

АНАЛИЗ ДАННЫХ И ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ [in GameDev]

Отчет по лабораторной работе #2 выполнил:

- Исмагилов Денис Рустамович
- РИ210945

Отметка о выполнении заданий (заполняется студентом):

Задание	Выполнение	Баллы
-----	-----	-----
Задание 1	*	60
Задание 2	#	20
Задание 3	#	20

знак "*" - задание выполнено; знак "#" - задание не выполнено;

Работу проверили:

- к.т.н., доцент Денисов Д.В.
- к.э.н., доцент Панов М.А.
- ст. преп., Фадеев В.О.

Цель работы

Ознакомиться с технологией работы связки Python – Google Sheets - Unity

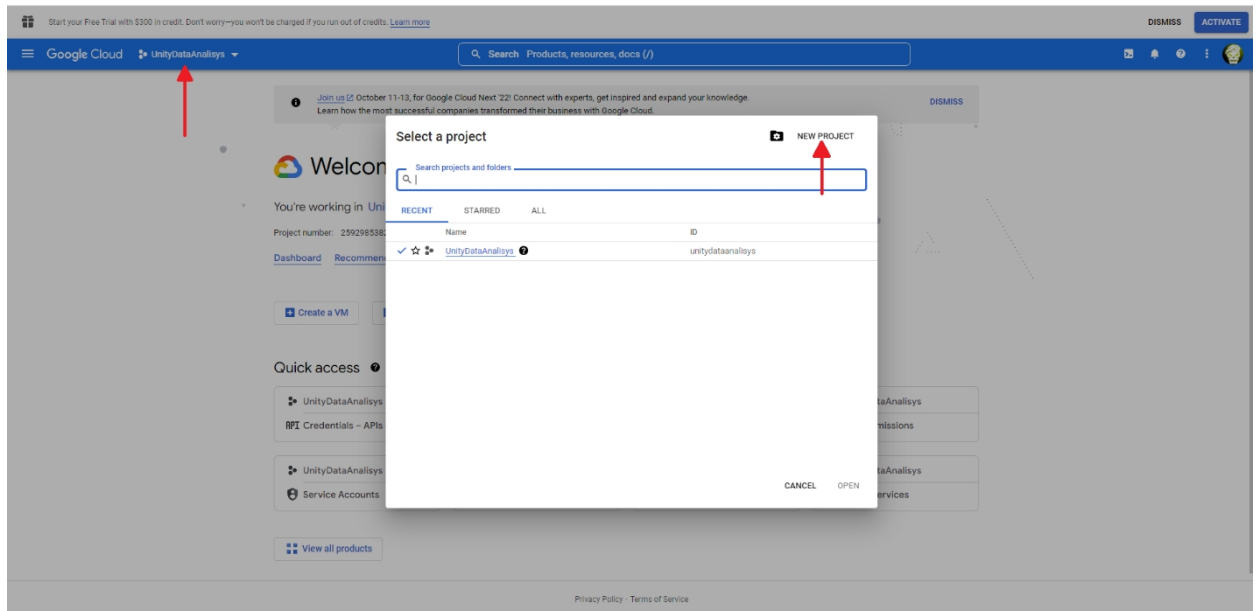
Задание 1

Реализовать совместную работу и передачу данных в связке Python - Google-Sheets – Unity.

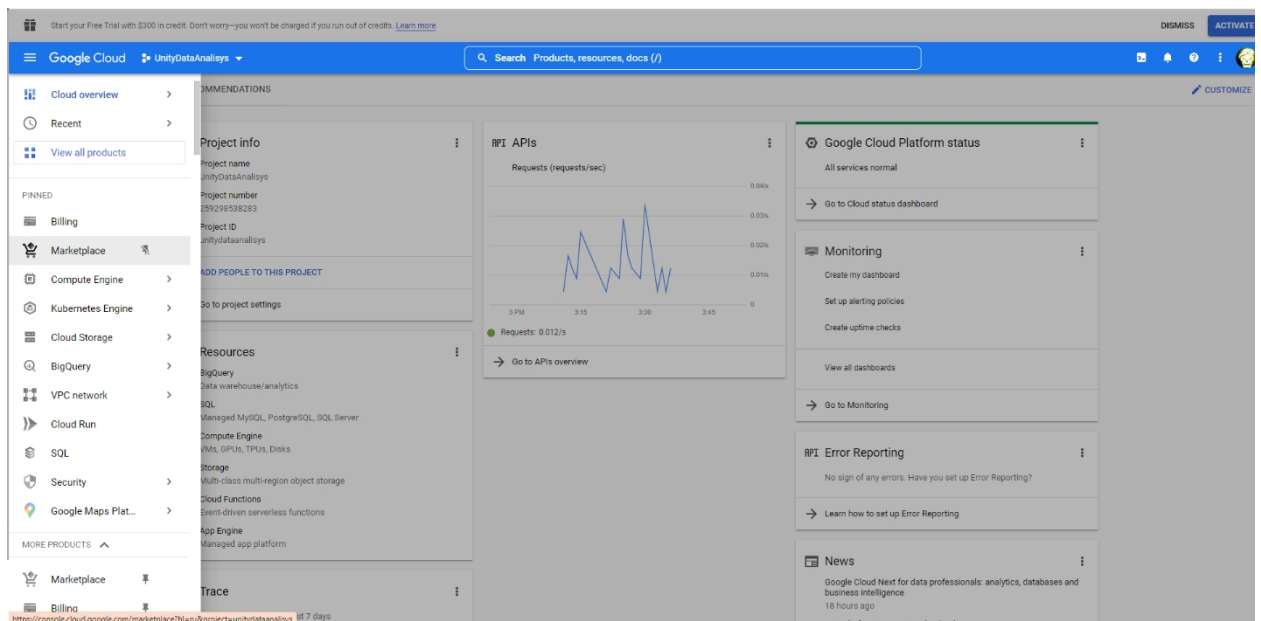
- В облачном сервисе google console подключить API для работы с google sheets и google drive.
- Реализовать запись данных из скрипта на python в google-таблицу. Данные описывают изменение темпа инфляции на протяжении 11 отсчётных периодов, с учётом стоимости игрового объекта в каждый период.
- Создать новый проект на Unity, который будет получать данные из google-таблицы, в которую были записаны данные в предыдущем пункте.
- Написать функционал на Unity, в котором будет воспроизводиться аудио-файл в зависимости от значения данных из таблицы.

