



SMART ZOO

Умный Зоопарк для Умного Города

Образование переживает стремительную революцию. Все чаще учебный процесс выходит за пределы учебных заведений. Но в каждом городе существуют традиционные места для внешкольного образования. Это музеи, исторические объекты, ботанические сады, зоопарки. Их ценность с годами не уменьшается. Но чтобы оставаться актуальными и востребованными – они должны соответствовать современным тенденциям.

Мы решили взяться за зоопарк. Посещая зоопарк несколько раз в год мы заметили, что немалое количество вольеров пустуют во время каждого нашего визита: звери могут спать, есть, прятаться, решать свои звериные дела и проблемы.

ЦМИТ
#БиномНьютона

Авторы:
Смирнов Артём
Травин Алексей

Руководители:
Филиппов С.Н.
Любина И.М.

Техническое
зрение

Маршрутизация

МОСКОВСКИЙ **ЗООПАРК:**

> 1,5 млн. посетителей в год

3 часа в среднем на визит

>=25% времени на посещение пустых вольеров

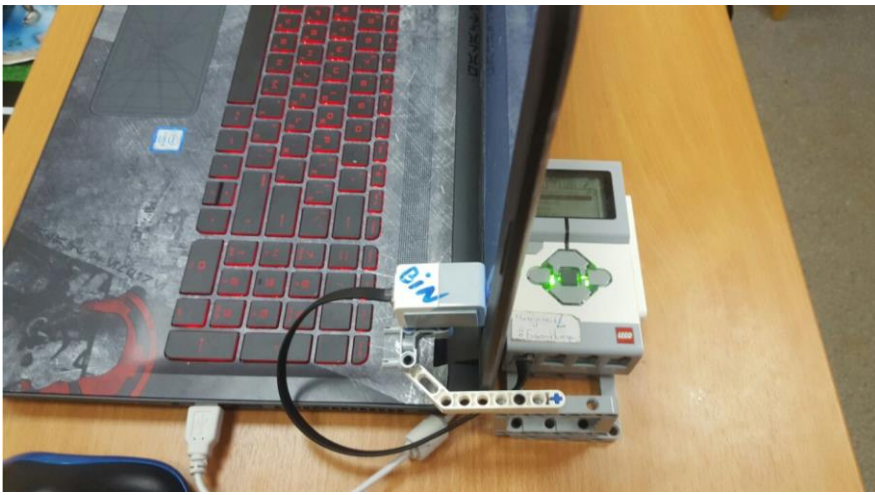
тратится неэффективно
128 лет ежегодно!

Территория зоопарка огромна. Времени на то, чтобы найти вольеры с интересными нам животными, уходит очень много.

Мы предположили, что в более продвинутых зоопарках мира эта проблема уже решена, но изучив этот вопрос на примере таких зоопарков, как Сингапурский и Лондонский, мы были удивлены, что эта проблема оказалась не затронута. После этого мы решили, что этим, возможно, стоит заняться. И обратились с идеей в Московский зоопарк. Где нас горячо поддержали. Мы, воодушевленные, занялись разработкой макета.

Проект состоит из макета зоопарка с подвижными животными, электрокара, камер технического зрения и приложения на компьютере. Животные перемещаются случайным образом, имитируя передвижение реальных животных в зоопарке. Через приложение пользователь выбирает, каких животных он бы хотел увидеть в зоопарке. Камеры отслеживают активность выбранных животных. На основе этой информации система формирует оптимальный маршрут, позволяющий посетителю зоопарка за кратчайшее время осмотреть всех активных животных из списка выбранных. Двигаться по оптимальному маршруту можно не только пешком, но и при помощи электрокара, что особенно актуально для пассажиров с детьми и людей с ограниченными возможностями. Техническое зрение реализовано на языке программирования python3 с помощью библиотеки OpenCV. Также для графической составляющей использовалась библиотека Tkinter. Электрокар работает на базе конструктора Lego Mindstorms EV3 и запрограммирован на программном обеспечении EV3-g.





