

ПРЕЗЕНТАЦИЯ К КУРСОВОЙ РАБОТЕ


Подготовил Бочкарев Андрей Исп-241

АКТУАЛЬНОСТЬ ПРОЕКТА

Современный и энергичный ритм жизни отнимает у людей большое количество сил и энергии. В такие моменты людям бывает нужно отвлечься. Именно поэтому мое приложение является актуальным.

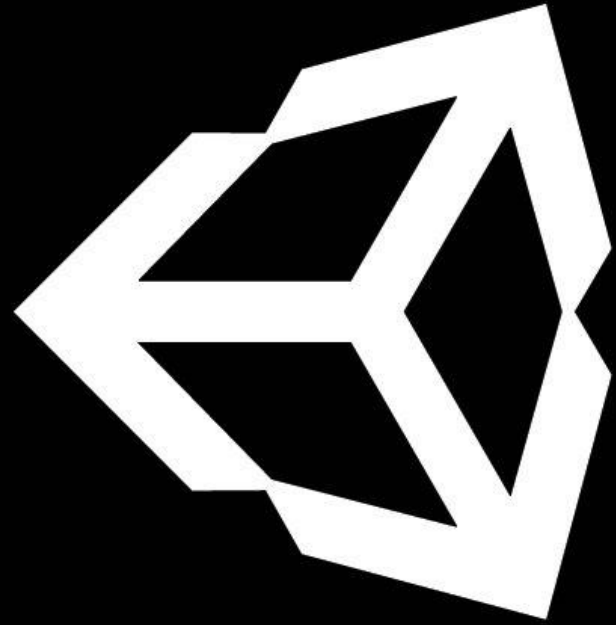
ПОСТАВЛЕННЫЕ ЗАДАЧИ

Главной целью курсовой работы было разработать и протестировать программу для мобильных устройств на операционной платформе Android, которая позволила бы снимать стресс и напряжение в свободное от работы время.



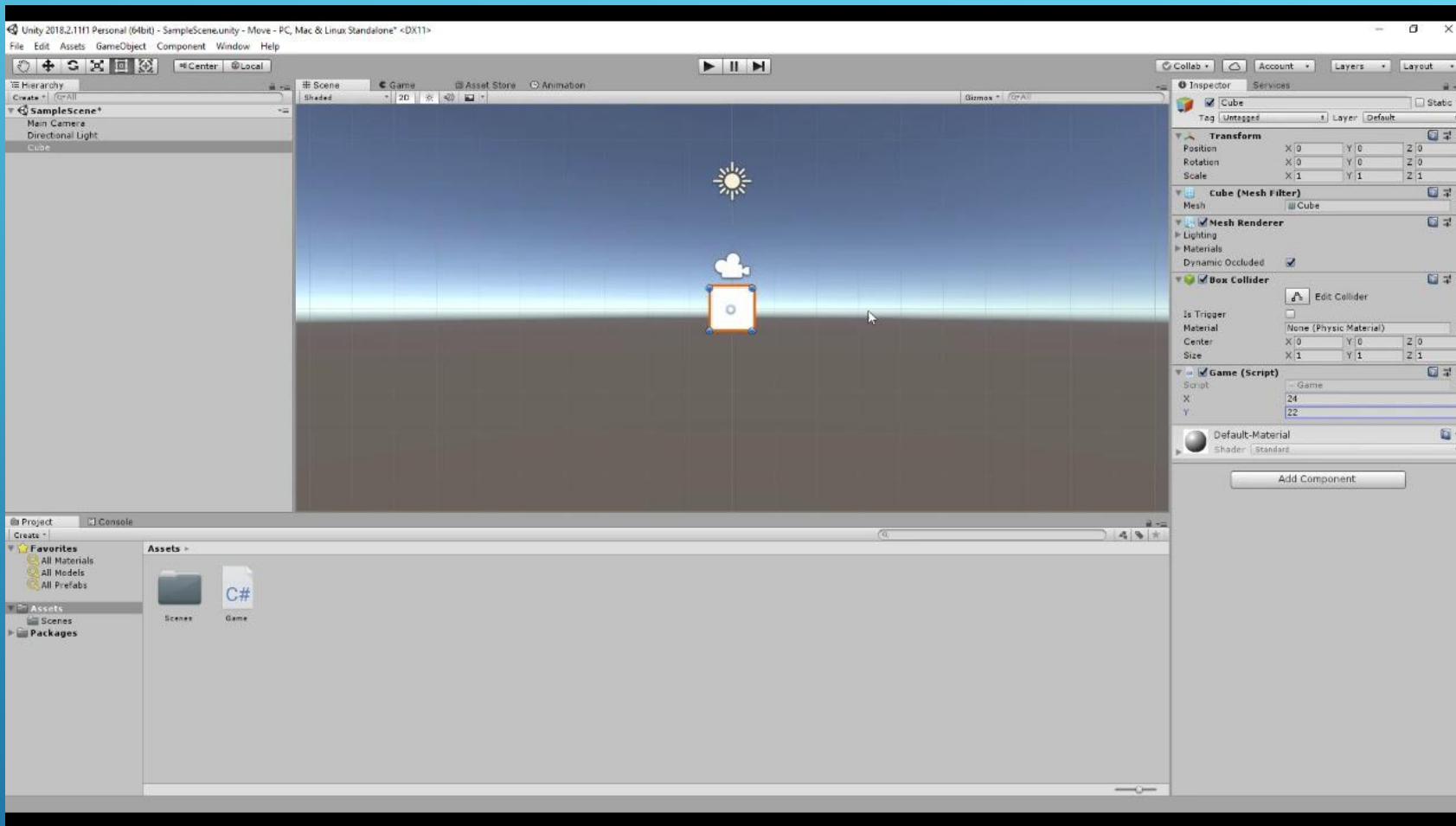
- ▶ **Unity** – больше, чем движок, это среда для разработки компьютерных игр, в которой объединены различные программные средства, используемые при создании ПО





