Донецкий Национальный Технический Университет

Лабораторная работа № 4

«Проектирование и разработка игровой программы Lightman средствами Unity»

Выполнил:

ст. группы ИПЗ -13

Лысенко А. С.

Проверил:

ст. препод. ПМИ

Павловский Е.В.

Покровск 2017

1. **Цель расчетной работы:** Научиться разрабатывать игровое приложение средствами Unity.

2. **Задачи:**

Разработать игровое приложение Lightman средствами Unity.

3. **Содержание отчета**

Введение

Задание к расчетной работе

1 Создание спрайтов главного героя, бонусов, препятствий и другие (согласно варианту задания).

2 Создание и настройка объектов.

3 Добавление к объектам событий и действий.

4 Создание и настройка комнат.

Выводы

Перечень ссылок

# Введение

1. Жанр и идея будущей игры.

Жанр - аркада, игра будет характеризоваться коротким по времени прохождением, но интенсивным игровым процессом, основным направлением которого будет развитие ловкости и реакции игрока. Идея игры заключается в том, чтобы пройти уровни, при этом получить как можно меньше повреждений и собрать как можно больше ресурсов.

1. Краткое описание целевой аудитории игры.

Образование Они интересуются новинками на рынке игр и стремятся поиграть в old school игры. Ещё пользователи ценят дизайн и простоту управления. Игра подходит для любого вида образования, политических взглядов, религиозной лояльности, социального статуса, место проживания, семейного положения.

Поведенческая часть профиля потребителя.

Мотивация получить увлечение от игры, круг интересов игры и их простота, привычки контроль над процессом. Проблемы игра короткая, надежды на выход продолжения.

1. Анализ лицензируемого материала и/или ограничивающего контента.

Игра разрабатывалась для обеих полов, возраст 6-56, с уровнем дохода «низкий», «средний» и «выше среднего».

1. Основные особенности игры USP (Unique Selling Points).

Основная особенность игры заключается в её простоте и понятии для игровой аудитории.

1. Сюжет и последовательность событий игры.

Сюжет игры достаточно тривиальный – сбор ресурсов, выживание и как можно быстрее добраться до цели. События расположены таким образом - сбор и выживание происходят одновременно, следом выполняется миссия приближения к цели, для перехода на следующий уровень.

6. Анализ системных требований.

Windows 2000, Windows 7, Windows Vista, Windows XP  
800MHz processor speed  
256 MB RAM  
32 MB VRAM 16 or 32-bit Video Card  
16 bit DirectX compatible sound card  
DirectX 7.0

# Задание к расчетной работе

Разработать игровое приложение Lightman средствами Unity.

Фактическая часть профиля потребителя.

Игра разрабатывалась для обеих полов, возраст 6-56, с уровнем дохода «низкий», «средний» и «выше среднего». Образование Они интересуются новинками на рынке игр и стремятся поиграть в old school игры. Ещё пользователи ценят дизайн и простоту управления. Игра подходит для любого вида образования, политических взглядов, религиозной лояльности, социального статуса, место проживания, семейного положения.

Поведенческая часть профиля потребителя.

Мотивация получить увлечение от игры, круг интересов игры и их простота, привычки контроль над процессом. Проблемы игра короткая, надежды на выход продолжения.

Требования к функциям игрового приложения.

1. Подсчет очков.

2. Движение персонажей.

# 1. Создание спрайтов главного героя, ресурсов, препятствий и другие.

Создаём папку, в которую будем помещать изображения в данном случае «Graphics»

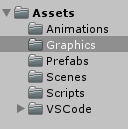


Рис. 1.1 - Папки проекта

Затем заносим в папку «Graphics» изображение и устанавливаем для него следующие параметры

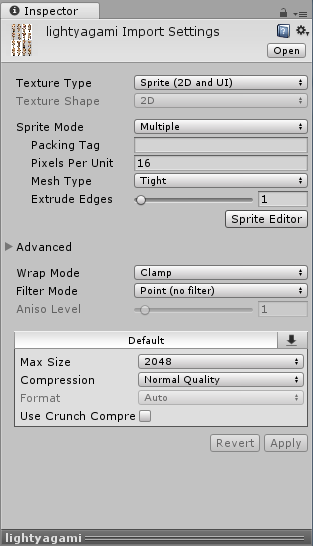


Рис. 1.2 – Окно настроек «Sprites»

Для того чтобы нарезать спрайты нажмем кнопку «Sprite Editor» в «Sprite Editor» находим кнопку «Slice» нажимаем, появляется следующее окно.