Ход работы:

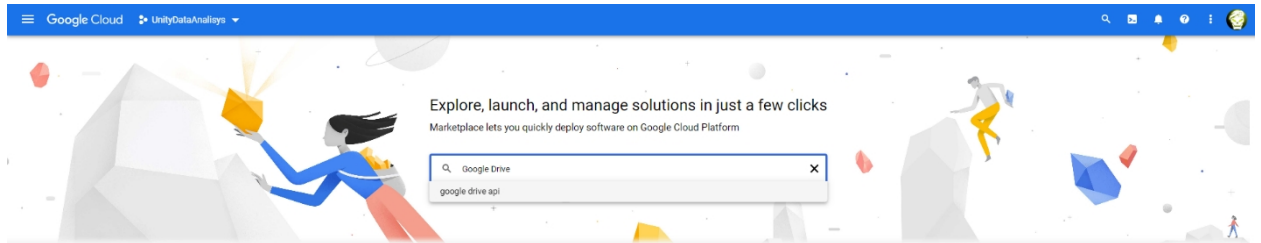
- Зайдём в Google console cloud и создадим новый проект. Для этого нажмём на кнопку справа от надписи Google Cloud – New Project



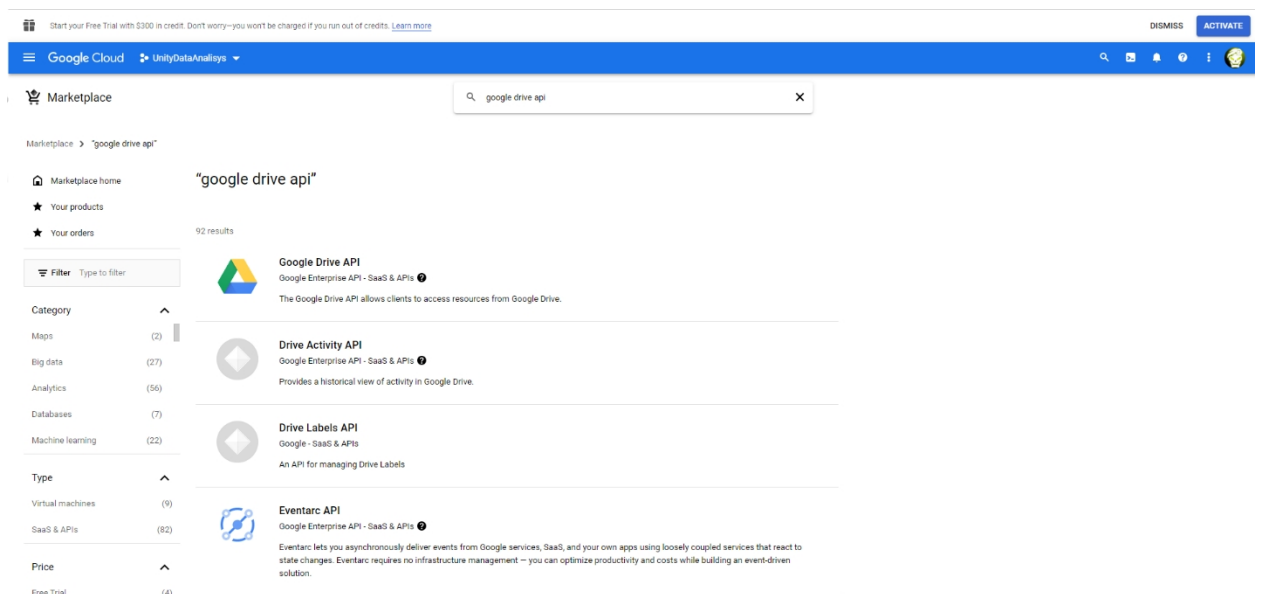
- Зайдём в созданный проект и перейдём в Marketplace. Для этого нажмём на меню, слева от названия Google Cloud.



- В появившейся поисковой строке введём “Google Drive Api” и нажмём на предложенный вариант.



- Зайдём в Google Drive API и нажмём Enable.



После этого также найдём “Google Sheets Api” и установим его. Далее зайдём в “APIs & Services” , перейдём во вкладку Credentials и создадим Service account(+ Create Credentials). Зададим имя новому аккаунту. После этого он появится ниже, в категории Service Account. Нажмём на него. Перейдём в категорию Keys и создадим новый JSON ключ(ADD KEY – Create Key – JSON). Будет предложено сохранить файл-ключ. Сохраните его в папке с новым файлом Python, с которым мы скоро начнём работу.