unity









ИНТЕРФЕЙС UNITY





Templates



2D



3D



High Definition RP



Universal Render Pipeline

Settings

Project Name *

Game

Location *

C:\Users\pc



CANCEL

CREATE

Build Settings

Scenes In Build

- ✓ Scenes/SampleScene 0

Add Open Scenes

Platform

- PC, Mac & Linux Standalone
- iOS iOS
- Android**
- WebGL
- Universal Windows Platform
- tvOS tvOS
- PS4 PS4
- PS5 PS5
- Xbox One

Texture Compression: Don't override

ETC2 fallback: 32-bit

Export Project:

Symlink Sources:

Build App Bundle (Google Pla):

Create symbols.zip:

Run Device: Fetching connected devices

Development Build:

Autoconnect Profiler:

Deep Profiling:

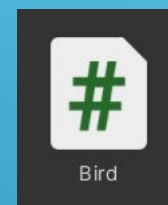
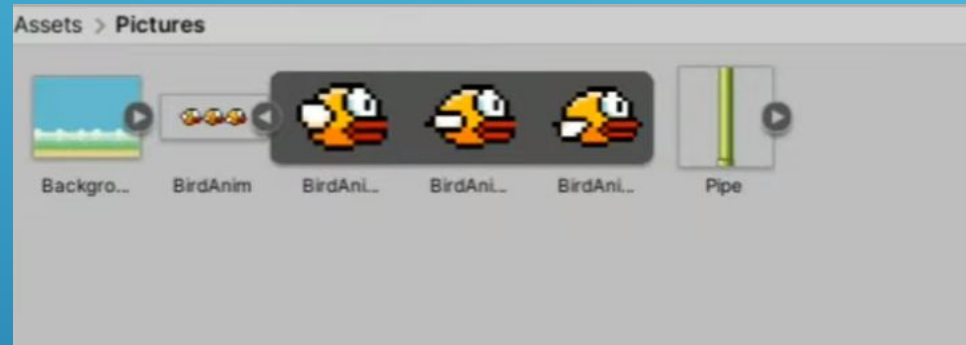
Script Debugging:

Scripts Only Build: Patch Patch And Run

Compression Method: LZ4

[Learn about Unity Cloud Build](#)

Player Settings... Build Build And Run



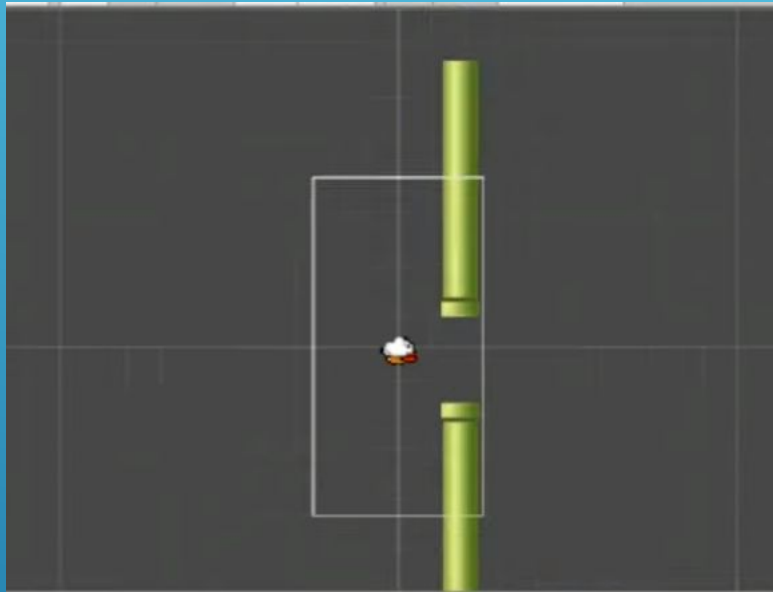
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Bird : MonoBehaviour
{
    public float force;
    Rigidbody2D BirdRigid;