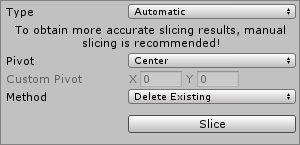


Рис.1.3 – Окно настроек «Sprite Editor»

В этом окне в значении «Type» устанавливаем параметр «Automatic», если нам заранее известно на какие пиксели делить спрайт можем выбрать «Grid By Cell Size»



Рис. 1.4 - Окно настроек «Sprite Editor»

Далее получаем изображение, деленное на спрайты, как мы можем заметить, внутри изображения вокруг каждой картинке появилась рамка, которая показывает границы спрайтов.

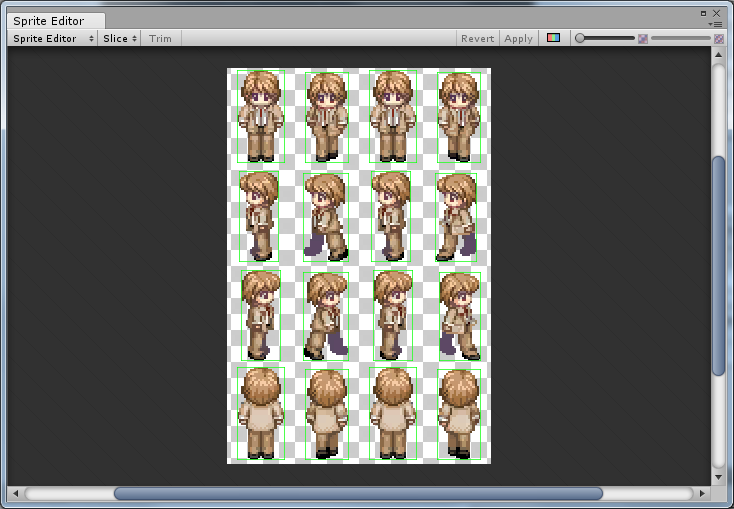


Рис. 1.5 – Окно деления изображения на спрайты

Для создания ресурсов переносим изображение из папки на сцену, затем создаём папку с «Prefabs» создаём в ней «Prefab», через правую клавишу мыши, «Create», «Prefab» и перетаскиваем изображение на «Prefab». Затем «Prefabs» переместить на игровую сцену.

Затем устанавливаем свойства объекта.

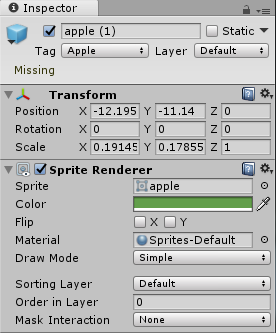
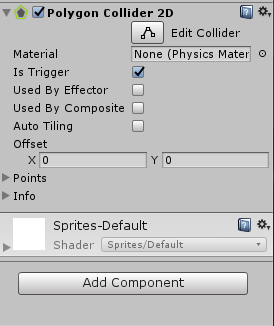
 

Рис. 1.6 – Окно свойств «Prefab» ресурсов

Для создания препятствий в данном случае стен комнаты используем спрайты. Для удобного соединения спрайтов нажимаем клавишу V в сцене, затем соединяем спрайты[1].

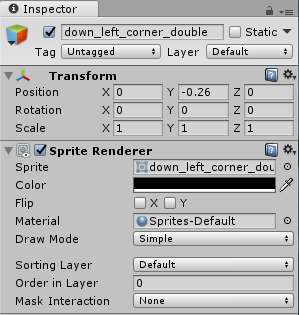
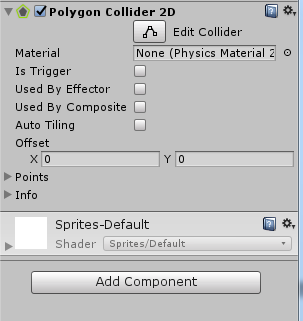
 

Рис. 1.7 – Окно свойств комнаты «Maze»

# 2. Создание и настройка объектов

1. Разработка и описание объектных моделей игровых сущностей (включая аватаров, если они предусмотрены концептом игры).
2. Справочник игровых сущностей.

