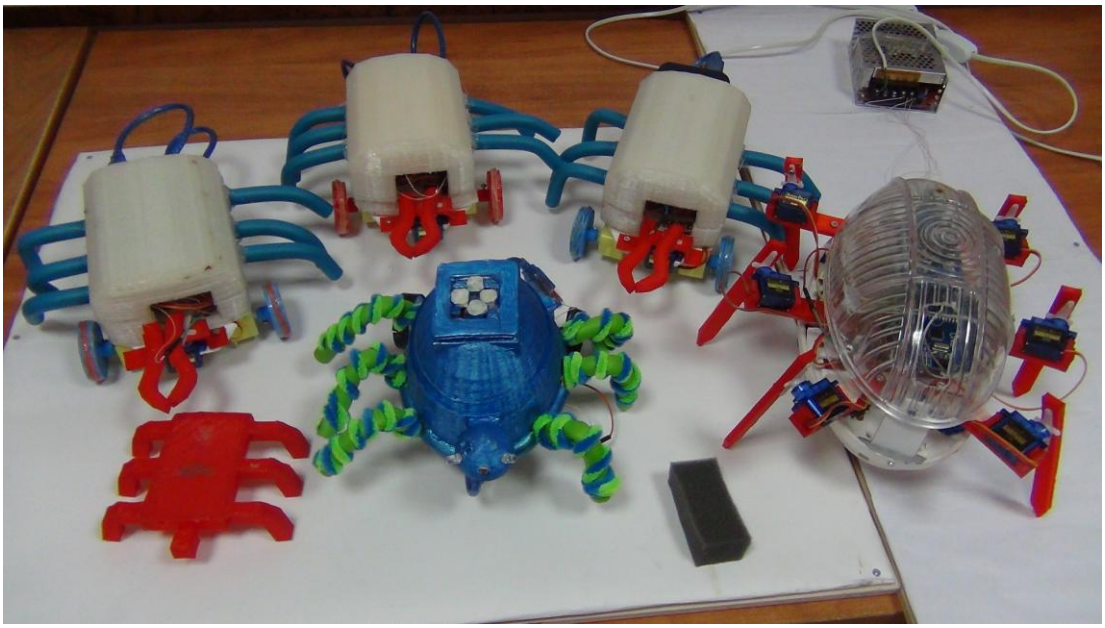
# Проект «Муравейник»

**Авторы проекта:**

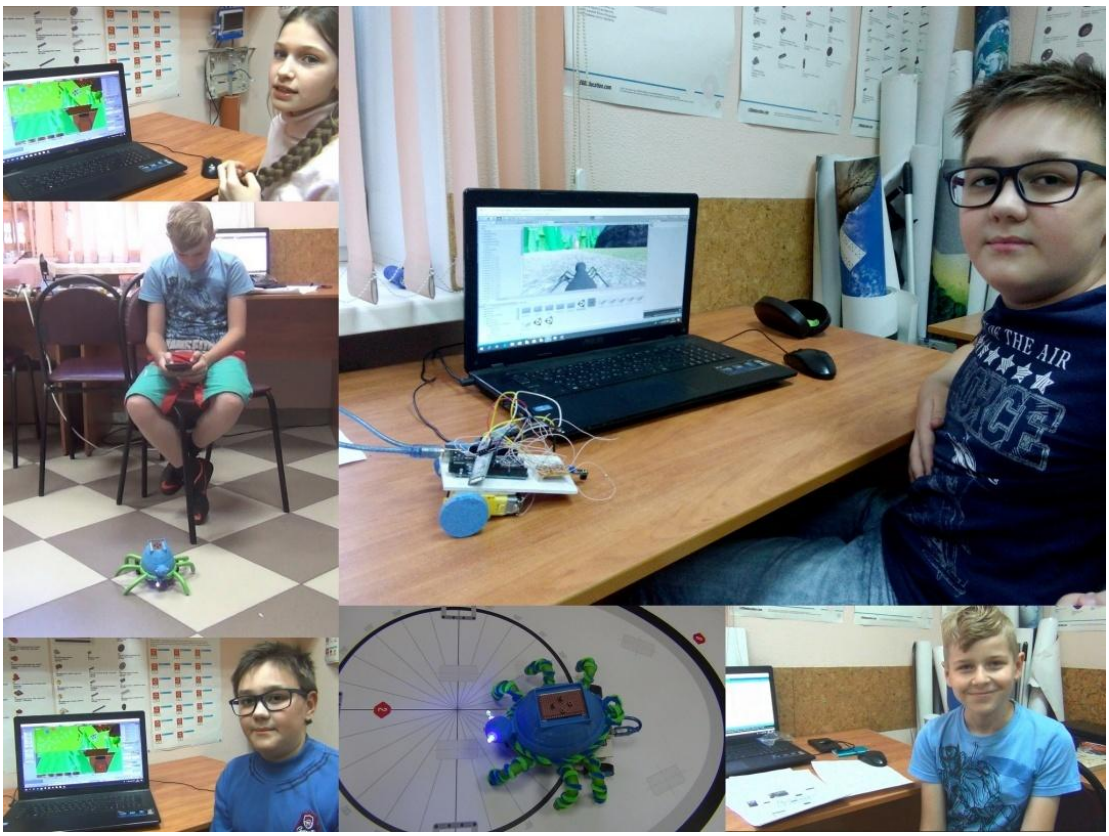
1. Колесников Всеволод
2. Дмитрук Юра
3. Каширина Светлана