Google Cloud UnityDataAnalysis

APIs & Services

Library

Credentials

OAuth consent screen

Domain verification

Page usage agreements

CREATE CREDENTIALS

DELETE

Create credentials to access Google Cloud APIs

Remember this

API key

Identifies your project using a simple API key to check quota and access

OAuth client ID

Requests user consent so your app can access the user's data

Service account

Enables server-to-server, app-level authentication using robot accounts

Help me choose

Asks a few questions to help you decide which type of credential to use

API Keys

Name

API key

Restrictions

Google Sheets API

SHOW KEY

OAuth 2.0 Client IDs

Name

Creation date

Type

Client ID

Actions

No OAuth clients to display

Service Accounts

Email

Name

UnityServiceDataAnalysis

Manage service accounts

Actions

DETAILS

PERMISSIONS

KEYS

METRICS

LOGS

Keys

Service account keys could pose a security risk if compromised. We recommend you avoid downloading service account keys and instead use the [Workload Identity Federation](#). You can learn more about the best way to authenticate service accounts on Google Cloud [here](#).

Add a new key pair or upload a public key certificate from an existing

Block service account key creation using [organization policies](#). [Learn more about setting organization policies for service accounts](#)

ADD KEY

Type

Status

Key

Active

05a7991b08c64b97c05b562c2f2e8d62

Create private key for "UnityServiceDataAnalysis"

Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.