Cущность Characters

Таблица 1 - члены типа Transform

|  |  |  |
| --- | --- | --- |
| Описание | Тип | Имя |
| Позиция | float | Position |
| Вращение | float | Rotation |
| Масштаб | float | Scale |

Таблица 2 - члены типа Sprite Renderer[2]

|  |  |  |
| --- | --- | --- |
| Описание | Тип | Имя |
| Cпрайт | png | Sprite |
| Цвет(rgb) | int | Color |
| Флип | bool | Flip |
| Материал | Default | Material |
| Отображение | Simple | DrawMode |
| Сортировка слоёв | Default | SortLayer |
| Описание | Тип | Имя |
| Порядок слоёв | int | OrderLayer |
| Взаимодействие масок | None | MaskInteraction |

Таблица 3 - члены типа Script

|  |  |  |
| --- | --- | --- |
| Описание | Тип | Имя |
| Название свойства | Тип в языке программирования | ID свойства |

Таблица 4 - члены типа Rigidbody 2D

|  |  |  |
| --- | --- | --- |
| Описание | Тип | Имя |
| Тип объекта | Dynamic | BodyType |
| Симуляция | bool | Simulated |
| Авто масса | bool | UseAutoMass |
| Описание | Тип | Имя |
| Масса | float | Mass |
| Линейное перемещение | float | LinearDrag |
| Угловое перемещение | float | AngularDrag |
| Гравитационная скала | float | GravityScale |
| Интерполирование | None | Interpolate |
| Спящий режим | Start Awake | SleepMode |
| Обнаружение столкновений | Discrete | CollDetect |
| Стоп-кадр | bool | FreezePosition |
| Замораживание вращения | bool | FreezeRotation |

Таблица 5 - члены типа Polygon Collider 2D

|  |  |  |
| --- | --- | --- |
| Описание | Тип | Имя |
| Триггер | bool | IsTrigger |
| Используется эффектором | bool | UsedEffector |
| Используется композитом | bool | UsedComposit |
| Авто черепица | bool | AutoTiling |
| Смещение | float | Offset |

Таблица 6 - члены типа Animator[2]

|  |  |  |
| --- | --- | --- |
| Описание | Тип | Имя |
| Аватар | None | Avatar |
| Применить движение корня | bool | ApplyRootMotion |
| Мод обновления | Normal | UpdateMode |
| Мод отборки | Always Animate | CullingMode |

Сущности Resources и Maps levels

Таблица 1 - члены типа Transform

|  |  |  |
| --- | --- | --- |
| Описание | Тип | Имя |
| Позиция | float | Position |
| Вращение | float | Rotation |
| Масштаб | float | Scale |

Таблица 2 - члены типа Sprite Renderer

|  |  |  |
| --- | --- | --- |
| Описание | Тип | Имя |
| Cпрайт | png | Sprite |
| Цвет(rgb) | int | Color |
| Флип | bool | Flip |
| Материал | Default | Material |
| Отображение | Simple | DrawMode |
| Сортировка слоёв | Default | SortLayer |
| Порядок слоёв | int | OrderLayer |
| Взаимодействие масок | None | MaskInteraction |

Таблица 3 - члены типа Rigidbody 2D

|  |  |  |
| --- | --- | --- |
| Описание | Тип | Имя |
| Тип объекта | Dynamic | BodyType |
| Симуляция | bool | Simulated |
| Авто масса | bool | UseAutoMass |
| Масса | float | Mass |
| Линейное перемещение | float | LinearDrag |
| Угловое перемещение | float | AngularDrag |
| Гравитационная скала | float | GravityScale |

|  |  |  |
| --- | --- | --- |
| Описание | Тип | Имя |
| Интерполирование | None | Interpolate |
| Спящий режим | Start Awake | SleepMode |
| Обнаружение столкновений | Discrete | CollDetect |
| Стоп-кадр | bool | FreezePosition |
| Замораживание вращения | bool | FreezeRotation |

Таблица 4 - члены типа Polygon Collider 2D

|  |  |  |
| --- | --- | --- |
| Описание | Тип | Имя |
| Триггер | bool | IsTrigger |
| Описание | Тип | Имя |
| Используется эффектором | bool | UsedEffector |
| Используется композитом | bool | UsedComposit |
| Авто черепица | bool | AutoTiling |
| Смещение | float | Offset |

Сущность Camera

Таблица 1 - члены типа Transform

|  |  |  |
| --- | --- | --- |
| Описание | Тип | Имя |
| Позиция | float | Position |
| Вращение | float | Rotation |
| Масштаб | float | Scale |

