

18

Время приключе

«РЕШУ ЕГЭ»: русс

Входящие - Почта

Во все тяжкие см

Организационны

Формирование ф

Ведомость ОП2

Мессенджер

pixel monster 2 - A

Drop icons - Iconfi

Искать

Мессенджер

DA-in-GameDe

гугл коллаб

Добро пожало

Копия блокн

colab.research.google.com

Копия блокнота "Добро пожаловать в Colaboratory!" - Colaboratory

★ 28 отзывов

ABP

Копия блокнота "Добро пожаловать в Colaboratory!"

Файл

Изменить

Вид

Вставка

Среда выполнения

Инструменты

Справка

Последнее сохранение: 11:59

Комментировать

Поделиться

Подключиться

Редактирование

Содержание

Начало работы

Анализ и обработка данных

Машинное обучение

Ресурсы по теме

Примеры

Раздел

Добро пожаловать в Colab!

Уже знакомы с Colab? В этом видео рассказывается о функциях, которые вы могли пропустить: интерактивных таблицах, истории выполненного кода и палитре команд.

3 Cool Google Colab Features

+ Код

+ Текст

Что такое Colab?

Colaboratory, или просто Colab, позволяет писать и выполнять код Python в браузере. При этом:

- не требуется никакой настройки;
- бесплатный доступ к графическим процессорам;
- предоставлять доступ к документам другим людям очень просто.

Это отличное решение для **студентов, специалистов по обработке данных и исследователей в области искусственного интеллекта**. Чтобы узнать больше, посмотрите [ознакомительное видео](#) или начните работу с инструментом ниже.

Активация Windows

Создание безопасного подключения

Введите здесь текст для поиска

11:59

28.09.2022

☒ ☐

```
Task_1.cs* BoomBullet.cs
Assembly-CSharp Task_1 Start()
1 using UnityEngine;
2
3 Скрипт Unity | Ссылки: 0
4 public class Task_1 : MonoBehaviour
5 {
6     // Start is called before the first frame update
7     Сообщение Unity | Ссылки: 0
8     void Start()
9     {
10         Debug.Log("Hello World");
11     }
12 }
13
14
```

Обозреватель решений

Обозреватель решений — поиск (Ctrl+)

- Решение "Tree" (проекты: 13 из 13)
  - Assembly-CSharp
  - Assembly-CSharp.Player
  - Unity.CollabProxy.Editor
  - Unity.Rider.Editor
  - Unity.TextMeshPro
  - Unity.TextMeshPro.Editor
  - Unity.Timeline
  - Unity.Timeline.Editor
  - Unity.VSCode.Editor
  - UnityEditor.TestRunner
  - UnityEditor.UI
  - UnityEngine.TestRunner**
  - UnityEngine.UI

Обозреватель тестов

0 0 0

Поиск в обозревателе тестов

Выполните сборку решения для обнаружения всех доступных тестов. Щелкните "Выполнить все", чтобы построить, обнаружить и выполнить все тесты в решении.

Примечание. Обнаружение тестов в проектах UWP доступно только при нажатии кнопки "Выполнить все".

Активация Windows

Чтобы активировать Windows, перейдите в раздел "Параметры".