**Научный руководитель:**

Корягин Андрей Владимирович



**Цель проекта:** разработка и создание группы мобильных роботизированных устройств и программного обеспечения для отработки коллективного взаимодействия группы роботов между собой и окружающей средой, направленное на решение общей задачи, в целях применения в образовательной, исследовательской и сугубо специализированной деятельности.



**Задачи проекта:**

1. Разработать рой мобильных роботов - муравьев: робот – разведчик, робот – рабочий (3 шт.), робот – охранник и систему технического зрения (СТЗ).

2. Разработать программное обеспечение для мобильных роботов.

3. Разработать программное обеспечение для компьютерного зрения и распознавания объектов.

4. Создать макет области контроля.

5. Разработать программное обеспечение для виртуального отображения окружающей среды и объектов системы.

6. Разработать программу для тех. зрения по отслеживанию мобильных роботов и по расчету их пути.

7. Разработать программу для обратной связи от мобильных роботов к компьютеру и обратно.

пифАград

**Задачи на будущее**:

1. Разработать программу мобильного робота с применением алгоритма муравья.
2. Улучшить проходимость роботов
3. Разработать систему по «общению» между роботами муравьями

**Вводная часть**

Современная робототехника направлена на изучение и управление группой роботов и их взаимодействие между собой и человека. Современные роботы должны обладать сенсорами для ощущения себя в среде и, особенно, должны обладать вариантами технического зрения и беспроводной связи.

Для получения более сложной работы группы роботов используют алгоритмы роя, например, алгоритм муравья или генетический алгоритм.

Наш проект – это модель группового взаимодействия наших роботов.

**Описание проекта**.

Группа роботов состоит из 5 мобильных роботов и системы тех. Зрения. Существует 3 принципа работы роботов:

1. – Ручное управление;

2. – Автономная работа №1

3. – Автономная работа №2.

**Ручное дистанционное управление**: СТЗ фиксирует наличие полезных ресурсов и вредоносных факторов. Процесс происходит автоматически: система видит полезные ресурсы и вредные объекты и отображает их специальными маркерами в компьютерной программе, операторы видят маркеры и отдают специальные команды роботам, а те в свою очередь выполняют поставленные задачи.

**Автономная работа №1**: СТЗ фиксирует наличие полезных ресурсов и вредоносных факторов. Процесс происходит автоматически: система видит полезные ресурсы и вредные объекты и отображает их специальными маркерами в

компьютерной программе – симуляторе. Роботы, согласно заданному алгоритму по разрешению оператором, совершают циклические действия по сбору полезных ресурсов.

**Автономная работа №2**: СТЗ фиксирует наличие полезных ресурсов и вредоносных факторов. Процесс происходит автоматически: система видит полезные ресурсы и вредные объекты и отображает их специальными маркерами в компьютерной программе-симуляторе. После чего система подает команды мобильным роботам и те выполняют поставленную задачу.

