

## §2 Второй отборочный этап

### Задания

Вам дан проект на игровом движке Unity. В проекте содержится 4 сцены, и по каждой необходимо выполнить ряд заданий. После чего запаковать проект архив и отправить сюда.

Внимание: при выполнении заданий не менять теги или имена объектов, иначе задания не будут засчитаны.

Обозначения:

*Имя сцены*

*Объекты сцены*

*Важные условия*

Файл задания:

<https://stepik.org/media/attachments/lesson/59857/halfautoD.rar>

#### Часть 1:

В сцене *Points* реализовать алгоритм автоматического перемещения объекта *Player* по точкам (*Waypoint1*, *Waypoint2*, *Waypoint3*, *Waypoint4*). Подобрать необходимую скорость, чтобы игрок пришел в первую точку в *3 секунды*, во вторую точку в *5 секунд*, в третью в *7 секунд* и в четвертую в *9 секунд*.

#### Часть 2:

В сцене *InputCube* реализовать механизм реакции на нажатия клавиш, реализовать механизм перемещения и стрельбы объекта *Cube*. По нажатию клавиши *W* куб должен сдвинуться на *2 единицы* вперед(по оси *Z*). По нажатию клавиши *S* куб должен сдвинуться на *2 единицы* назад(по оси *Z*). По нажатию клавиши *D* куб должен сдвинуться на *5 единиц* вправо (по оси *X*). По нажатию клавиши *A* куб должен сдвинуться на *3 единицы* влево (по оси *X*). При нажатии клавиши пробел куб должен мгновенно подняться на *2 единицы* вверх. По нажатию левой кнопки мыши из точки *FirePoint* должен появиться префаб *Cylinder* и быть выпущен вперед с начальной скоростью(*Rigidbody.velocity*) *20* под углом *30* (угол поворота самого снаряда) градусов (по оси *X*).

#### Часть 3:

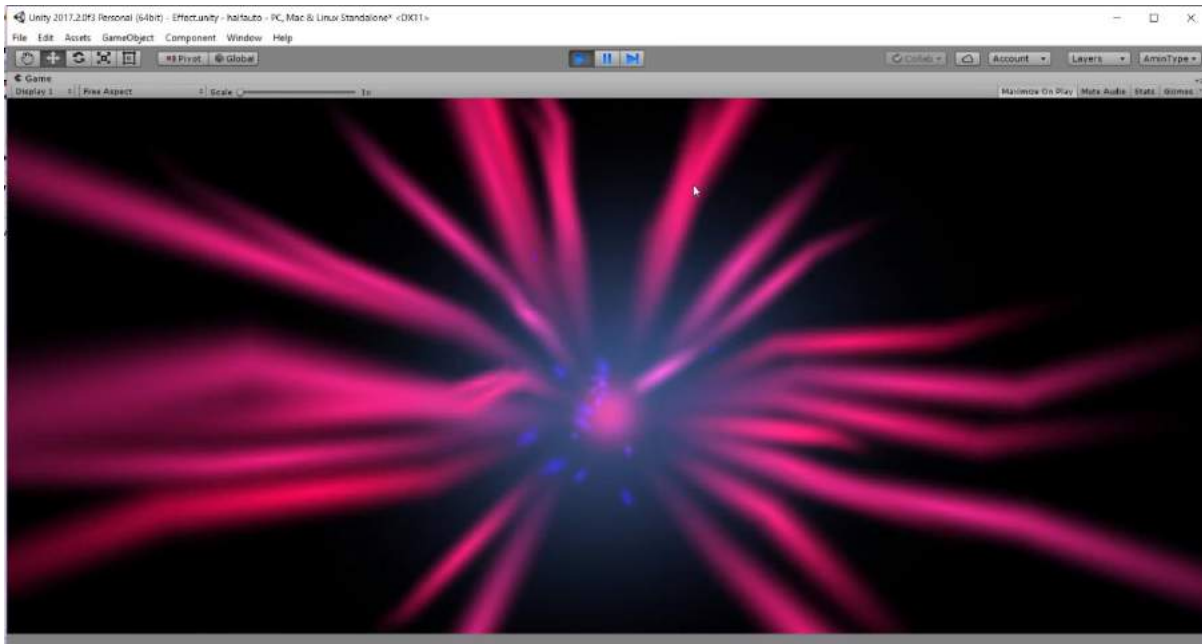
В сцене *Dialog* реализовать диалог двух персонажей с именами *Синий* и *Красный*. Имя должно отображаться в элементе *NameText*, а текст сообщения в *MainText*, кроме того если говорит Синий и имя, и текст должны быть синего цвета, а если красный то красного. Кроме того на расстоянии *0.8* единиц вверх от объекта *RedMan* или *BlueMan* в зависимости от того чья фраза сейчас активна должна появляться сфера *DialogSphere*(Находится в префабах) и уничтожаться, когда говорит другой персонаж. Первое сообщение появляется сразу. Затем