Key type

JSON

Recommended

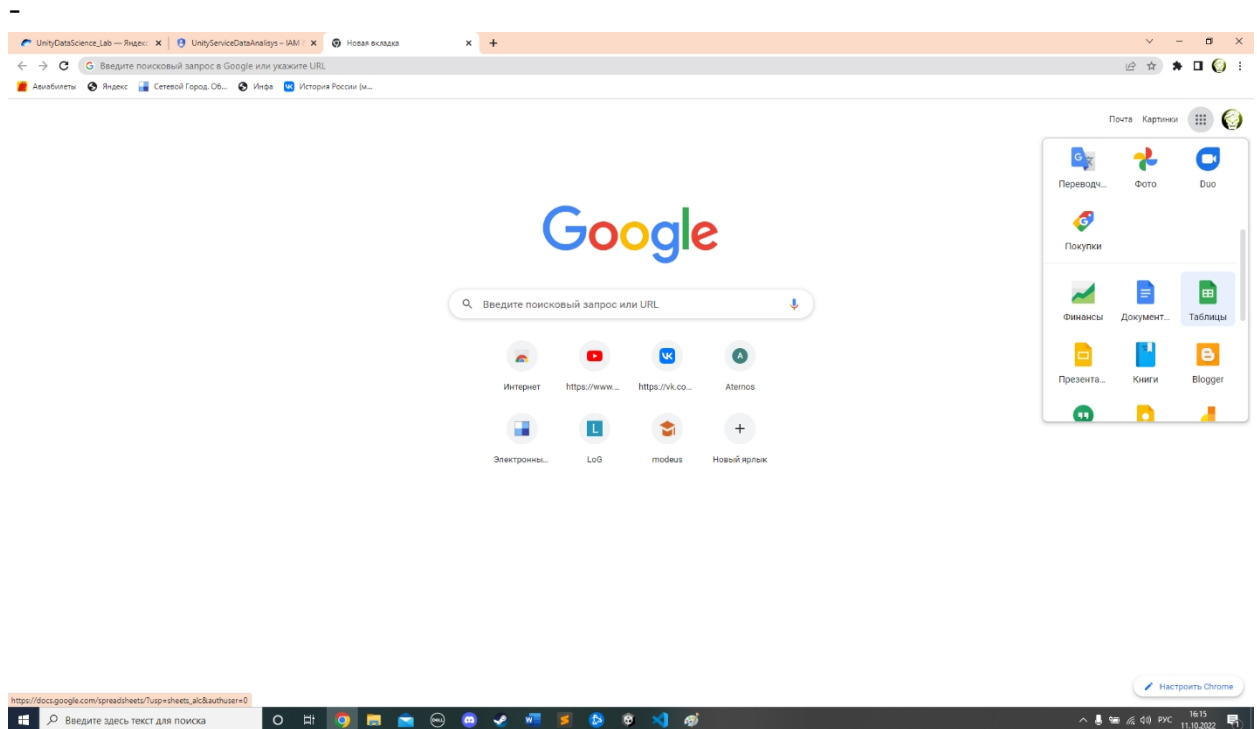
P12

For backward compatibility with code using the P12 format

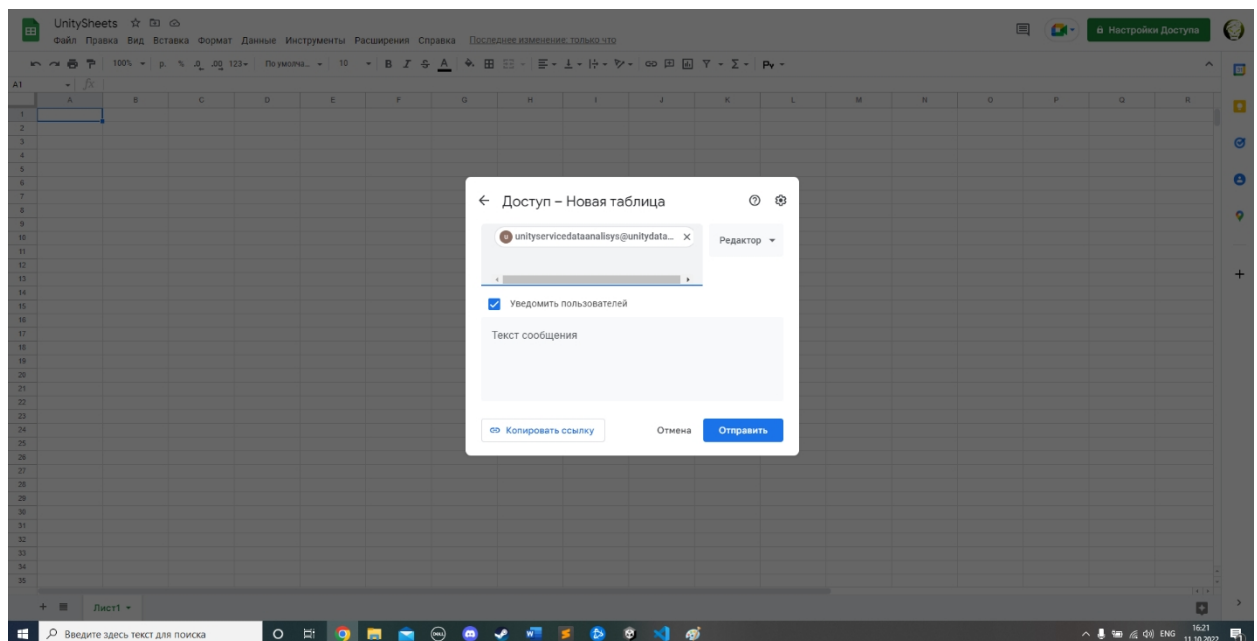
CANCEL

CREATE

Перейдём в категорию Details и скопируем Email. Далее зайдём на стартовую страницу и откроем Таблицы. Создадим новый файл.

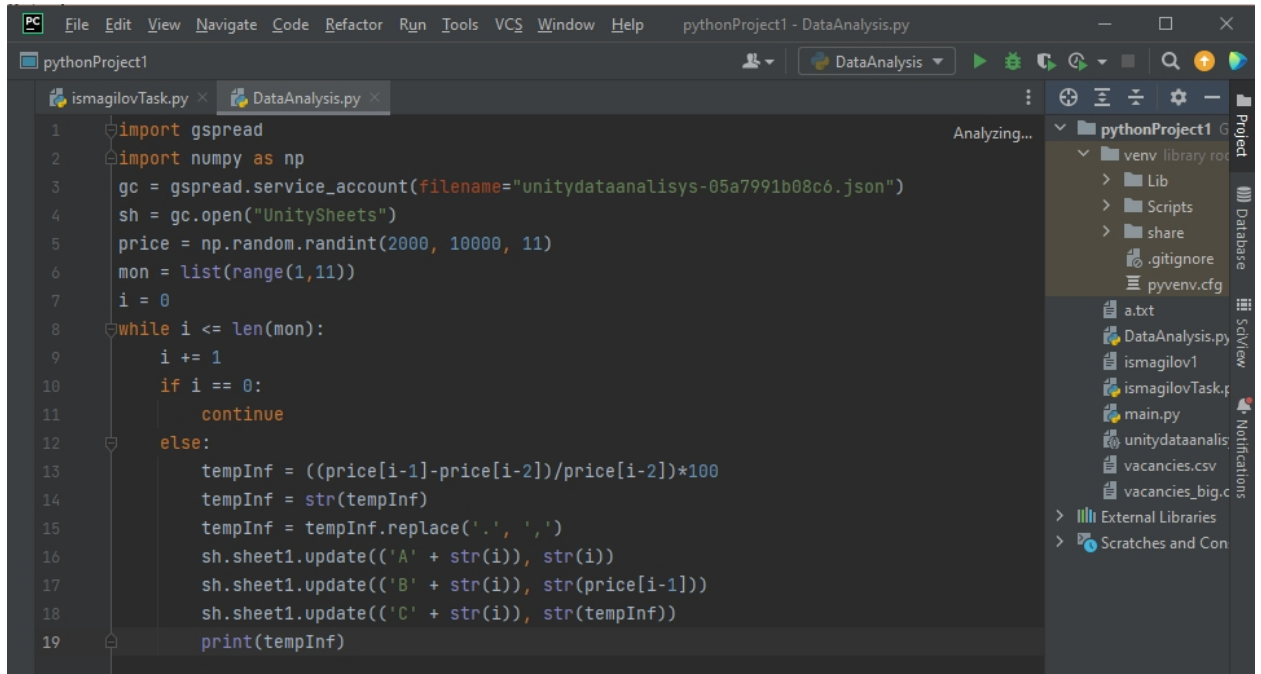


- Нажмём кнопку Настройки доступа справа сверху и введём скопированный адрес Email(Для этого также придётся назвать свою таблицу, например UnitySheets). После этого выберем для него роль “Редактор”



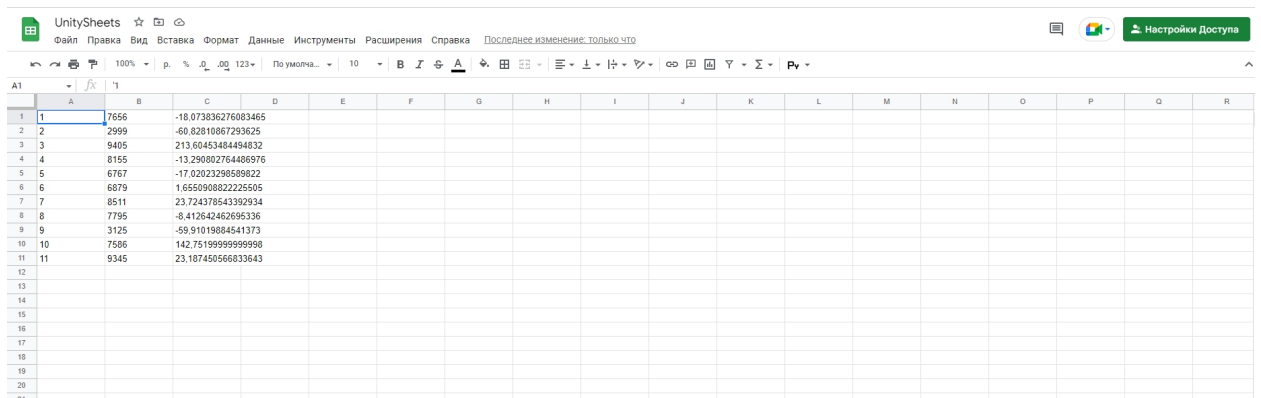
- Теперь перейдём в файл Python. Установим для него библиотеки Numpy и gspread(File - Settings – Python Interpreter). Далее напишем код для случайной