Таблица 2 - члены типа Camera2D[3]

|  |  |  |
| --- | --- | --- |
| Описание | Тип | Имя |
| Очистить флаги | Scybox | ClearFlags |
| Задний план | int | Background |
| Маска отбора | Everything | CullingMask |
| Проекция | Orthographic | Projection |
| Размер | int | Size |
| Плоскость отсечения | float | ClippingPlanes |
| Глубина | int | Depth |
| Путь рендеринга | Use graphics settings | RenderingPath |
| Целевая текстура | None | TargetTexture |
| Отборка колизии | bool | OcclusionCulling |
| HDR | bool | AllowHDR |
| Описание | Тип | Имя |
| MSAA | bool | AllowMSAA |
| Отображение цели | Display1 | TargetDisplay |
| Слой GUI | bool | GUILayer |
| Яркость слоя | bool | FlareLayer |
| Аудио слушатель | bool | AudioListener |

1. Описание контента игры.



Рисунок 2.1 – Объектная модель Characters



Рисунок 2.2 – Объектная модель Resources and Maps Levels



Рисунок 2.3 – Объектная модель Camera

# 3. Добавление к объектам событий и действий

Добавление к объектам событий и действий происходит с помощью скриптов, для начала создадим папку «Scripts» в корне проекта.

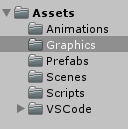
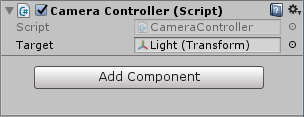


Рис. 3.1 – Окно папок проекта

Затем добавляем в неё скрипту, находим объект, на который хотим повесить скрипт и с помощью кнопки «Add Component» добавляем скрипт на объект.



# Выводы

1.Расчет сроков и бюджета проекта.

Срок разработки и вывода на рынок: 2 месяца  
Фонд оплаты труда: 3 000 грн.  
Аренда: 2 000 грн.  
Организация рабочего пространства (мебель, техника, лицензии и т.д.): 30 000 грн.  
Расходы на рекламу: 50 000 грн.  
Прочие расходы (серверное оборудование, ведение юридического лица и др.): 1 500 грн.  
Итого: 86 500 гривен.

2. Описание способов монетизации игры.

Игра платформенная –по модели freemium зарабатывают как правило так: 10 уровней игры можно скачать и пройти бесплатно, далее предлагается скачать игру за 1-3 доллара, далее в игре за деньги можно купить какую-нибудь крутую штуку, - например, дополнительную жизнь, которая помогает дольше выживать и вы проходите проблемный уровень. В платформенных играх (данный тип игр профессионалы рынка еще называют free to play) вы играете бесплатно сколько угодно, но за ускоренный рост персонажа можно заплатить, и вы автоматически становитесь в более выигрышной ситуации, чем ваши соперники. Доходы лучших игр в отрасли измеряются миллионами долларов в год.

3. Механизм повышения виртуальности игры.

Виртуальность игры будет заключаться в том, что пользователь может выложить свой результат у себя на страничке в соц. сетях, а другие люди замечая это, если захотят могут скачать игру себе и так же поиграть в неё.

# Перечень ссылок

1. <https://docs.unity3d.com/ru/530/Manual> - создание комнаты

2. <https://www.youtube.com/watch?v=tjxKxZsofdk> – создание анимации

3.<https://www.youtube.com/watch?v=DkjrXLKv4JU&list=PLczcTSDLfwNSHdPggLGWUrujHY2on0Iew> – создание save points и др.

# Листинг программы C#

BotController.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class BotController : MonoBehaviour {

public GameObject point\_0, point\_1, point\_2, point\_3, point\_4, point\_5, point\_6, point\_7,

point\_8, point\_9, point\_10, point\_11, point\_12, point\_13, point\_14, point\_15, point\_16,

point\_17, point\_18, point\_19, point\_20, point\_21, point\_22, point\_23, point\_24, point\_25,

point\_26, point\_27, point\_28, point\_29, point\_30, point\_31, point\_32, point\_33, point\_34,

point\_35, point\_36, point\_37, point\_38, point\_39, point\_40, point\_41, point\_42, point\_43,

point\_44, point\_45, point\_46, point\_47, point\_48, point\_49, point\_50, point\_51;

public float playerSpeed = 9;