сообщения переключаются нажатием **левой кнопки мыши**. Последовательность сообщений **Синий, Синий, Красный, Синий, Синий**. Написать небольшой осмысленный текст

#### Часть 4:

В сцене **Effect** повторить **систему частиц**, как на следующем видео

<https://fpdl.vimeocdn.com/vimeo-prod-skyfire-std-us/01/3071/9/240357959/860692088.mp4?token=1516800826-0x590afcf7188b0dcb2466747dff2968d98617c0d2>



#### Инструкция по оцениванию

Открыть проект. Импортировать ассет **ntiTest1** (правой кнопкой по вкладке Project - Import Package - Custom Package). Сообщения просматриваются на вкладке Console.

1. Открыть сцену **Points**(папка Scenes). Перетащить на сцену **PointsTest** (папка Prefabs/Test). Запустить сцену(значок Play). Получить четыре сообщения (*0 точка достигнута вовремя, 1 точка достигнута вовремя, 2 точка достигнута вовремя, 3 точка достигнута вовремя*). За каждой сообщении начислить по одному баллу.

**Максимум 4 балла.**

2. Открыть сцену **InputCube**(папка Scenes). Перетащить на сцену **InputCubeTest** (папка Prefabs/Test). Запустить сцену(значок Play). Нажать клавишу **W** - получить сообщение - *действие по W верное*. Нажать клавишу **S** - получить сообщение - *действие по S верное*. Нажать клавишу **A** - получить сообщение - *действие по A верное*. Нажать клавишу **D** - получить сообщение - *действие по D верное*. Нажать клавишу **Пробел** - получить сообщения - *Скорость цилиндра верная и Угол поворота цилиндра верный*. За каждой сообщении начислить по одному баллу.

**Максимум 6 баллов.**

3. Открыть сцену **Dialog**(папка Scenes). Перетащить на сцену **DialogTest** (папка Prefabs/Test). Запустить сцену(значок Play). Получить три сообщения (*Имя соответствует для фразы 1, Цвет названия и текста соответствует для фразы 1, Позиция значка диалога соответствует для фразы 1*). Нажать левую кнопку мыши на вкладке с игрой. Получить еще три аналогичных сообщения со следующим номером. Повторить действия до цифры 5. За каждой сообщении начисляют по 0.5. За осмысленность текста добавить 2 балла.  
**Максимум 10 баллов.**

4. Открыть сцену **Effect**, сравнить эффект с видео либо с префабом **SpaceBloom**(папка Prefabs/Test). За соответствие каждой части эффекта начисляются по 2 балла (1 - частичное соответствие) Части: центр, синее облако, розовые лучи, мигающие звезды.  
**Максимум 8 баллов.**

Ссылка на пакет:

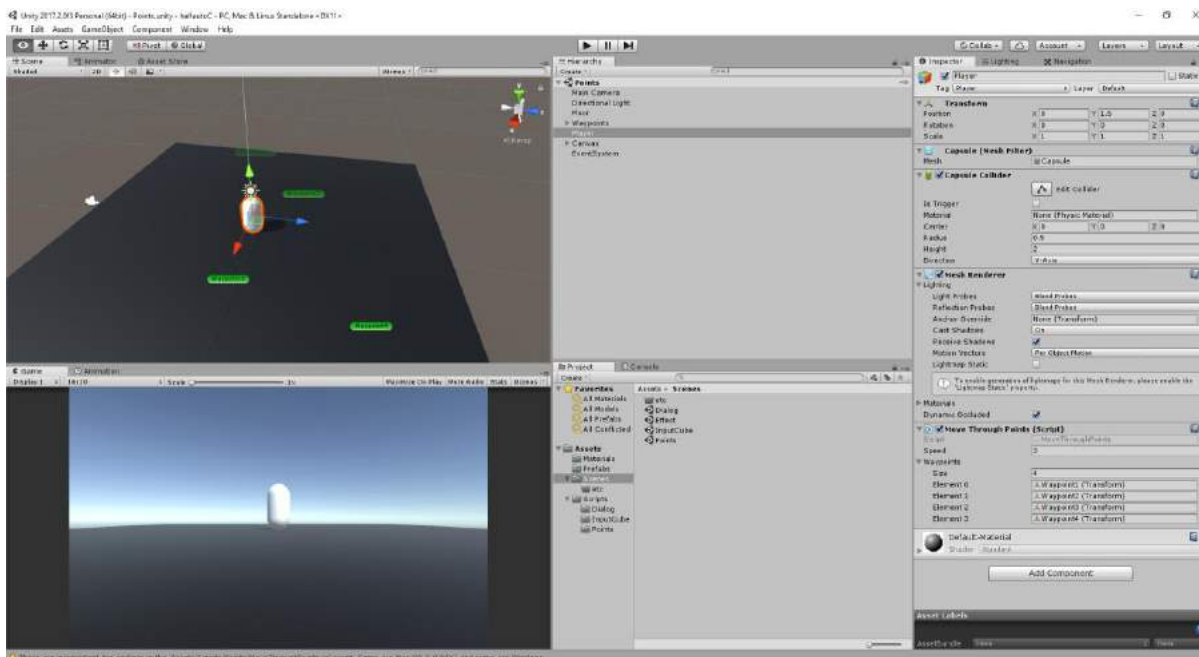
<https://drive.google.com/file/d/0BybRBk0uoucWUXh1Qm05TGhIX1U/view?usp=sharing>

## Разбор решений

### Часть 1:

Перемещаться по точка можно различными способами: двигая объект в нужном направлении стандартными функциями Unity, через NavMesh, или даже телепортируя объект. В примере используется простое перемещение через функцию **Vector3.MoveTowards**, перемещая объект по точкам со скоростью 3.

Скрипт **MoveThroughPoints** прикреплен к объекту **Player**:



### Скрипт 1:

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

using UnityEngine;

```
public class MoveThroughPoints : MonoBehaviour {
```

```
    public float speed = 3;  
    public Transform[] waypoints;
```

```
    Vector3 target;  
    int currentWaypoint = 0;
```

```
    void Start()
```