генерации чисел и сохранении этих чисел в файле таблицы, которую мы создали на предыдущих шагах.



```
1 import gspread
2 import numpy as np
3 gc = gspread.service_account(filename="unitydataanalysis-05a7991b08c6.json")
4 sh = gc.open("UnitySheets")
5 price = np.random.randint(2000, 10000, 11)
6 mon = list(range(1,11))
7 i = 0
8 while i <= len(mon):
9     i += 1
10    if i == 0:
11        continue
12    else:
13        tempInf = ((price[i-1]-price[i-2])/price[i-2])*100
14        tempInf = str(tempInf)
15        tempInf = tempInf.replace('.', ',')
16        sh.sheet1.update(('A' + str(i)), str(i))
17        sh.sheet1.update(('B' + str(i)), str(price[i-1]))
18        sh.sheet1.update(('C' + str(i)), str(tempInf))
19        print(tempInf)
```

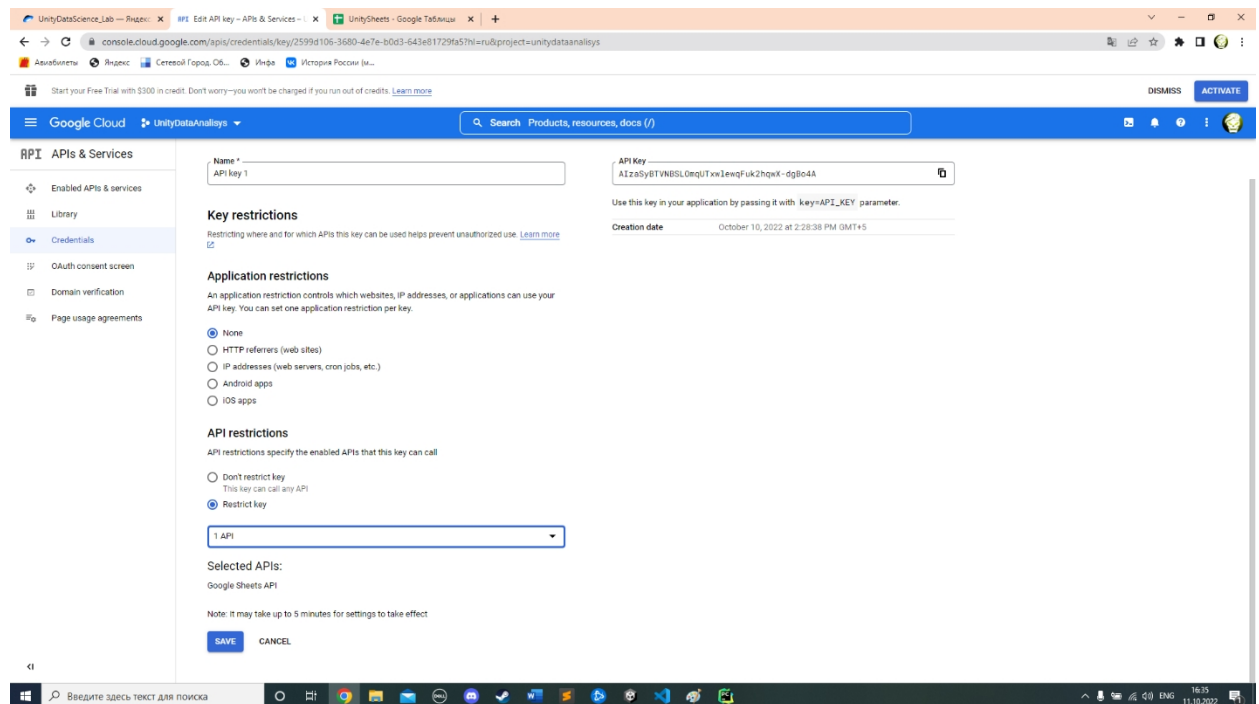
- Имя файла service_ассоунт равно названию файла-ключа, который мы предварительно скачали в ту же папку, что и файл Python.
- Название gc.open(...) равно вашему названию таблицы.
- После этого в файле таблицы появятся новые данные.