Примеры программ:


```
gktrsbhjk.py - C:\Users\mrasm\Desktop\programming\python\gktrsbhjk.py (3.7.3)
File Edit Format Run Options Window Help

def detect(self):
    want = [1,1,1,1]
    a11 = open('11.txt', 'r')
    a22 = open('22.txt', 'r')
    a33 = open('33.txt', 'r')
    a44 = open('44.txt', 'r')
    an11 = int(a11.read(1))
    want[0] = an11
    an22 = int(a22.read(1))
    want[1] = an22
    an33 = int(a33.read(1))
    want[2] = an33
    an44 = int(a44.read(1))
    want[3] = an44

    print(want)

    grf = giraffe()
    tgr = tiger()
    ln = lion()
    br = beer()
    #if cv2.waitKey(1) == ord('f'):
    if want[1]==1:
        g = grf
    else:
        g = 0

    if want[2]==1:
        t = tgr
    else:
        t = 0

    if want[3]==1:
        b = br
    else:
        b = 0

    if want[0]==1:
        l = ln
    else:
        l = 0

    return(l, g, b, t)

Ln: 338 Col: 0
```

```
'gktrsbhjk.py - C:\Users\mrasm\Desktop\programming\python\gktrsbhjk.py (3.7.3)'
File Edit Format Run Options Window Help

def show(self, root, field, img1, img2, img3, img4):
    while 1:
        an1, an2, an3, an4 = self.detect()

        print(an4)

        x1, x2, x3, x4 = self.way(an1, an2, an3, an4)

        if an1 == 1:
            print('a')
            field.create_image((x1, 400), image=img1)
        if an2 == 1:
            print('b')
            field.create_image((x2, 400), image=img2)
        if an3 == 1:
            print('v')
            field.create_image((x3, 400), image=img3)
        if an4 == 1:
            print('g')
            field.create_image((x4, 400), image=img4)

        if x1<x2 and x1<x3 and x1<x4 and x1 != 0: #1
            a11 = open('11.txt', 'w')
            a11.write('0')
            a11.close()
            field.create_rectangle(1300, 600, 1920, 1080, fill = 'yellow')
        elif x2<x1 and x2<x3 and x2<x4 and x2 != 0: #2
            a22 = open('22.txt', 'w')
            a22.write('0')
            a22.close()
            field.create_rectangle(1300, 600, 1920, 1080, fill = 'red')
        elif x3<x2 and x3<x1 and x3<x4 and x3 != 0: #3
            a44 = open('44.txt', 'w')
            a44.write('0')
            a44.close()
            field.create_rectangle(1300, 600, 1920, 1080, fill = 'purple')
        elif x4<x2 and x4<x3 and x4<x1 and x4 != 0: #4
            a33 = open('33.txt', 'w')
            a33.write('0')
            a33.close()
            field.create_rectangle(1300, 600, 1920, 1080, fill = 'black')
        else: #0
            field.create_rectangle(1300, 600, 1920, 1080, fill = 'white')

        root.update()
        time.sleep(13)
        field.delete(ALL)

Ln: 283 Col: 56
```

```
gktrsbhjk.py - C:\Users\mrasm\Desktop\programming\python\gktrsbhjk.py (3.7.3)
File Edit Format Run Options Window Help

from tkinter import *
import time
import random
import cv2
import numpy as np

def giraffe():
    cap1 = cv2.VideoCapture(4)
    for i in range(5):
        _, image = cap1.read()

    #image = cv2.imread('Photo39.jpg')

    hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV) #Преобразуем в HSV
    color_low = (0,0,100)
    color_high = (25,255,255)
    only_hsv = cv2.inRange(hsv, color_low, color_high)

    moments = cv2.moments(only_hsv, 1) # получим моменты
    x_moment = moments['m01']
    y_moment = moments['m10']
    area = moments['m00']
    x = int(x_moment / area)
    y = int(y_moment / area)
    print("Жираф")
    print(area)
    cv2.putText(image, "Giraffe!", (x,y), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,255), 2)
    if(area < 21000):
        grf = 0
        print('NO')
    else:
        grf = 1

    return(grf)

Ln: 22 Col: 0
```