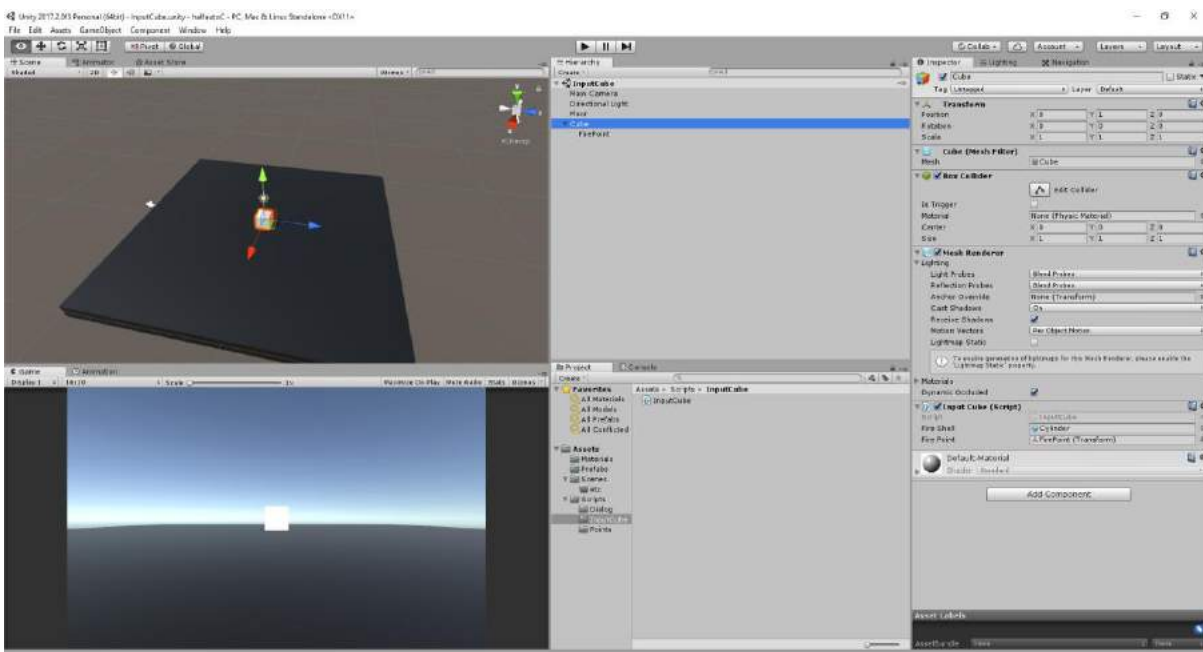
```
    {  
        target = waypoints[currentWaypoint].position;  
    }
```

```
    void Update ()
```

```
    {  
        transform.position = Vector3.MoveTowards(transform.position, target, speed *  
Time.deltaTime);  
        if (transform.position == target)  
        {  
            target = waypoints[++currentWaypoint % waypoints.Length].position;  
        }  
    }  
}
```

## Часть 2:

Стрельбу и перемещение реализует скрипт **InputCube**, прикрепленный к объекту **Cube**. Стрельба реализуется созданием заготовленного объекта в нужной точке и с нужным углом функцией **Instantiate** и заданием ему нужной скорости через компонент физики **Rigidbody**, по нажатию левой кнопки мыши. Перемещение реализовано через **transform.Translate**, по нажатию нужной клавиши на нужное расстояние