**Принципиальная схема мобильного робота**

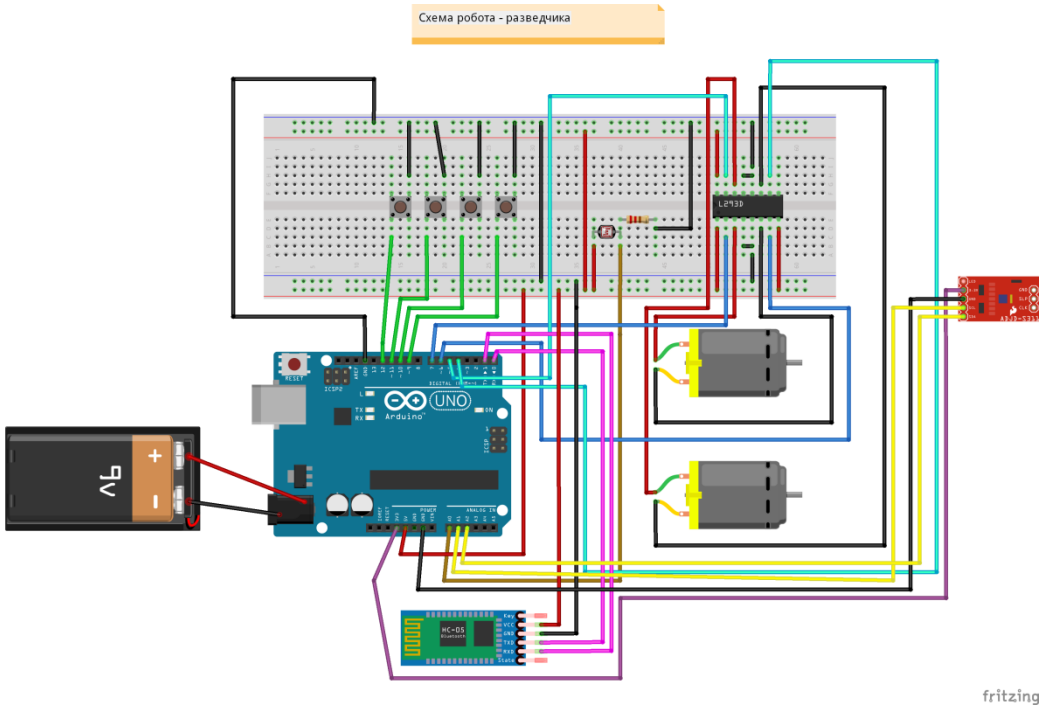
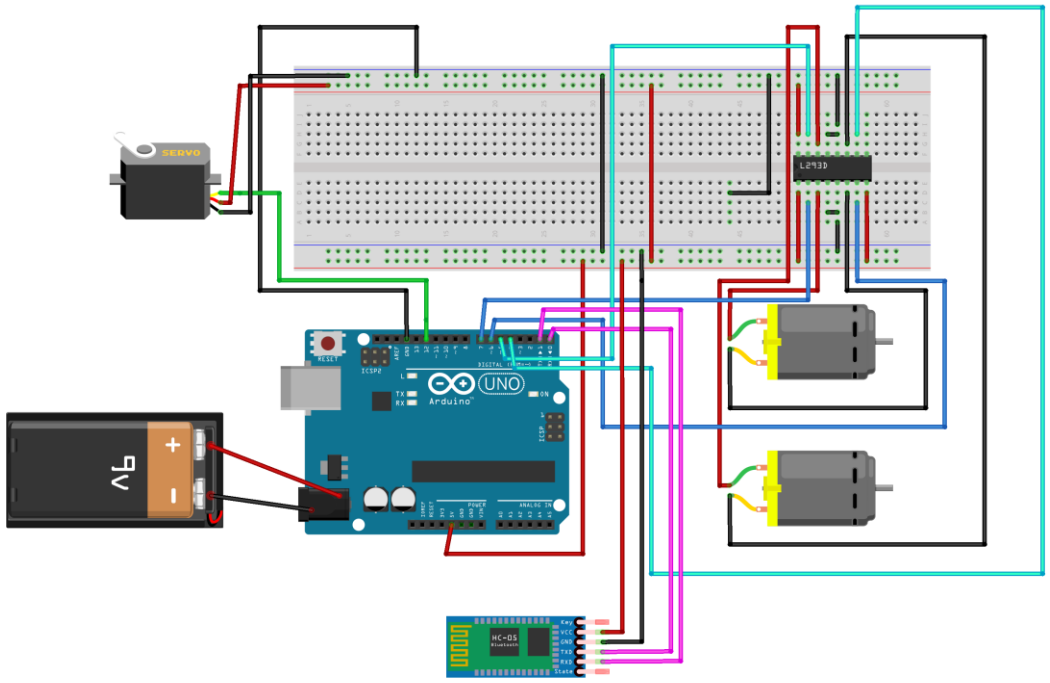