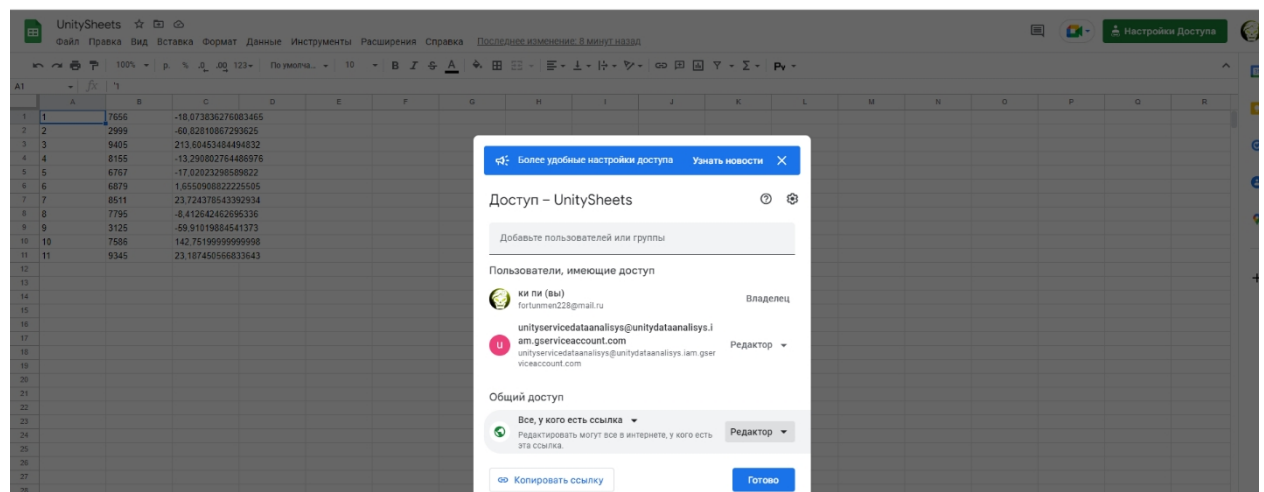
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	1	7656	-18.073836276083465															
2	2	2999	-60.82810867293625															
3	3	9405	213.60453484494832															
4	4	8155	-13.290802764406976															
5	5	6767	-17.0202329080822															
6	6	6879	1.6550908822225505															
7	7	8511	23.724378543392934															
8	8	7795	-8.470642402695336															
9	9	3125	-59.51019804541373															
10	10	7506	142.75199999999998															
11	11	9345	23.187450566833643															
12																		
13																		
14																		
15																		
16																		
17																		
18																		
19																		
20																		
21																		

- После этого вернёмся на страницу Console Google cloud, вкладку APIs & Services, Credentials и создадим API key. После этого напротив ключа, справа,

нажмём на три точки – Edit Api Key. На настройке API restrictions выберем Restrict key и в выпадающем меню выберем “Google Sheets API”

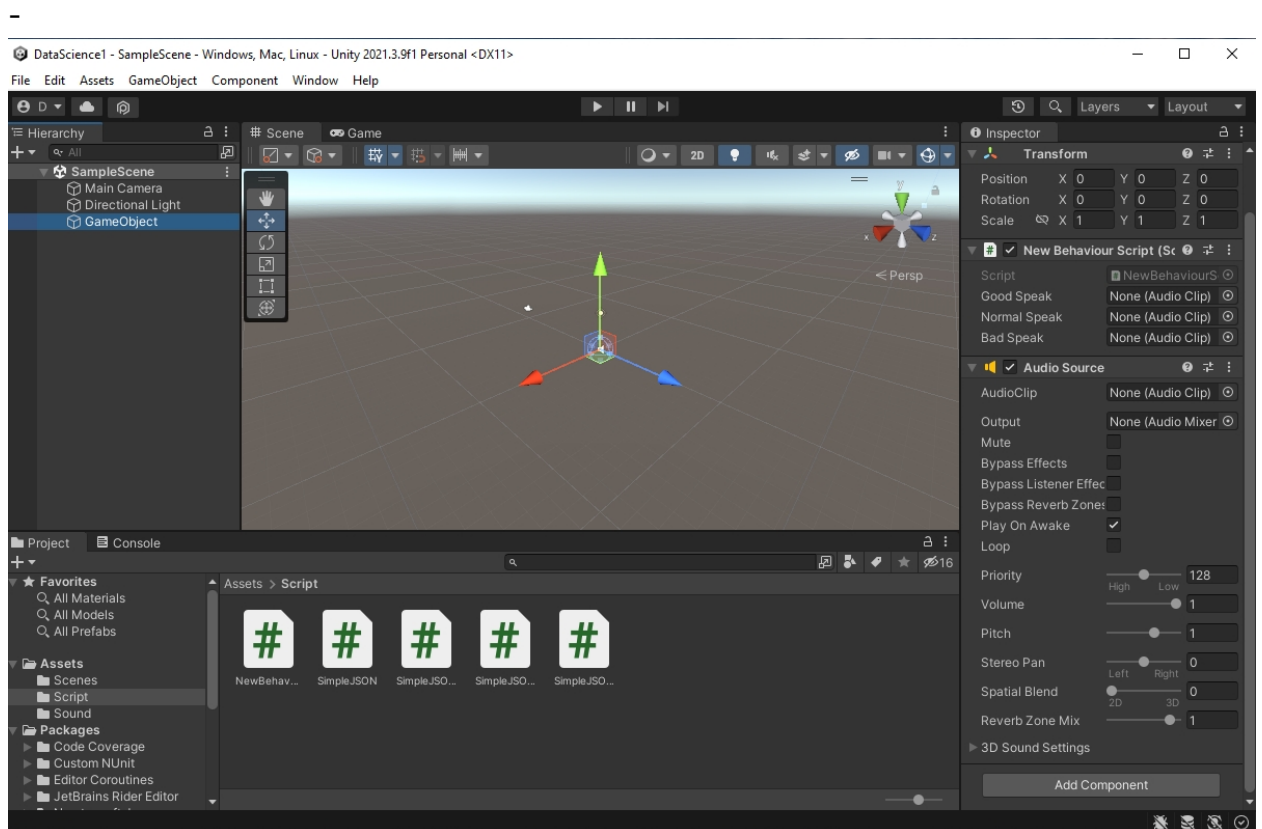


- Сохраним изменения. Далее перейдём в таблицу и настроим доступ. Зайдем в настройки доступа и позволим редактировать таблицу всем, у кого есть ссылка. Для этого во вкладке общий доступ выберем роль редактор.



- Далее перейдём в Unity и создадим новый проект с 3D сценой. После этого скачаем в папку Assets файлы с скриптами и звуковыми файлами, которые прилагаются вместе с отчётом.