Transform[] all\_Point = new Transform[52];

public Vector2 curSavePos;

private Vector3 target\_Pos;

private Animator anim;

public int i = 0;

public int Count = 1;

float distance;

public Transform target;

void Start() {

anim = this.GetComponent<Animator>();

point\_0 = GameObject.FindGameObjectWithTag("Point\_0");

point\_1 = GameObject.FindGameObjectWithTag("Point\_1");

point\_2 = GameObject.FindGameObjectWithTag("Point\_2");

point\_3 = GameObject.FindGameObjectWithTag("Point\_3");

point\_4 = GameObject.FindGameObjectWithTag("Point\_4");

point\_5 = GameObject.FindGameObjectWithTag("Point\_5");

point\_6 = GameObject.FindGameObjectWithTag("Point\_6");

point\_7 = GameObject.FindGameObjectWithTag("Point\_7");

point\_8 = GameObject.FindGameObjectWithTag("Point\_8");

point\_9 = GameObject.FindGameObjectWithTag("Point\_9");

point\_10 = GameObject.FindGameObjectWithTag("Point\_10");

point\_11 = GameObject.FindGameObjectWithTag("Point\_11");

point\_12 = GameObject.FindGameObjectWithTag("Point\_12");

point\_13 = GameObject.FindGameObjectWithTag("Point\_13");

point\_14 = GameObject.FindGameObjectWithTag("Point\_14");

point\_15 = GameObject.FindGameObjectWithTag("Point\_15");

point\_16 = GameObject.FindGameObjectWithTag("Point\_16");

point\_17 = GameObject.FindGameObjectWithTag("Point\_17");

point\_18 = GameObject.FindGameObjectWithTag("Point\_18");

point\_19 = GameObject.FindGameObjectWithTag("Point\_19");

point\_20 = GameObject.FindGameObjectWithTag("Point\_20");

point\_21 = GameObject.FindGameObjectWithTag("Point\_21");

point\_22 = GameObject.FindGameObjectWithTag("Point\_22");

point\_23 = GameObject.FindGameObjectWithTag("Point\_23");

point\_24 = GameObject.FindGameObjectWithTag("Point\_24");

point\_25 = GameObject.FindGameObjectWithTag("Point\_25");

point\_26 = GameObject.FindGameObjectWithTag("Point\_26");

point\_27 = GameObject.FindGameObjectWithTag("Point\_27");

point\_28 = GameObject.FindGameObjectWithTag("Point\_28");

point\_29 = GameObject.FindGameObjectWithTag("Point\_29");

point\_30 = GameObject.FindGameObjectWithTag("Point\_30");

point\_31 = GameObject.FindGameObjectWithTag("Point\_31");

point\_32 = GameObject.FindGameObjectWithTag("Point\_32");

point\_33 = GameObject.FindGameObjectWithTag("Point\_33");

point\_34 = GameObject.FindGameObjectWithTag("Point\_34");

point\_35 = GameObject.FindGameObjectWithTag("Point\_35");

point\_36 = GameObject.FindGameObjectWithTag("Point\_36");

point\_37 = GameObject.FindGameObjectWithTag("Point\_37");

point\_38 = GameObject.FindGameObjectWithTag("Point\_38");

point\_39 = GameObject.FindGameObjectWithTag("Point\_39");

point\_40 = GameObject.FindGameObjectWithTag("Point\_40");

point\_41 = GameObject.FindGameObjectWithTag("Point\_41");

point\_42 = GameObject.FindGameObjectWithTag("Point\_42");

point\_43 = GameObject.FindGameObjectWithTag("Point\_43");

point\_44 = GameObject.FindGameObjectWithTag("Point\_44");

point\_45 = GameObject.FindGameObjectWithTag("Point\_45");

point\_46 = GameObject.FindGameObjectWithTag("Point\_46");

point\_47 = GameObject.FindGameObjectWithTag("Point\_47");

point\_48 = GameObject.FindGameObjectWithTag("Point\_48");

point\_49 = GameObject.FindGameObjectWithTag("Point\_49");

point\_50 = GameObject.FindGameObjectWithTag("Point\_50");

point\_51 = GameObject.FindGameObjectWithTag("Point\_51");

all\_Point[0] = point\_0.transform;

all\_Point[1] = point\_1.transform;

all\_Point[2] = point\_2.transform;

all\_Point[3] = point\_3.transform;

all\_Point[4] = point\_4.transform;

all\_Point[5] = point\_5.transform;

all\_Point[6] = point\_6.transform;

all\_Point[7] = point\_7.transform;

all\_Point[8] = point\_8.transform;

all\_Point[9] = point\_9.transform;

all\_Point[10] = point\_10.transform;

all\_Point[11] = point\_11.transform;