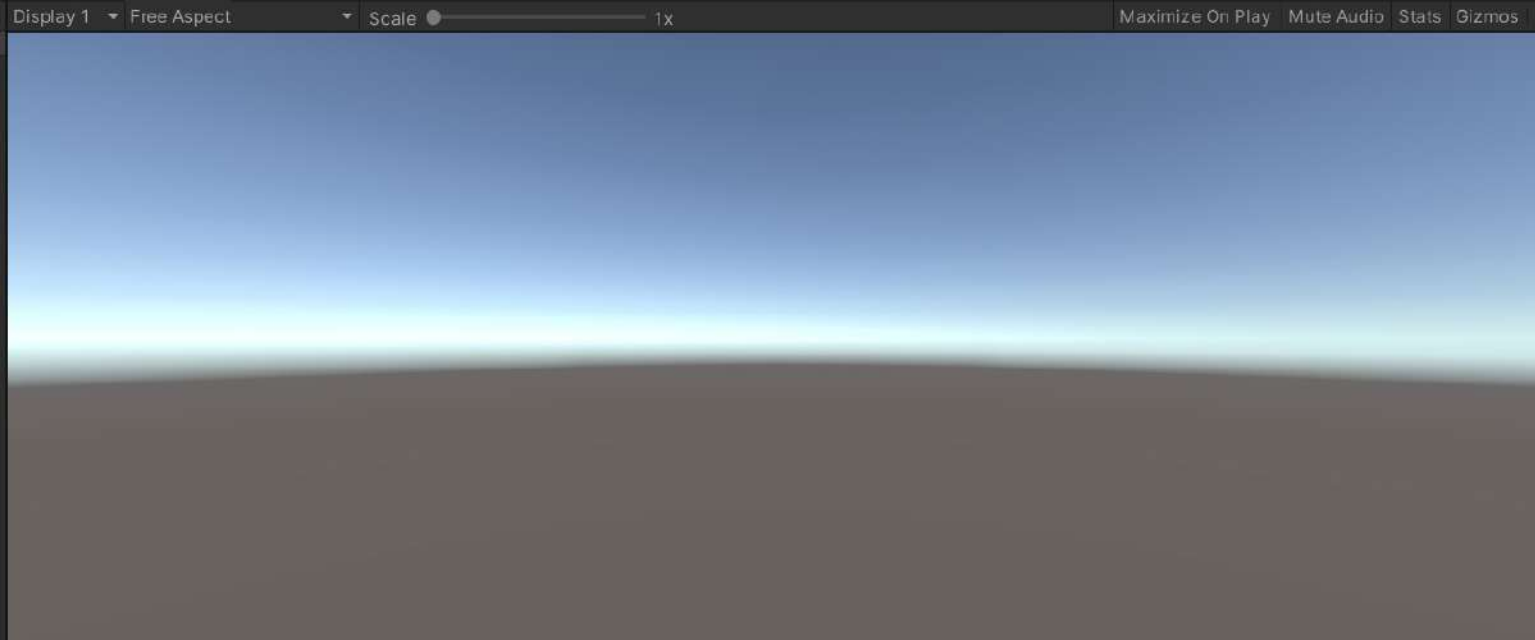
Обозреватель тестов Свойства



Hierarchy Scene Game Asset Store

Display 1 Free Aspect Scale 1x

Untitled\*  
Main Camera  
Directional Light



Project Console

Clear Collapse Clear on Play Clear on Build Error Pause Editor

[12:09:33] Hello World  
UnityEngine.Debug:Log(Object)

Inspector

Task\_1 Import Settings  
Open... Execution Order...

Imported Object

Task\_1

Assembly Information  
Filename Assembly - CSharp.dll

```
using UnityEngine;  
  
public class Task_1 : MonoBehaviour  
{  
    // Start is called before the first frame update  
    void Start()  
    {  
        Debug.Log("Hello World");  
    }  
}
```

Активация Windows

Чтобы активировать Windows, перейдите в раздел  
"Параметры".

Hello World

Auto Generate Lighting Off

Введите здесь текст для поиска





```
x = [3, 21, 22, 34, 54, 34, 55, 67, 89, 99]
x = np.array(x)
y = [2, 22, 24, 65, 79, 82, 55, 138, 158, 199]
y = np.array(y)

plt.scatter(x, y)

def model(a, b, x):
    return a * x + b

def loss_function(a, b, x, y):
    num = len(x)
    prediction = model(a, b, x)
    return (0.5 / num) * (np.square(prediction - y)).sum()

def optimize(a, b, x, y):
    num = len(x)
    prediction = model(a, b, x)
    da = (1.0 / num) * ((prediction - y) * x).sum()
    db = (1.0 / num) * ((prediction - y).sum())
    a = a - lr * da
    b = b - lr * db
    return a, b

def iterate(a, b, x, y, times):
    for i in range(times):
        a, b = optimize(a, b, x, y)
    return a, b

a = np.random.rand(1)
b = np.random.rand(1)
lr = 0.000001

a, b = iterate(a, b, x, y, 1)
prediction = model(a, b, x)
loss = loss_function(a, b, x, y)
print(a, b, loss)
plt.scatter(x, y)
plt.plot(x, prediction)
```