## Скрипт 2:

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

public class InputCube : MonoBehaviour {

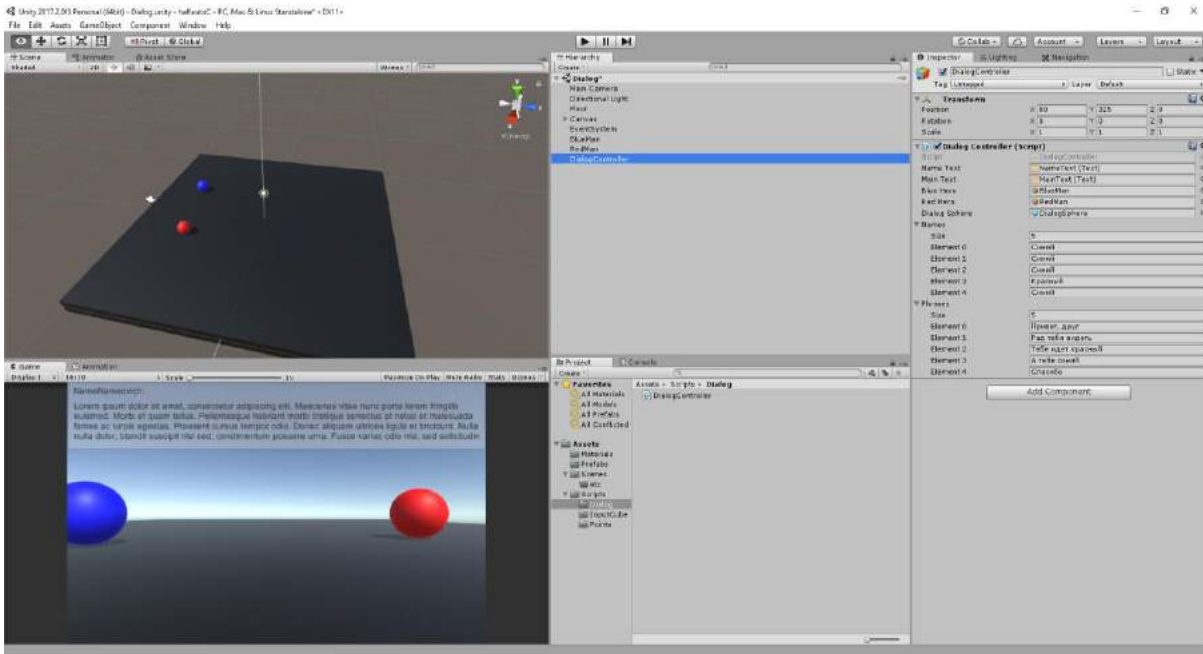
    public GameObject fireShell;
    public Transform firePoint;

    void Update ()
    {
        if (Input.GetMouseButtonDown(0))
        {
            GameObject shell = Instantiate(fireShell, firePoint.position, Quaternion.Euler(30, 0, 0));
            shell.GetComponent<Rigidbody>().velocity = Vector3.forward * 20;
        }