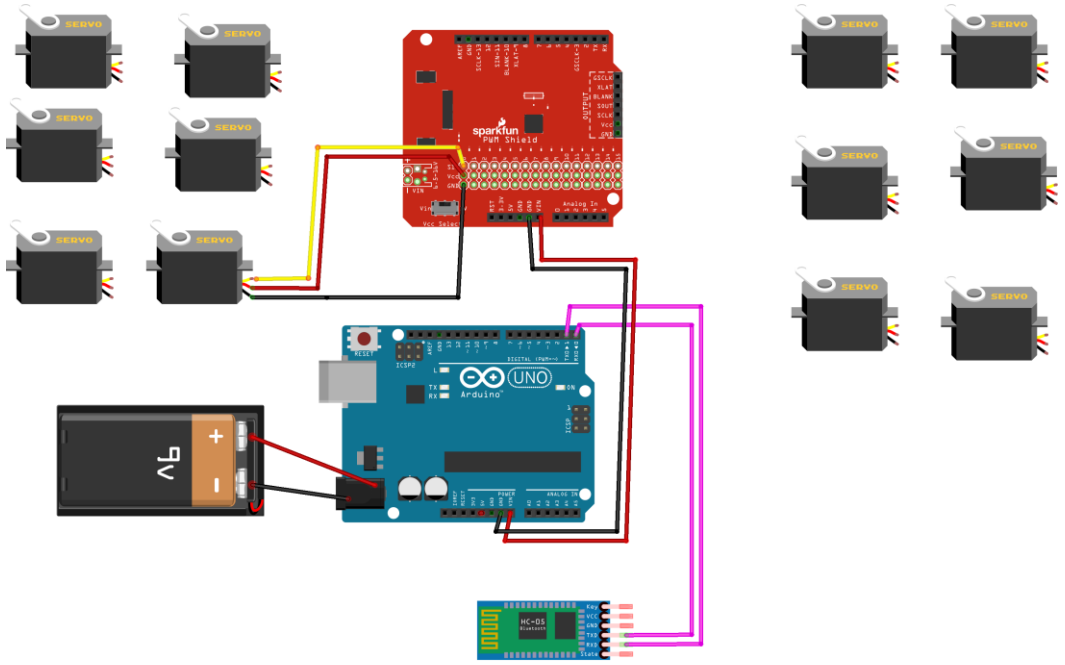
Схема робота - рабочий




fritzing

пифАград

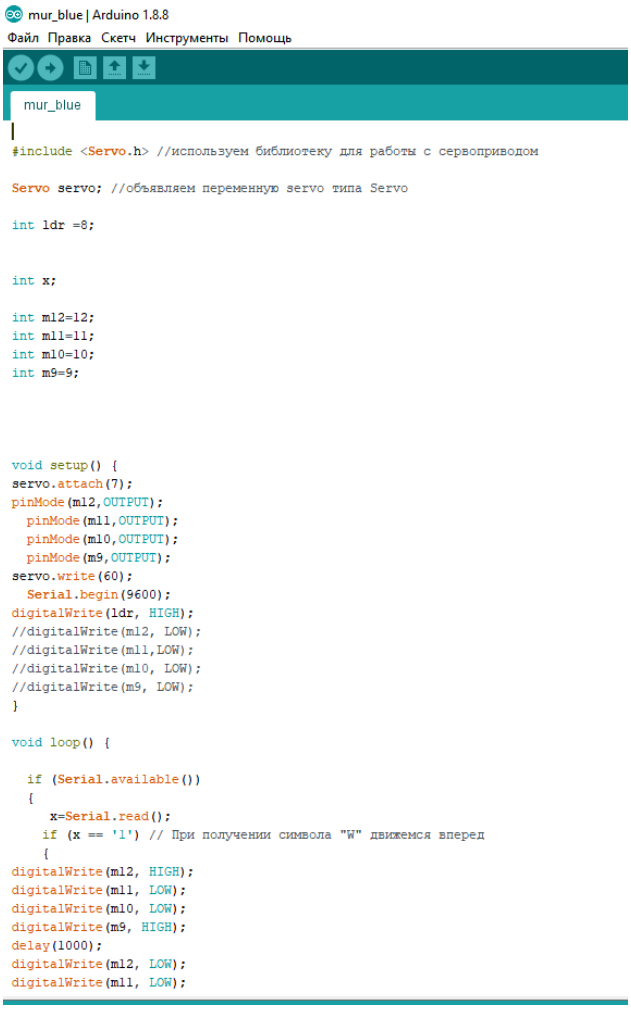
Схема робота - Охранник




fritzing

пифАград

# Программа работы мобильного робота.




```
#include <Servo.h> //используем библиотеку для работы с сервоприводом

Servo servo; //объявляем переменную servo типа Servo

int ldr =8;


int x;

int ml2=12;
int ml1=11;
int ml0=10;
int m9=9;




void setup() {
servo.attach(7);
pinMode(ml2,OUTPUT);
  pinMode(ml1,OUTPUT);
  pinMode(ml0,OUTPUT);
  pinMode(m9,OUTPUT);
servo.write(60);
  Serial.begin(9600);
digitalWrite(ldr, HIGH);
//digitalWrite(ml2, LOW);
//digitalWrite(ml1,LOW);
//digitalWrite(ml0, LOW);
//digitalWrite(m9, LOW);
}

void loop() {

  if (Serial.available())
  {
     x=Serial.read();
    if (x == '1') // При получении символа "W" движемся вперед
    {
digitalWrite(ml2, HIGH);
digitalWrite(ml1, LOW);
digitalWrite(ml0, LOW);
