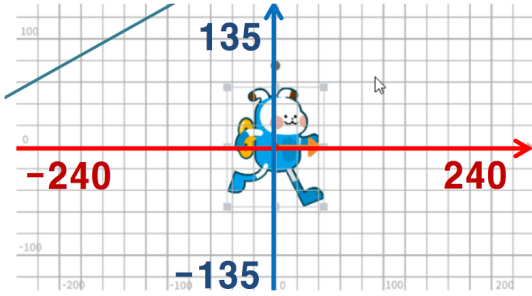
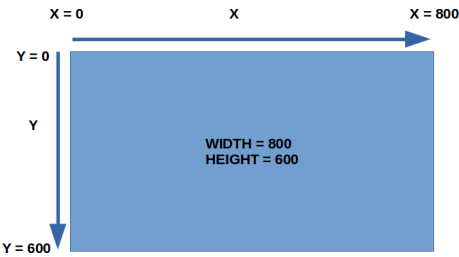
```
from kutubxona nomi import *
```

Uni shunday chaqirishning afzalligi **avval siz kutubxona ichidagi funktsiyani chaqirish uchun doimo <<kutubxona nomi.funksiya nomi>> sintaksisidan foydalanishingiz kerak bo'lgan bo'lsa, endi siz uni avvalgidek, xuddi shu tarzda ishlatishingiz mumkin, faqat funktsiya nomi yordamida, old tomonda kutubxona nomi ko'rsatmasdan.**

Entry koordinatalari tizimi va Pygame koordinatalari tizimini taqqoslash

Biz yaxshi bilganimizdek, Entry-dagi koordinatalar tizimi quyidagi chapdagi rasmda ko'rsatilgan va uning xususiyati shundaki, ekranning belgilangan o'lchami (kengligi 480 piksel, balandligi 270 piksel) bor va ekranning markazi x va y koordinatalarining boshlanish nuqtasi (0, 0) hisoblanadi. Boshqa tomondan, **Pygame koordinatalar tizimida ekranning umumiy o'lchamini foydalanuvchi o'zi belgilashi mumkin, ammo bu holda koordinatalarning boshlanish nuqtasi (0, 0) yuqori chap burchakda joylashgan bo'ladi va o'ngga va pastga qarab qiymatlar oshib boradi.**

Entry	Pygame	
-------	--------	--

Entry	Pygame	
		

Keling, hozircha Entry ning bajarish ekrani hajmini taqlid qilaylik. Shuning uchun ekranning kengligi va balandligi mos ravishda 480 va 270 pikselni tashkil qiladi va shunga asoslanib, kod quyidagicha bo'ladi:

```
{% code lineNumbers="true" %}
```

```
from pgzhelper import *

TITLE = 'Kuchuk va Mushuk salom aytadi'
WIDTH = 480
HEIGHT = 270
```

```
{% endcode %}
```

O'yinda alohida fon rasmi mavjud emas. Entry'da fon rangi avtomatik ravishda oq rangda bo'ladi, ammo Pygame Zero-da bu avtomatik ravishda bo'lib o'tmaydi. Shuning uchun, bizning ehtiyojimizga ko'ra fonni oq rangda bo'yash kerak bo'ladi. Buni kodlashning uchta usuli mavjud:

Oldindan belgilangan rang qiymati kalit so'zlari yordamida kodlash

```
def draw():
    screen.fill('white')
```

draw funksiyasi oldingi bobda ko'rganimizdek, Pygame Zero o'yin tsiklida mavjud bo'lgan oldindan belgilangan qayta chaqirish qilish funksiyasi. Ekranda biror narsa chizmoqchi bo'lganimizda, u avtomatik ravishda chaqiriladi. Ekranni rangga to'ldirish uchun kod ushbu funktsiya ichida yozilishi kerak va ekranni to'ldirish uchun biz oq rangni argument sifatida o'tkazib, **fill** deb nomlangan funktsiyadan foydalanamiz. **fill** funksiyasi **screen** yordamida chaqiriladi, chunki **fill** funksiyasi Pygame Zero kutubxonasining **screen** ob'ektida mavjud.