- Создадим новый пустой объект и добавим в него компонент звуковой компонент(Add component – Audio Source). После этого создадим новый скрипт в импортированной папке Scripts и назовём его New Behaviour. Присоединим его к новому объекту перетягиванием скрипта на объект в меню, слева от сцены. Откроем скрипт и редактируем его




```
NewBehaviourScript.cs | Параметры
Assets > Script > NewBehaviourScript.cs > NewBehaviourScript > GoogleSheets0
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.Networking;
5 using SimpleJSON;
6
7 0 references
8 public class NewBehaviourScript : MonoBehaviour
9 {
10     0 references
11     public AudioClip goodSpeak;
12     0 references
13     public AudioClip NormalSpeak;
14     0 references
15     public AudioClip BadSpeak;
16     0 references
17     private AudioSource selectAudio;
18     2 references
19     private Dictionary<string,float> dataSet = new Dictionary<string, float>();
20     0 references
21     private bool statusStart = false;
22     0 references
23     private int i = 1;
24
25     0 references
26     void Start()
27     {
28         StartCoroutine(GoogleSheets());
29     }
30
31     0 references
32     void Update()
33     {
34     }
35
36     1 reference
37     IEnumerator GoogleSheets()
38     {
39         UnityWebRequest currentResp = UnityWebRequest.Get("https://sheets.googleapis.com/v4/spreadsheets/1fFvxyTIB-vN5k29u4QPb3kRnW8AptT392QfKIXw5Wlo/values/1mcr1?key=AIzaSy8TVNB5LOmql
40         yield return currentResp.SendWebRequest();
41         string rawResp = currentResp.downloadHandler.text;
42         var rawJson = JSON.Parse(rawResp);
43         foreach (var itemRawJson in rawJson["values"])
44         {
45             var parseJson = JSON.Parse(itemRawJson.ToString());
46             var selectRow = parseJson[0].AsStringList;
47             dataSet.Add(("Mon_" + selectRow[0]), float.Parse(selectRow[2]));
48         }
49         Debug.Log(dataSet["Mon_1"]);
50     }
51 }
```

- Подключим необходимые библиотеки , такие как:

- UnityEngine
- UnityEngine.Networking
- SimpleJSON

(ВНИМАНИЕ!

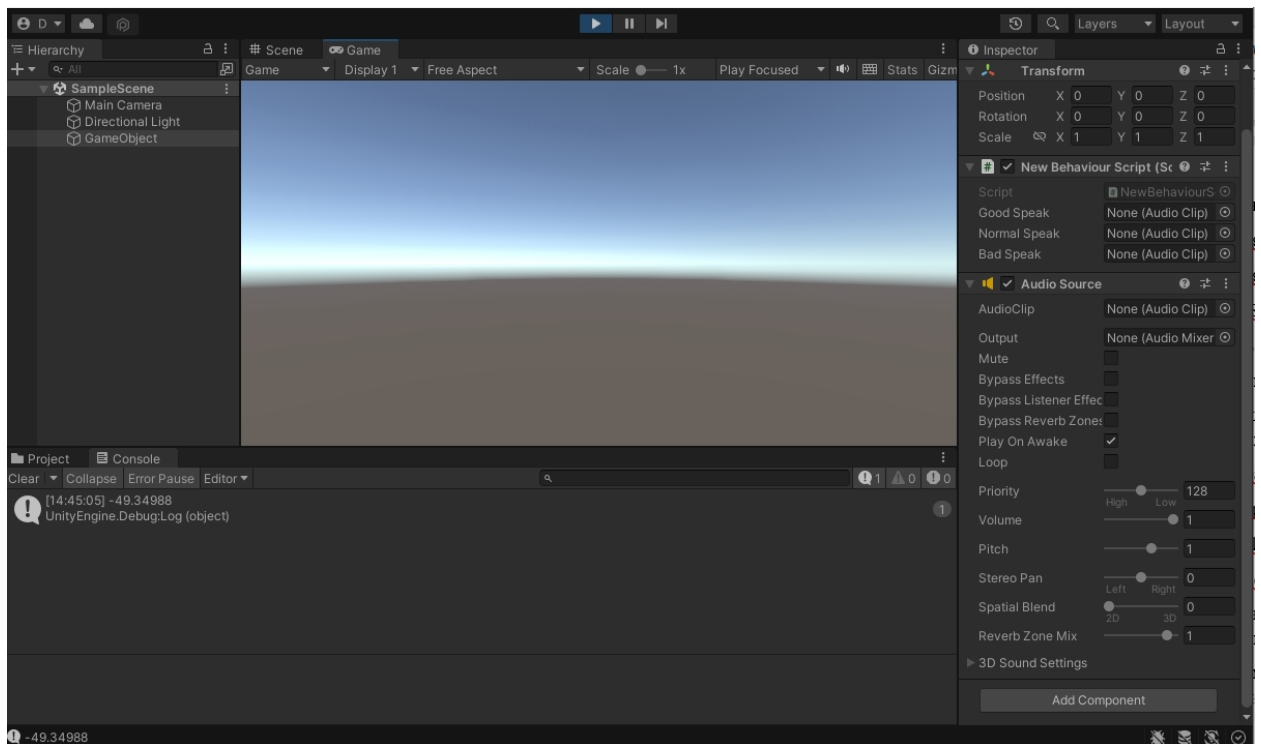
Для работы этих библиотек необходимо установить .NET Framework 4.7.1)

Наследуем наш скрипт от MonoBehaviour. Создадим заготовки для звуковых файлов с названиями:

- goodSpeak
- normalSpeak
- badSpeak
- selectAudio

Реализуем скачивание данных с GoogleSheets и запись их в словарь. Сделаем вывод данных в Debug.log

После перейдём в Unity и запустим нашу сцену. В консоли появится надпись. Это будет означать, что наш скрипт успешно выполнен.