    void Start()
    {
        BirdRigid = GetComponent<Rigidbody2D>();
    }

    void Update()
    {
        if (Input.GetMouseButtonDown(0))
        {
            BirdRigid.velocity = Vector2.up * force;
        }
    }
}
```

ПОСЛЕ ОТКРЫВ ФАЙЛ С КОДОМ ПРОПИСЫВАЕМ В НОВУЮ ПЕРЕМЕННУЮ И ДАЕМ ЕЙ ОБОЗНАЧЕНИЯ. ТАК ЖЕ ПРОПИСЫВАЕМ ФУНКЦИЮ VELOCITY, КОТОРАЯ ПОЗВОЛИТ НАМ ЗАСТАВИТЬ ПО НАЖАТИЮ НА ЭКРАН ПТИЦУ ДВИГАТЬСЯ



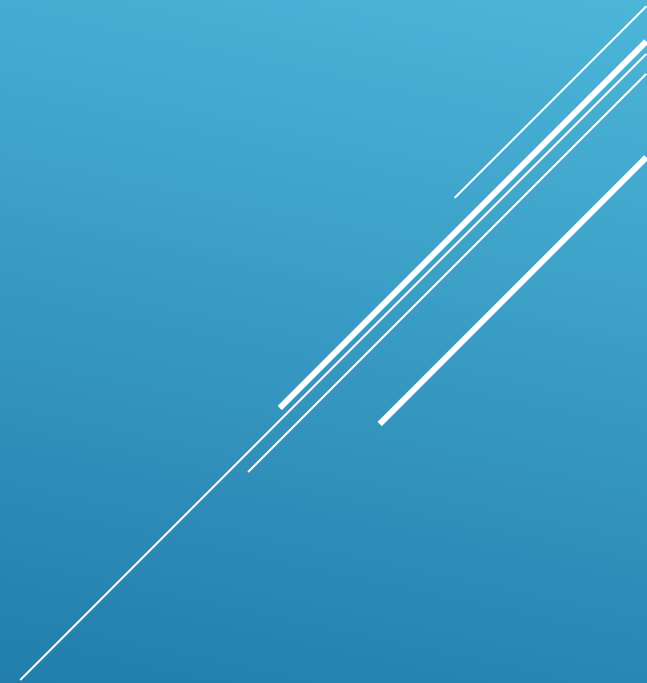
```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class PipeMove : MonoBehaviour
6 {
7     public float speed;
8
9     void Update()
10    {
11        transform.Translate(-speed * Time.deltaTime, 0, 0);
12    }
13 }
14
```

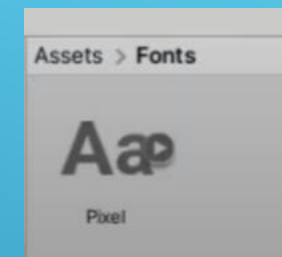
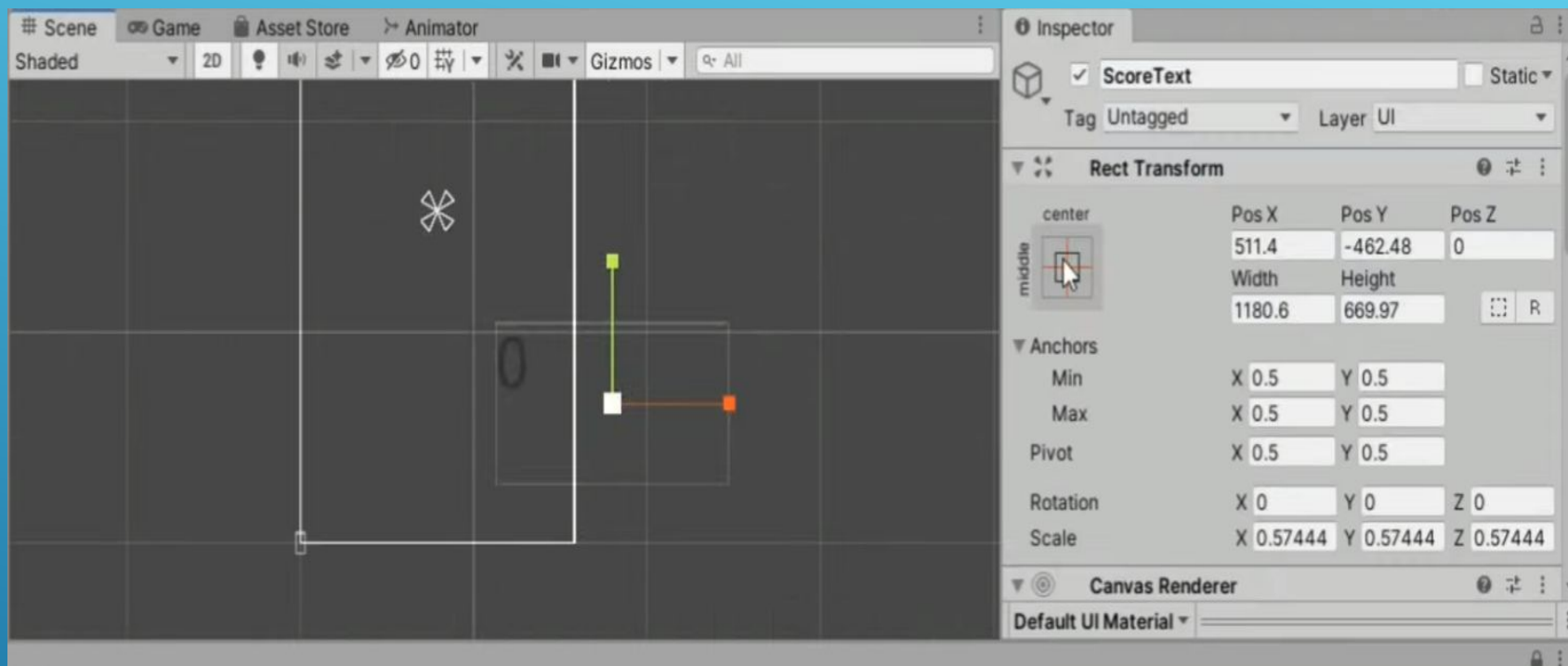