Bu qism Entry-Python kitobida qoplangan edi, shuning uchun sizdan buni yaxshi tushunish kutilmoqda.

Ekranni to'ldirish uchun rang qiymatini o'tkazishning uchta usuli mavjud. Ulardan biri til o'rnatilgan kalit so'zlar kabi oldindan belgilangan qiymatlardan foydalanish, ikkinchisi esa RGB vakolatxonasidan foydalanishdir. Oldindan belgilangan rang kalit so'zlarini quyidagi rang qiymatlari jadvalida topish mumkin.

CSS Colors

black	bisque	forestgreen	slategrey
dimgray	darkorange	limegreen	lightsteelblue
dimgray	darkorange	darkgreen	cornflowerblue
gray	darkorange	green	royalblue
gray	darkorange	lime	ghostwhite
darkgray	darkorange	seagreen	lavender
darkgray	darkorange	mediumseagreen	midnightblue
silver	darkorange	springgreen	navy
lightgray	darkorange	mintcream	darkblue
lightgray	darkorange	mediumspringgreen	mediumblue
gainsboro	darkorange	mediumaquamarine	blue
whitesmoke	darkorange	aquamarine	slateblue
white	darkorange	turquoise	darkslateblue
snow	darkgoldenrod	lightseagreen	mediumslateblue
rosybrown	goldenrod	mediumturquoise	mediumpurple
lightcoral	cornsilk	azure	rebeccapurple
indianred	gold	lightcyan	blueviolet
brown	lemonchiffon	paleturquoise	indigo
firebrick	khaki	darkslategray	darkorchid
maroon	palegoldenrod	darkslategrey	darkviolet
darkred	darkkhaki	teal	mediumorchid
red	ivory	darkcyan	thistle
mistyrose	beige	aqua	plum
salmon	lightyellow	cyan	violet
tomato	lightgoldenrodyellow	darkturquoise	purple
darksalmon	olive	cadetblue	darkmagenta
coral	yellow	powderblue	fuchsia
orangered	olivedrab	lightblue	magenta
lightsalmon	yellowgreen	deebskyblue	orchid
sienna	darkolivegreen	skyblue	mediumvioletred
seashell	greenyellow	lightskyblue	deeppink
chocolate	chartreuse	steelblue	hotpink
saddlebrown	lawngreen	aliceblue	lavenderblush
sandybrown	honeydew	dodgerblue	palevioletred
peachpuff	darkseagreen	lightslategray	crimson
peru	palegreen	lightslategrey	pink
linen	lightgreen	slategray	lightpink

Keyin rangni RGB ning uchta asosiy rangi (qizil, yashil, ko'k) kombinatsiyasi orqali ifodalash mumkin, bu yerda har bir asosiy rang maksimal qiymati 256 gacha bo'lishi mumkin. Istalgan rangni ifodalash uchun har bir asosiy rangni o'nlik (0 dan 255 gacha) yoki o'n oltili (00 dan FF gacha) sanoq sistemasida ifodalash usullari mavjud.

CSS RGB COLORS

RED : #FF0000 or rgb(255,0,0)
 GREEN : #00FF00 or rgb(0,255,0)
 BLUE : #0000FF or rgb(0,0,255)