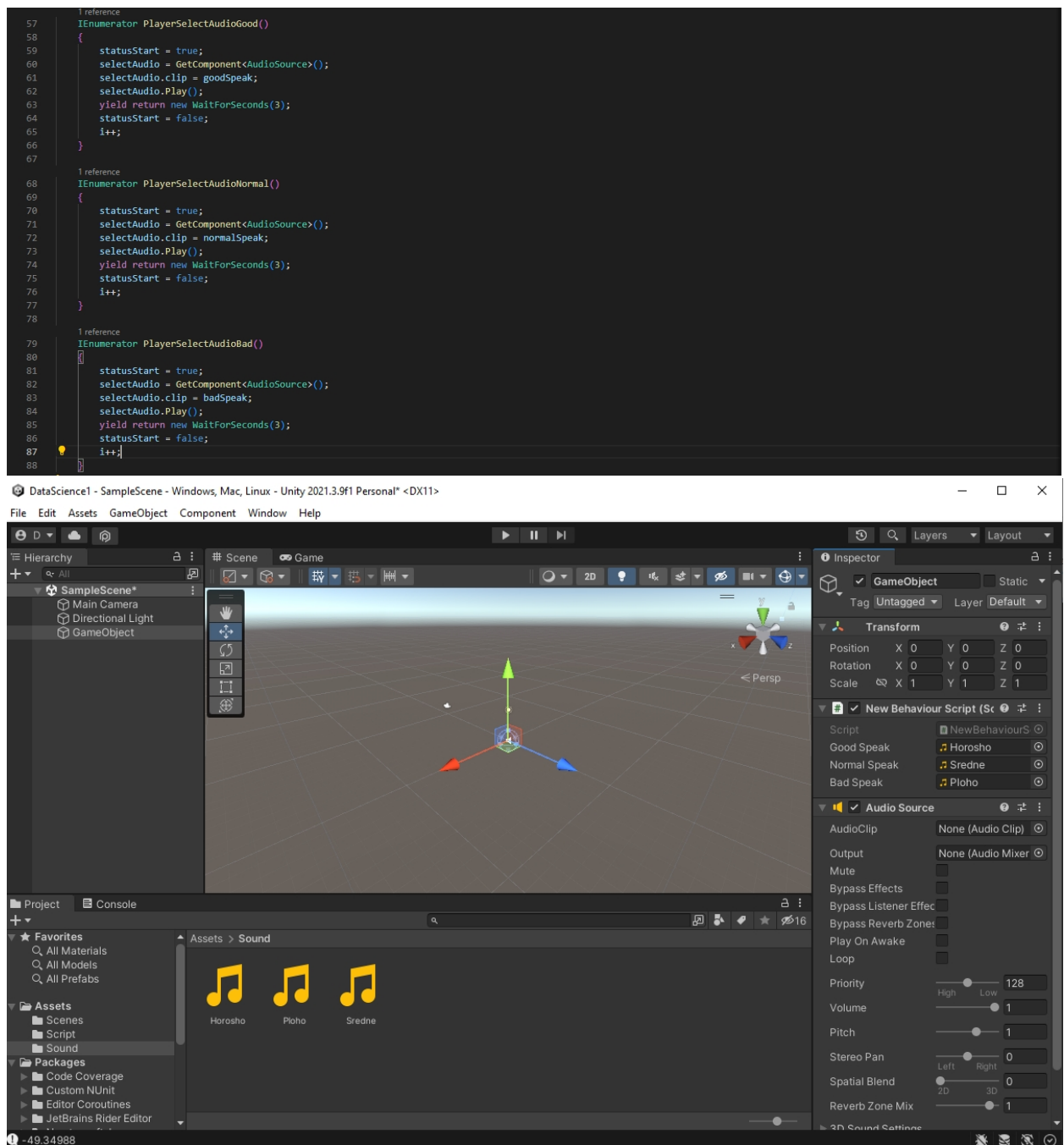
-Уберём вывод первого элемента в консоль. Реализуем вывод элементов GoogleSheets в функции Update.

```

22 void Update()
23 {
24     if(dataSet["Mon_" + i.ToString()] <= 10 & statusStart == false & i != dataSet.Count)
25     {
26         StartCoroutine(PlayerSelectAudioGood());
27         Debug.Log(dataSet["Mon_" + i.ToString()]);
28     }
29
30     if(dataSet["Mon_" + i.ToString()] > 10 & dataSet["Mon_" + i.ToString()] < 100 & statusStart == false & i != dataSet.Count)
31     {
32         StartCoroutine(PlayerSelectAudioNormal());
33         Debug.Log(dataSet["Mon_" + i.ToString()]);
34     }
35
36     if(dataSet["Mon_" + i.ToString()] >= 100 & statusStart == false & i != dataSet.Count)
37     {
38         StartCoroutine(PlayerSelectAudioBad());
39         Debug.Log(dataSet["Mon_" + i.ToString()]);
40     }
41 }
42

```

-Создадим воспроизведение звуковых эффектов. Для этого создадим методы, которые будут вызываться во время вывода элементов в консоль. В Unity уберём авто воспроизведение(Play on awake) и подключим звуковые файлы, которые мы установили ранее.



-Запустим программу в Unity. Теперь будет выводиться звук в зависимости от того, какое число мы передали в словарь из GoogleSheets

Вывод: Я научился использовать API из google console для работы с GoogleSheets и работать в связке Python - Google Sheets - Unity.

Весь код, использованный в отчёте можно найти в репозитории, вместе с отчётом.