digitalWrite(m9, HIGH);
delay(1000);
digitalWrite(ml2, LOW);
digitalWrite(ml1, LOW);
digitalWrite(ml0, LOW);
digitalWrite(m9, LOW);
    }


else if (x=='2'){
  digitalWrite(ml2, LOW);
digitalWrite(ml1, HIGH);
digitalWrite(ml0, HIGH);
digitalWrite(m9, LOW);
delay(1000);
digitalWrite(ml2, LOW);
digitalWrite(ml1, LOW);
digitalWrite(ml0, LOW);
digitalWrite(m9, LOW);
  }

else if (x=='3'){
 digitalWrite(ml2, LOW);
digitalWrite(ml1, HIGH);
digitalWrite(ml0, LOW);
digitalWrite(m9, HIGH);
delay(1000);
digitalWrite(ml2, LOW);
digitalWrite(ml1, LOW);
digitalWrite(ml0, LOW);
digitalWrite(m9, LOW);
  }

else if(x=='4'){
    digitalWrite(ml2, HIGH);
digitalWrite(ml1, LOW);
digitalWrite(ml0, HIGH);
digitalWrite(m9, LOW);
delay(1000);
digitalWrite(ml2, LOW);
digitalWrite(ml1, LOW);
digitalWrite(ml0, LOW);
digitalWrite(m9, LOW);
    }
else if(x=='5'){
    servo.write(110);
        }
else if(x=='6'){
    servo.write(50);
        }
else  {
```