tutorial.techaltum.com

HEXADECIMAL COLOR CODES

Red Color Code Green Color Code Blue Color Code

RR GG BB

Signifies Hex Color Code

Each of the two-digit color codes are in the range 00...FF hexadecimal

EXAMPLES

#FFCC99 #3C00A6

RGB rang qiymatlarining o'nlik ko'rinishida oq rangni kodlash

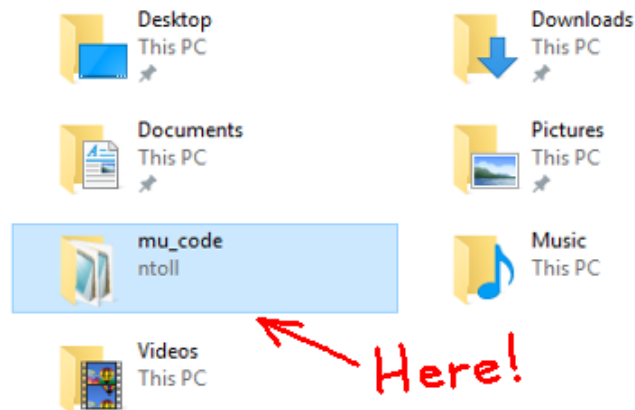
```
def draw():
    screen.fill((255, 255, 255))
```

RGB rang qiymatlarining o'nlik oltilik ko'rinishida oq rangni kodlash

```
def draw():
    screen.fill('#FFFFFF')
```

Ranglarning ifodasiga kelsak, internetda juda ko'p manbalar mavjud, shuning uchun har bir kishi ushbu materiallar bilan to'g'ridan-to'g'ri tanishib chiqishi yaxshiroqdir. Ma'lumot uchun, nega ranglarni ifodalash uchun RGB ning uchta asosiy ranglari ishlatilishi [haqidagi blogni](#) tekshiring.

Keling, shu vaqtgacha yozilgan kodni ishga tushirib, u kutilganidek ishlashini tekshirib ko'raylik. Ishga tushirish uchun avvalo dastur kodini faylga saqlash kerak. Dastur kodi (source code) saqlanishi lozim bo'lgan asosiy joy — bu **[foydalanuvchi nomi]mu_code**. (quyidagi rasmga qarang).



{% hint style="info" %} **Yozilgan dastur kodini saqlaganda, saqlash joyi, albatta, qat'iy belgilanmagan bo'lib, foydalanuvchi uni istagan joyiga saqlashi mumkin. Biroq, ushbu kitobdagi misol amaliyotlarini bajarishda kerak bo'ladigan resurslar (rasmlar, shriftlar, effekt ovozlari, fon musiqasi)dan foydalanish uchun fayllarni yuqorida aytib o'tilgan Mu Editor-ning asosiy kod saqlash joyi ([foydalanuvchi nomi]mu_code)ga saqlashingizni so'raymiz.** Nega bunday qilish kerakligi haqida keyingi bo'limlarda batafsil tushuntiriladi. {% endhint %}

Dastur kodini saqlab bo'lganingizdan so'ng, endi F5 tugmasini bosib, uni ishga tushiring. Ekran oq rang bilan to'g'ri bo'yalgan bo'ladi. Ekran rangini yuqorida o'rgangan uchta usulni qo'llab o'zgartirib ko'rish sizga amaliyot uchun topshiriq sifatida qoldiriladi.

{% code lineNumbers="true" %}

```
from pgzhelper import *

TITLE = 'Kuchuk va mushuk Salom aytadi'
WIDTH = 480
HEIGHT = 270

def draw():
    screen.fill('white')
```