all\_Point[12] = point\_12.transform;

all\_Point[13] = point\_13.transform;

all\_Point[14] = point\_14.transform;

all\_Point[15] = point\_15.transform;

all\_Point[16] = point\_16.transform;

all\_Point[17] = point\_17.transform;

all\_Point[18] = point\_18.transform;

all\_Point[19] = point\_19.transform;

all\_Point[20] = point\_20.transform;

all\_Point[21] = point\_21.transform;

all\_Point[22] = point\_22.transform;

all\_Point[23] = point\_23.transform;

all\_Point[24] = point\_24.transform;

all\_Point[25] = point\_25.transform;

all\_Point[26] = point\_26.transform;

all\_Point[27] = point\_27.transform;

all\_Point[28] = point\_28.transform;

all\_Point[29] = point\_29.transform;

all\_Point[30] = point\_30.transform;

all\_Point[31] = point\_31.transform;

all\_Point[32] = point\_32.transform;

all\_Point[33] = point\_33.transform;

all\_Point[34] = point\_34.transform;

all\_Point[35] = point\_35.transform;

all\_Point[36] = point\_36.transform;

all\_Point[37] = point\_37.transform;

all\_Point[38] = point\_38.transform;

all\_Point[39] = point\_39.transform;

all\_Point[40] = point\_40.transform;

all\_Point[41] = point\_41.transform;

all\_Point[42] = point\_42.transform;

all\_Point[43] = point\_43.transform;

all\_Point[44] = point\_44.transform;

all\_Point[45] = point\_45.transform;

all\_Point[46] = point\_46.transform;

all\_Point[47] = point\_47.transform;

all\_Point[48] = point\_48.transform;

all\_Point[49] = point\_49.transform;

all\_Point[50] = point\_50.transform;

all\_Point[51] = point\_51.transform;

}

void Update()

{

target\_Pos = all\_Point[i].transform.position;

transform.Translate(Vector3.Normalize(target\_Pos - transform.position) \* Time.deltaTime \* playerSpeed);

distance = Vector3.Distance(target\_Pos, transform.position);

if (distance < 0.5f)

{

if (i < all\_Point.Length - 1)

{

i++;

}

else

{

i = 0;

}

}

}

}

CameraController.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class CameraController : MonoBehaviour {

public Transform target = null;

int distance = -10;

float lift = 1.5f;

void Update () {

transform.position = new Vector3(0, lift, distance) + target.position;

transform.LookAt(target);

if (Input.GetKeyDown(KeyCode.Escape))

{

Application.LoadLevel("InfoMenu");

}

}

}

InfoMenu.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class InfoMenu : MonoBehaviour

{

public int button;

public GUISkin GUISkin;

public Vector2 curSavePos;

// Use this for initialization

void Start()

{

button = 0;

}

// Update is called once per frame

void Update()

{

}

void OnGUI()

{

GUI.skin = GUISkin;

GUI.Box(new Rect(Screen.width / 2 - 100,

Screen.height / 2 - 100, 200, 70), "Меню игры");

if (button == 0)

{

if (GUI.Button(new Rect(Screen.width / 2 - 80, Screen.height / 2 - 70, 70, 30), "Да"))

{

button = 1;

}

if (GUI.Button(new Rect(Screen.width / 2 + 10, Screen.height / 2 - 70, 70, 30), "Нет"))

{

button = 2;

}

if (button == 1)

{

Application.LoadLevel("MainMenu");

}

if (button == 2)

{

Application.LoadLevel("Level1");

}

}

}

}

LoseMenu.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class LoseMenu : MonoBehaviour {

public Transform target = null;

int distance = -10;

float lift = 1.5f;

void Update()

{

transform.position = new Vector3(0, lift, distance) + target.position;

transform.LookAt(target);

if (Input.GetKeyDown(KeyCode.Escape))

{

Application.LoadLevel("MainMenu");

}

}

}

MainMenu.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class MainMenu : MonoBehaviour

{

public GUISkin GUISkin;

public int window;

// Use this for initialization

void Start () {

window = 0;

}

void OnGUI()

{

GUI.skin = GUISkin;

if (window == 0)

{

GUI.Box(new Rect(Screen.width / 2 - 100,

Screen.height / 2 - 100, 200, 145), "Меню игры");

if (GUI.Button(new Rect(Screen.width / 2 - 90, Screen.height / 2 - 70, 180, 30), "Начать игру"))

{

window = 1;