# Программа работы СТЗ.

*сетка.py - C:\Users\Andry\Documents\wew\сетка.py (3.6.5)*

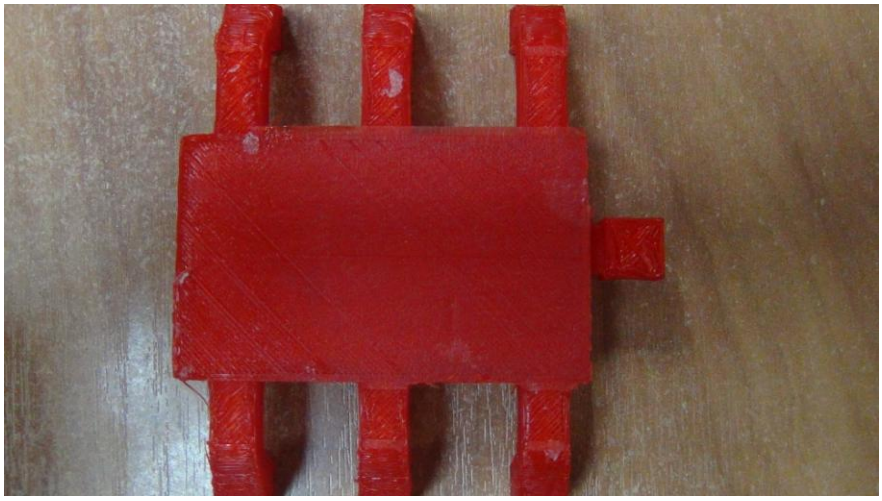
File Edit Format Run Options Window Help

```
from PIL import Image
import cv2
import numpy as np
import math

from pylab import*

cap = cv2.VideoCapture(1)
ret, frame = cap.read()
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter('Woutput181.avi', fourcc, 20.0, (640,480))
template = cv2.imread("WIN1.png",0)
w, h = template.shape[::-1]
x,y = 0,0
while True:
    _, frame = cap.read ()

    i= cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    lower_red = np.array([0,180,200])
    upper_red = np.array([255,225,225])
    #белый - 10,10,0  200,100,255.
    #зелёный - 10 10 0   150 200 150
    #серо-зелёный - 10 10 0  150 200 150

    #mask = cv2.inRange(hsv, lower_red, upper_red)
    #res = cv2.bitwise_and(frame, frame, mask = mask)
    res1 = cv2.matchTemplate(i,template,3)
    min_val, max_val, min_loc, max_loc = cv2.minMaxLoc(res1)
    top_left = max_loc
    bottom_right = (top_left[0] + w, top_left[1] + h)
    x = (top_left[0] + bottom_right[0])/2
    y = (top_left[1] + bottom_right[1])/2
    #for dl in [5,210,410,635]:
        #if x>0+dl and x<200+dl and y>10 and y<500:
            #cv2.rectangle(frame, (dl,0),(210+dl,150) , (255, 0, 0), 3)
        #else:
            #cv2.rectangle(frame, (dl,0),(210+dl,150) , (0, 255, 0), 3)


    cv2.line(frame, (5,0), (5,500), (250,255,255), 2)
    cv2.line(frame, (210,0), (210,500), (250,255,255), 2)
    cv2.line(frame, (410,0), (410,500), (250,255,255), 2)
    cv2.line(frame, (635,0), (635,500), (250,255,255), 2)
    cv2.line(frame, (700,0), (0,0), (250,255,255), 2)
    cv2.line(frame, (700,150), (0,150), (250,255,255), 2)
    cv2.line(frame, (700,320), (0,320), (250,255,255), 2)
    cv2.line(frame, (700,477), (0,477), (250,255,255), 2)
```

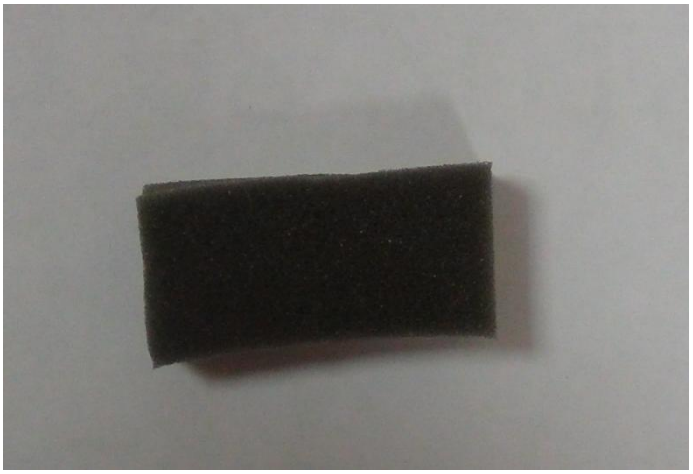
```
  if y<500:
      cv2.rectangle(frame, top_left, bottom_right, (0, 255, 0), 3)
      cv2.putText(frame, "["+ str(x)+","+str(y)+"]", (bottom_right[0]+10,bottom_right[1]+10), cv2.FONT_HERSHEY_SIMPLEX, 1, (255,255,0), 2)
      if x >=30 and x <=150 and y >=80 and y <= 100:
          PressKey(Z)
      if x >=200 and x <=400 and y >=80 and y <= 100:
          PressKey(X)
      if x >=400 and x <=550 and y >=80 and y <= 100:
          PressKey(C)
      if x >=30 and x <=150 and y >=230 and y <= 270:
          PressKey(V)
      if x >=200 and x <=400 and y >=230 and y <= 270:
          PressKey(B)
      if x >=440 and x <=550 and y >=230 and y <= 270:
          PressKey(N)
      if x >=30 and x <=150 and y >=380 and y <= 430:
          PressKey(M)
      if x >=200 and x <=400 and y >=380 and y <= 430:
          PressKey(K)
      if x >=400 and x <=550 and y >=380 and y <= 430:
          PressKey(L)

  cv2.circle(frame,(100,1), 5, (255,255,255), -1)
  cv2.line(frame, (100,101), (int (x),int (y)), (250,255,255), 2)
  d = np.sqrt((int (x) - 100)**2 + (int (y) - 1)**2)
  d1 = d * 0.086
  cv2.putText(frame, str(d1)+ ' '+'cm',  (100, 50), cv2.FONT_HERSHEY_SIMPLEX, 1, (255,0,0), 2)
  if int (x) >=x1[0] and int (x) <= x1[1] and int (y) >= y1[0] and int (y) <= y1[1]:
      print ("T")
      PressKey(T)
      time.sleep(1)
      ReleaseKey(T)

      for i in range (83):
          cv2.circle(frame,(100,1 + i), 5, (255,0,255), 5)
      #cv2.circle(frame,(int (x/2),int (y/2)), 5, (255,0,255), 5)
  if int (x) >=x1[0] and int (x) <= x1[1] and int (y) >= y2[0] and int (y) <= y2[1]:
      print ('Y')
      f1()
```