        if (Input.GetKeyDown(KeyCode.W))
        {
            transform.Translate(0,0,2);
        }
        if (Input.GetKeyDown(KeyCode.S))
        {
            transform.Translate(0, 0, -2);
        }
        if (Input.GetKeyDown(KeyCode.D))
        {
            transform.Translate(5, 0, 0);
        }
        if (Input.GetKeyDown(KeyCode.A))
        {
            transform.Translate(-3, 0, 0);
        }
        if (Input.GetKeyDown(KeyCode.Space))
        {
            transform.Translate(0, 3, 0);
        }
    }
}
```

### Часть 3:

В скрипте **DialogController**, прикрепленном к одноименному объекту, по нажатию левой кнопки мыши запускается функция **NextDialog**, которая меняет текст и цвет сообщения и имени игрока, а также перемещает префаб сферы в точку над игроком.



### Скрипт 3:

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

public class DialogController : MonoBehaviour
{
    public Text nameText;
    public Text mainText;
    public GameObject blueHero;
    public GameObject redHero;
    public GameObject dialogSphere;

    public string[] names;
    public string[] phrases;

    private int curentPhraseNumber;
    private GameObject sphere;

    void Start()
    {
        NextDialog();
    }
}
```

```

}

void Update ()
{
    if (Input.GetMouseButtonDown(0) && curentPhraseNumber != phrases.Length)
    {
        NextDialog();
    }
}

private void NextDialog()
{
    Destroy(sphere);
    if (names[curentPhraseNumber] == "Синий")
    {
        nameText.color = Color.blue;
        mainText.color = Color.blue;
        sphere = Instantiate(dialogSphere, blueHero.transform.position + new Vector3(0, 0.8f,
0), Quaternion.identity);
    }
    else if (names[curentPhraseNumber] == "Красный")
    {
        nameText.color = Color.red;
        mainText.color = Color.red;
        sphere = Instantiate(dialogSphere, redHero.transform.position + new Vector3(0, 0.8f,
0), Quaternion.identity);
    }

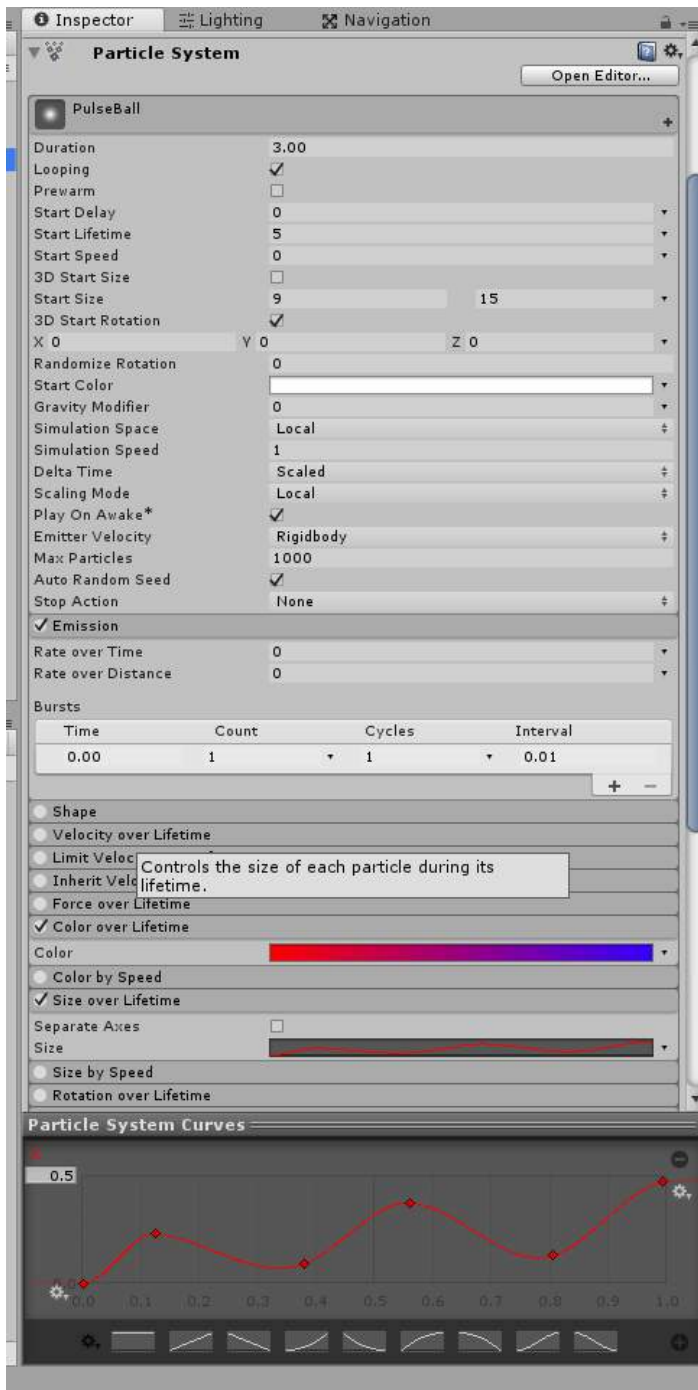
    nameText.text = names[curentPhraseNumber];
    mainText.text = phrases[curentPhraseNumber];
    curentPhraseNumber++;
}
}

```

## Часть 4:

Система частиц состоит из 4х частей.

### PulseBall:





# Stornado:

**Inspector** | Lighting | Navigation

**Stornado**

Duration: 1.00  
Looping:   
Prewarm:   
Start Delay: 0  
Start Lifetime: 0.3 | 1  
Start Speed: 0  
3D Start Size:   
Start Size: 10 | 20  
3D Start Rotation:   
Start Rotation: 0  
Randomize Rotation: 0  
Start Color: [Color Picker]  
Gravity Modifier: 0  
Simulation Space: World  
Simulation Speed: 1  
Delta Time: Scaled  
Scaling Mode: Shape  
Play On Awake\*:   
Emitter Velocity: Rigidbody  
Max Particles: 100  
Auto Random Seed:   
Stop Action: None