{% endcode %}

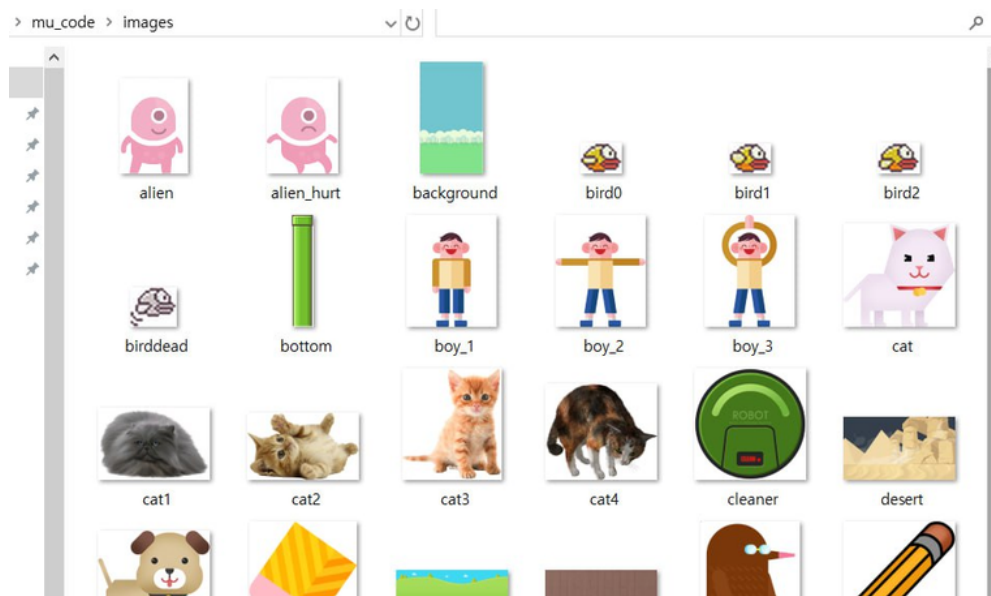
Endi keyingi qadam — o'yin qahramonlari bo'lgan «Kuchukcha» va «Mushukcha» obyektlarini chaqirib, ularni ekranga joylashtirishdir. Buni kodlash uchun har bir obyektning tashqi ko'rinishini ifodalovchi rasm fayllari kerak bo'ladi.

Entry blokli kodlashdagi tajribangizni eslang. O'yinni yaratishda birinchi qilgan ishingiz nima edi? Ehtimol, siz o'yinda ishtirok etadigan obyektlarni ekranga joylashtirish bilan boshlagansiz. Matnli kodlashda ham shunga o'xshash. Biz ham qahramon obyektlarini ekranga chiqarishdan boshlaymiz.

Biroq, Entry-da bo'lgani kabi, agar o'yinda foydalaniladigan obyektlar dasturga oldindan kiritilmagan bo'lsa, o'yin uchun kerak bo'lgan rasm fayllarini kiritib, keyin kodlashni boshlashingiz kerak edi. Xuddi shunday, agar kodlashda foydalaniladigan rasmlar oldindan kiritilgan rasmlar bo'lmasa, ularni alohida saqlash kerak bo'ladi. Saqlash yo'li — bu Mu Editori o'rnatilganda avtomatik ravishda yaratiladigan **[foydalanuvchi nomi]\mu_codeimages** papkasi (direktoriyasi) va rasmlar shu papkada joylashishi kerak bo'ladi.

Ushbu papkaning joylashuvini osongina topmoqchi bo'lsangiz, Mu muharriridagi menyu panelidagi **"Rasmlar"** tugmasini bosing. Bu tugma siz foydalanayotgan rasmlar saqlangan papkani darhol ochib beradi. Papka ichida ushbu kitobda foydalaniladigan rasmlar sizning qulayligingiz uchun oldindan saqlanganini ko'rishingiz mumkin. Manba kodini asosiy saqlash joyiga saqlashingizni so'raganimizning sababi, biz ushbu oldindan saqlangan rasmlardan foydalanib, qo'shimcha resurslar bilan bog'liq ishlarni chetlab o'tib, faqat kodlashga e'tibor qaratishingiz uchun edi.





{% hint style="info" %} Aytgancha, o'yinni dasturlashda zarur bo'lgan resurslarni (rasmlar, shriftlar, ovoz effektlari, fon musiqasi) saqlaydigan papkalarni yaratish uchun siz Pygame Zero kutubxonasining asosiy cheklovlarida o'rnatilgan ushbu pastki papkalar uchun oldindan belgilangan nomlarga (tartibda: rasmlar, shriftlar, tovushlar, musiqa) rioya qilishingiz kerak. Ushbu muvofiqlik Mu editordagi menyu tugmalarining nomlari aytib o'tilgan papkalar nomlariga mos kelishi bilan bog'liq. Shuning uchun, tegishli tugmalarni bosish orqali resurs turiga mos papkalarni avtomatik ravishda ochishga qaratilgan dasturlash, resurslarni topa olmaslik bilan bog'liq ish vaqtida xatolar ehtimolini kamaytiradigan qulay usuldir.