}

if (GUI.Button(new Rect(Screen.width / 2 - 90, Screen.height / 2 - 40, 180, 30), "Об игре"))

{

window = 3;

}

if (GUI.Button(new Rect(Screen.width / 2 - 90, Screen.height / 2 - 10, 180, 30), "Выход"))

{

window = 4;

}

if(window == 1)

{

Debug.Log("Уровень загружен");

Application.LoadLevel("Level1");

}

if(window == 2)

{

}

if(window == 3)

{

Application.LoadLevel("InfoGame");

}

}

if(window == 4)

{

GUI.Box(new Rect(Screen.width / 2 - 100, Screen.height / 2 - 100, 225, 70), "Вы действительно хотите выйти?");

if(GUI.Button(new Rect(Screen.width / 2 - 75, Screen.height / 2 - 70, 70, 30), "Да"))

{

Application.Quit();

}

if (GUI.Button(new Rect(Screen.width / 2 + 25, Screen.height / 2 - 70, 70, 30), "Нет"))

{

Application.LoadLevel("MainMenu");

}

}

}

}

PlayerController.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class PlayerController : MonoBehaviour {

public float playerSpeed = 7.5f;

int score = 0;

int hp = 3;

public int button;

public GUISkin GUISkin;

public GameObject gameObject;

//public Texture2D AppleTexture = null;

public Vector2 curSavePos;

public Texture2D AppleTexture = null;

public Texture2D HP = null;

public GUIStyle myStyle = new GUIStyle();

private Animator anim;

public int appleCount = 1;

void Start()

{

anim = this.GetComponent<Animator>();

button = 0;

}

void Update() {

float transV = Input.GetAxis("Vertical") \* playerSpeed \* Time.deltaTime;

float transH = Input.GetAxis("Horizontal") \* playerSpeed \* Time.deltaTime;

if (transH > 0)

{

anim.SetBool("Right", true);

anim.SetBool("Left", false);

anim.SetBool("Up", false);

anim.SetBool("Down", false);

}

if (transH < 0)

{

anim.SetBool("Left", true);

anim.SetBool("Right", false);

anim.SetBool("Up", false);

anim.SetBool("Down", false);

}

if (transV < 0)

{

anim.SetBool("Down", true);

anim.SetBool("Right", false);

anim.SetBool("Up", false);

anim.SetBool("Left", false);

}

if (transV > 0)

{

anim.SetBool("Up", true);

anim.SetBool("Right", false);

anim.SetBool("Left", false);

anim.SetBool("Down", false);

}

transform.Translate(new Vector3(transH, transV, 0));

/\* if (hp >= 3)

hp = 3;

if (hp <= 0)

hp = 0;

if (hp == 2)

hp = 2;\*/

if (score >= 40)

{

gameObject = GameObject.FindGameObjectWithTag("Ryuk");

Destroy(gameObject);

}

}

void OnTriggerEnter2D(Collider2D other)

{

if (other.tag == "Apple")

{

score += 1;

Debug.Log("Яблок собрано: " + score);

Destroy(other.gameObject);

//print ("Яблок собрано: " + appleCount++);

if (score == 105)

Application.LoadLevel("Win");

}

else if (other.tag == "killPoint" || other.tag == "Misa")

{

transform.position = curSavePos;

//Application.LoadLevel("Level1");

hp -= 1;

if (hp == 0)

Application.LoadLevel("Lose");

}

else if (score >= 40 && other.tag == "Wall")

{

Application.LoadLevel("Win");

// Destroy(other.gameObject);

}

else if (score >= 40)

{

other.tag = "Ryuk";

Destroy(other.gameObject);

}

}

private void OnGUI()

{

GUI.skin = GUISkin;

myStyle.fontSize = 35;

myStyle.normal.textColor = Color.black;

// GUI.Label(new Rect(10, 40, AppleTexture.width, AppleTexture.height), AppleTexture);

GUI.Label(new Rect(55, 15, 0, 0), ": " + score, myStyle);

GUI.Label(new Rect(0, 0, 60, 60), AppleTexture);

GUI.Label(new Rect(5, 60, 50, 50), HP);

GUI.Label(new Rect(55, 65, 0, 0), ": " + hp, myStyle);

/\* GUI.skin = GUISkin;

GUI.Box(new Rect(Screen.width / 2 - 100, Screen.height / 2 - 100, 200, 70), "Меню игры");

if (GUI.Button(new Rect(Screen.width / 2 - 80, Screen.height / 2 - 70, 70, 30), "Да"))