. . .

**Враг.**



Полезные ресурсы



# Программа связи муравья с компьютером

SEVA.py - I:\33333\Программа_связь_муравей\SEVA.py (3.6.5)

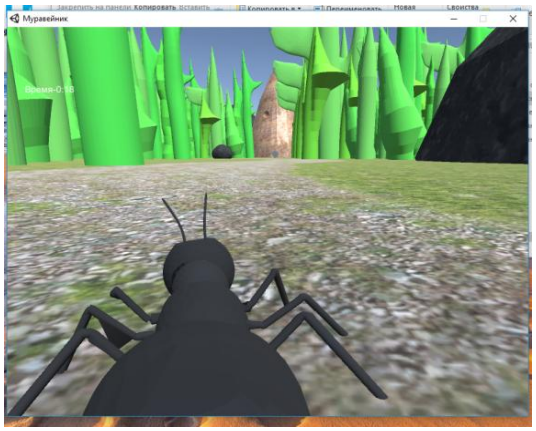
File Edit Format Run Options Window Help

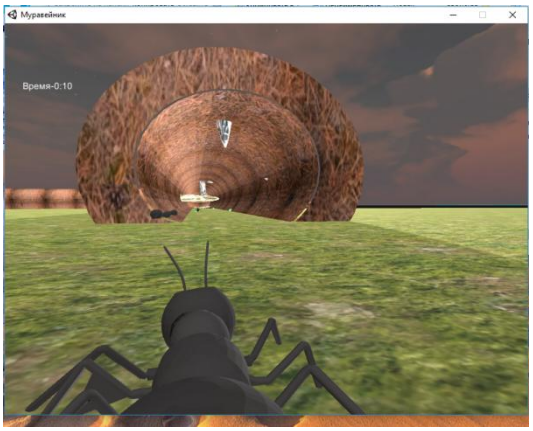
```
import serial
import time
import re
import os
from directkeys import ReleaseKey, PressKey, w, a, s, d, Q

r = serial.Serial('com4', 9600)

while True:
    t=str(r.readline())
    time.sleep(1)

    e=time.clock()
    l=round(e,2)
    #print(t)
    g=re.findall(r'\d', t)
    print(g)
    if g==['1']:
       PressKey(w)
       time.sleep(1)
       ReleaseKey(w)
    elif g==['2']:
        PressKey(s)
        time.sleep(1)
        ReleaseKey(s)
    elif g==['3']:
        PressKey(a)
        time.sleep(1)
        ReleaseKey(a)
    elif g==['4']:
        PressKey(d)
        time.sleep(1)
        ReleaseKey(d)
```