```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class PipeSpawner : MonoBehaviour
6  {
7      public GameObject Pipes;
8
9      void Start()
10     {
11     }
12
13
14     IEnumerator Spawner()
15     {
16         while (true)
17         {
18             yield return new WaitForSeconds(2);
19             float rand = Random.Range(-1f, 4f);
20             Instantiate(Pipes, new Vector3(2, rand, 0), Quaternion.identity);
21         }
22     }
23 }
24
```

```
while (true)
{
    yield return new WaitForSeconds(2);
    float rand = Random.Range(-1f, 4f);
    GameObject newPipes = Instantiate(Pipes, new Vector3(2, rand, 0), Quaternion.identity);
    Destroy(newPipes, 10);
}
```

ДАЛЕЕ СОЗДАЕМ СКРИПТ ДЛЯ ПОЯВЛЕНИЯ ТРУБ.

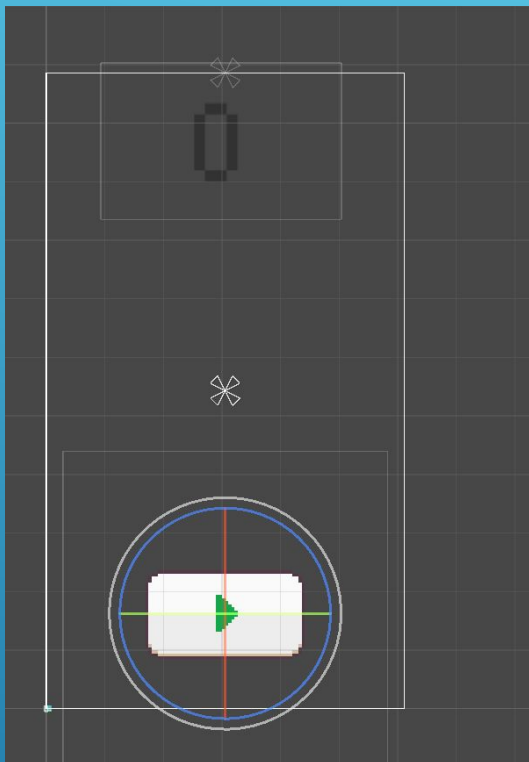




ДАЛЕЕ СОЗДАДИМ СЧЕТЧИК ОЧКОВ, ТОГО СКОЛЬКО ТРУБ ПРОЛЕТЕЛ ИГРОК.
ДЛЯ ЭТОГО СОЗДАЕМ ТЕКСТОВЫЙ ОБЪЕКТ (CANVAS).

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5
6 public class Score : MonoBehaviour
7 {
8     public int score;
9     public Text scoreText;
10
11     void Start()
12     {
13         score = 0;
14     }
15
16     void Update()
17     {
18         scoreText.text = score.ToString();
19     }
20
21     private void OnTriggerEnter2D(Collider2D collision)
22     {
23         if (collision.tag == "Score")
24         {
25             score++;
26         }
27     }
28 }
29
30
```

СОЗДАЕМ СКРИПТ ДЛЯ ПОДСЧЕТА ОЧКОВ И НАЗЫВАЕМ ЕГО SCORE. ПОДКЛЮЧАЕМ БИБЛИОТЕКУ UI



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class Scene : MonoBehaviour
7 {
8     public void Restartlevel()
9     {
10         SceneManager.LoadScene(SceneManager.GetActiveScene().name);
11     }
12 }
13
```

ТАК ЖЕ ДОБАВИМ КНОПКУ ПЕРЕЗАГРУЗКИ, ДЛЯ ЭТОГ ДОБАВИМ ТЕГ «BUTTON» В ИЕРАРХИИ

ЗАКЛЮЧЕНИЕ

В рамках курсовой работы были получены следующие результаты:

Была проведен разбор мобильной ОС Android, а так же основ разработки под нее.