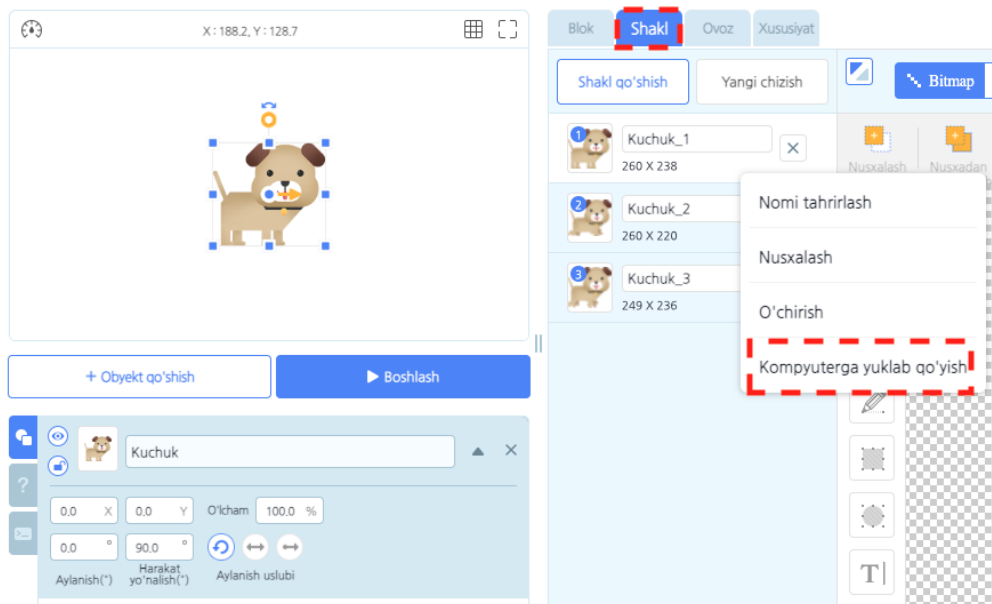
Shunga tegishli Mu editorning qo'llanmasiga qarang:

https://codewith.mu/en/howto/1.2/pgzero_sounds_images {% endhint %}

Keyingi keltiriladigan ma'lumotlar siz bilishingiz kerak bo'lgani uchun berilgan. Biz foydalanayotgan maxsus Mu Editori holatida, kodlashda kerak bo'ladigan barcha resurslar (rasmlar, ovozlar va hokazo) sizning qulayligingiz uchun oldindan kiritib qo'yilgan, shuning uchun bunday qo'shimcha ishlar talab qilinmaydi. Biroq, bu izoh Entry-da har bir obyektning rasmlarini qanday olib kirish mumkinligi bilan qiziqadiganlar uchun qo'shimcha tushuntirishdir.

Entry dan rasm obyektini import qilish

Entry dagi "kuchuk" va "mushuk" obyektlarining rasmlaridan foydalanish uchun Entry-ni oching va kerakli obyektlarning rasm fayllarini import qiling. Import qilish uchun "shakl" yorlig'ini bosing, import qilmoqchi bo'lgan shaklni o'ng sichqon tugmasi bilan bosing va paydo bo'lgan menyudan "kompyuterga yuklab olish" ni tanlang. Uni Pygame Zero uchun yuqorida aytib o'tilgan "images" papkasiga saqlang.



{% hint style="info" %} Ushbu kitobdagi misollar uchun zarur bo'lgan barcha rasmlarni [bu yerdan](#) yuklab olishingiz mumkin. {% endhint %}

Endi har bir obyektini («Kuchukcha» va «Mushukcha») ekranda aks ettirishga o'tamiz va keyingi bo'limda ularning o'zaro salomlashishini kodlashni yakunlaymiz.