Emission

Shape

Shape: Box  
Emit from: Volume

Position: X 0 | Y 0 | Z 0  
Rotation: X 0 | Y 0 | Z 0  
Scale: X 0.5 | Y 1 | Z 0.5

Align To Direction:   
Randomize Direction: 1  
Spherize Direction: 0  
Randomize Position: 0

Velocity over Lifetime

X [Graph] | Y [Graph] | Z [Graph]  
Space: Local

Limit Velocity over Lifetime  
 Inherit Velocity

Force over Lifetime

X 12 | Y 2 | Z 12  
-12 | 25 | -12  
Space: World  
Randomize:

Color over Lifetime

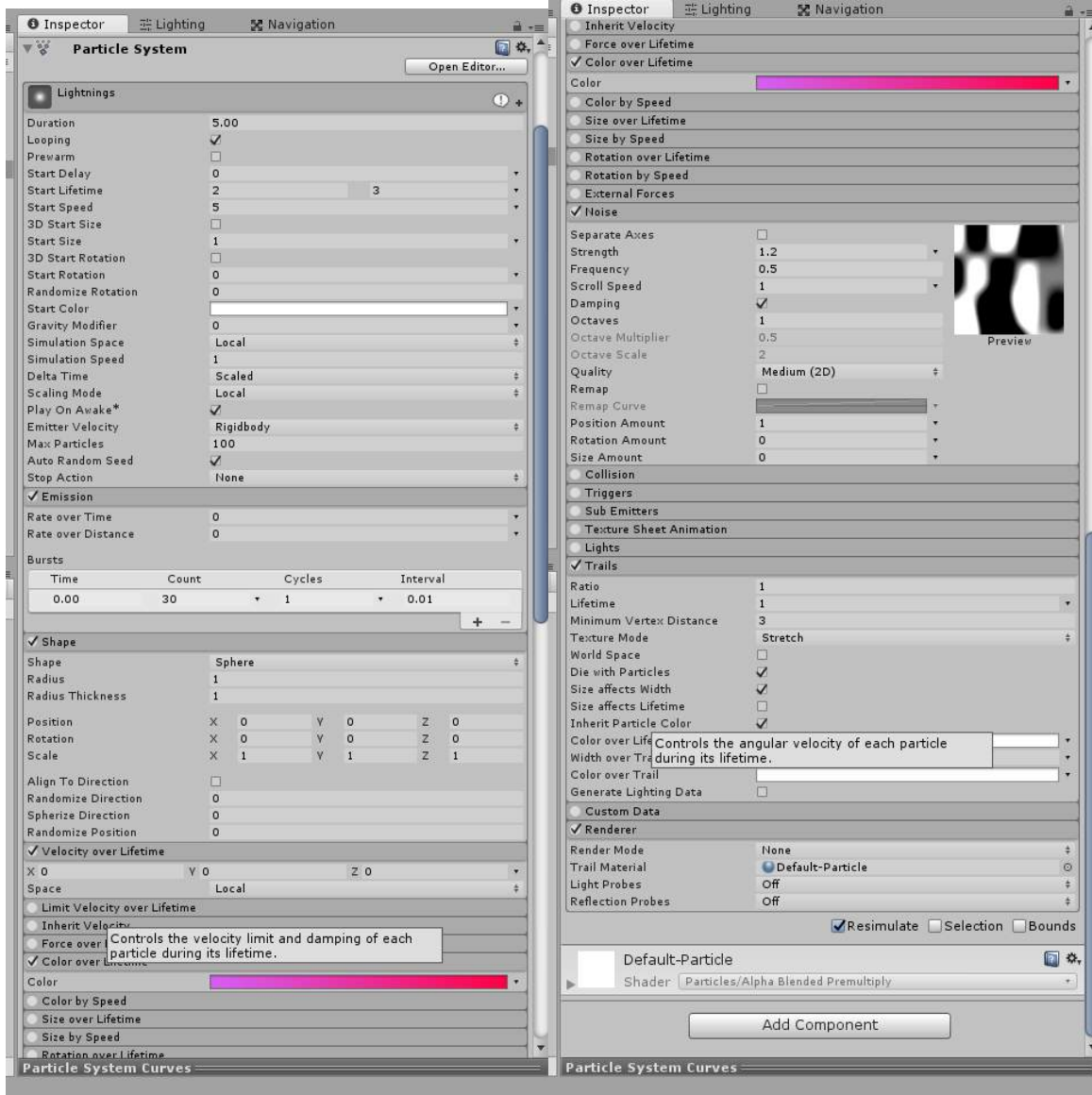
**Particle System Curves**

Z

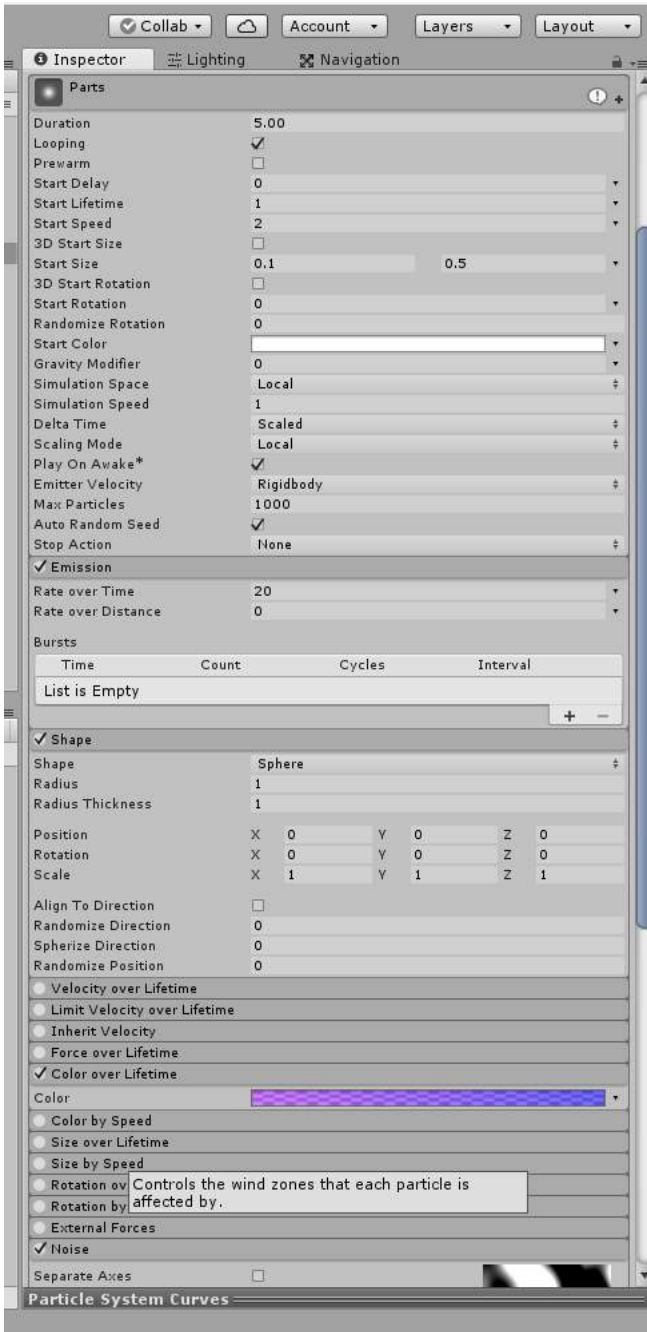
80  
50  
-50

0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

# Lightnings:



## Parts:



## Файл с решением:

<https://drive.google.com/file/d/1PCNuI8a5k4LyYBtBVfrXiLCIZ8uifVfZ/view?usp=sharing>