## Симуляторы среды

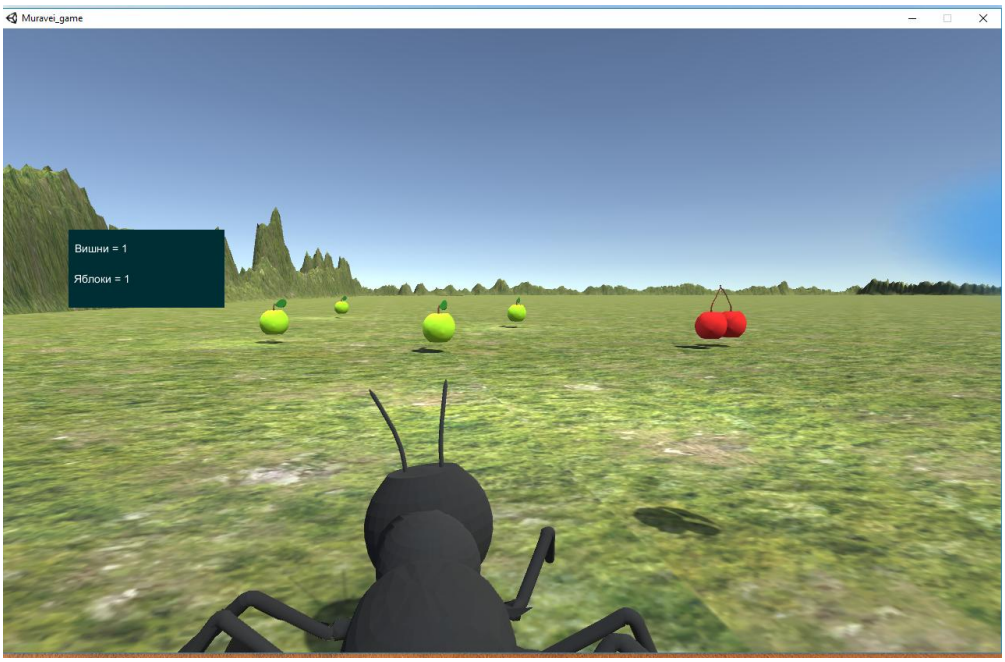
1. Образовательно-исследовательский раздел:

   - Изучение жизни муравьёв



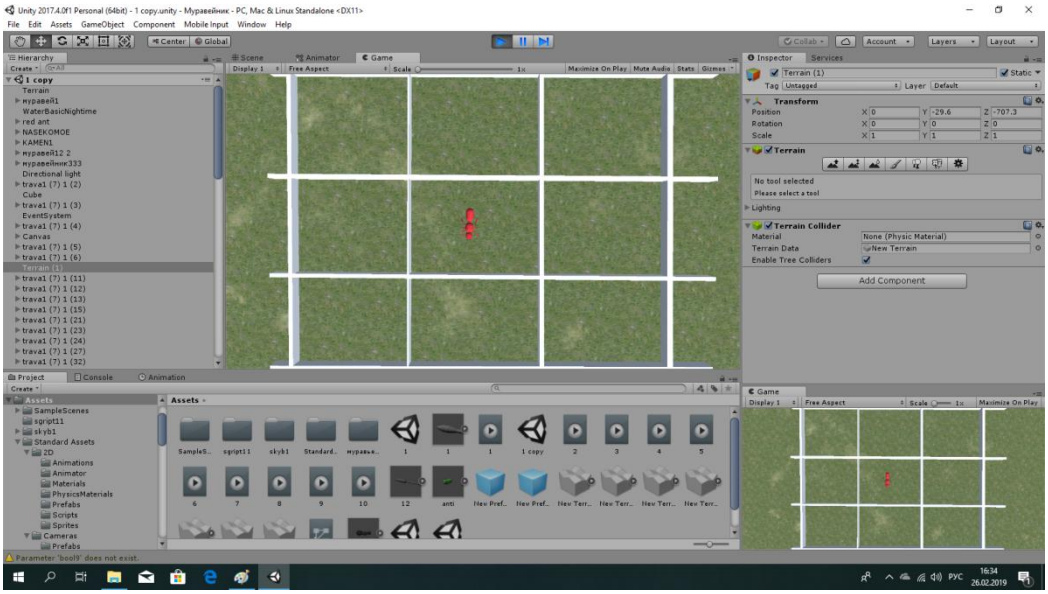




- интерактивные упражнения и задания в области образования для дошкольного и начального школьного обучения.




пифАград

- Специализированная программа под задачи (виртуальный радар поиска и отображения врагов, ресурсов и самих муравьёв)



**Сфера применения** – работов роя

1. Сельское хозяйство и лесничество (борьба с вредителями и сбор урожая, диагностика окружающей среды).
2. Военное применение – разведка, оборона, снабжение
3. Медицина – уменьшенные роботы до десятков микронов можно применить при микрохирургии, умной системы оказания первой помощи
4. Сама система применима и для летательных и подводных роботов с некоторыми модернизациями