{

button = 1;

}

if (GUI.Button(new Rect(Screen.width / 2 + 10, Screen.height / 2 - 70, 70, 30), "Нет"))

{

button = 2;

}

if (button == 1)

{

Application.LoadLevel("MainMenu");

}

if (button == 2)

{

button = -1;

}

if (button == -1)

{

}\*/

}

}

WayPoints.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class WayPoints : MonoBehaviour

{

private Animator anim;

public GameObject point\_0, point\_1, point\_2, point\_3, point\_4, point\_5, point\_6, point\_7, point\_8,

point\_9, point\_10, point\_11, point\_12, point\_13, point\_14, point\_15, point\_16, point\_17, point\_18,

point\_19, point\_20, point\_21, point\_22, point\_23, point\_24, point\_25, point\_26, point\_27, point\_28,

point\_29, point\_30, point\_31, point\_32, point\_33, point\_34, point\_35, point\_36, point\_37, point\_38,

point\_39, point\_40, point\_41, point\_42, point\_43, point\_44, point\_45, point\_46, point\_47, point\_48,

point\_49, point\_50, point\_51, point\_52, point\_53, point\_54, point\_55, point\_56, point\_57, point\_58,

point\_59, point\_60, point\_61, point\_62, point\_63, point\_64, point\_65, point\_66, point\_67, point\_68,

point\_69, point\_70, point\_71, point\_72, point\_73, point\_74, point\_75, point\_76, point\_77, point\_78,

point\_79, point\_80, point\_81, point\_82, point\_83, point\_84, point\_85, point\_86, point\_87, point\_88,

point\_89, point\_90, point\_91, point\_92, point\_93, point\_94, point\_95, point\_96, point\_97, point\_98,

point\_99, point\_100, point\_101;

Transform[] all\_Point = new Transform[102];

private Vector3 target\_Pos;

public int i = 0;

public float speed\_move = 9;

void Start ()

{

anim = this.GetComponent<Animator> ();

/\*for(i = 0; i <= 102; i++)

{

point[i] = GameObject.FindGameObjectWithTag("Point[" + i + "]");

all\_Point[i] = point[i].transform;

}\*/

point\_0 = GameObject.FindGameObjectWithTag("Point0");

point\_1 = GameObject.FindGameObjectWithTag("Point1");

point\_2 = GameObject.FindGameObjectWithTag("Point2");

point\_3 = GameObject.FindGameObjectWithTag("Point3");

point\_4 = GameObject.FindGameObjectWithTag("Point4");

point\_5 = GameObject.FindGameObjectWithTag("Point5");

point\_6 = GameObject.FindGameObjectWithTag("Point6");

point\_7 = GameObject.FindGameObjectWithTag("Point7");

point\_8 = GameObject.FindGameObjectWithTag("Point8");

point\_9 = GameObject.FindGameObjectWithTag("Point9");

point\_10 = GameObject.FindGameObjectWithTag("Point10");

point\_11 = GameObject.FindGameObjectWithTag("Point11");

point\_12 = GameObject.FindGameObjectWithTag("Point12");

point\_13 = GameObject.FindGameObjectWithTag("Point13");

point\_14 = GameObject.FindGameObjectWithTag("Point14");

point\_15 = GameObject.FindGameObjectWithTag("Point15");

point\_16 = GameObject.FindGameObjectWithTag("Point16");

point\_17 = GameObject.FindGameObjectWithTag("Point17");

point\_18 = GameObject.FindGameObjectWithTag("Point18");

point\_19 = GameObject.FindGameObjectWithTag("Point19");

point\_20 = GameObject.FindGameObjectWithTag("Point20");

point\_21 = GameObject.FindGameObjectWithTag("Point21");

point\_22 = GameObject.FindGameObjectWithTag("Point22");

point\_23 = GameObject.FindGameObjectWithTag("Point23");

point\_24 = GameObject.FindGameObjectWithTag("Point24");

point\_25 = GameObject.FindGameObjectWithTag("Point25");

point\_26 = GameObject.FindGameObjectWithTag("Point26");

point\_27 = GameObject.FindGameObjectWithTag("Point27");

point\_28 = GameObject.FindGameObjectWithTag("Point28");

point\_29 = GameObject.FindGameObjectWithTag("Point29");

point\_30 = GameObject.FindGameObjectWithTag("Point30");

point\_31 = GameObject.FindGameObjectWithTag("Point31");

point\_32 = GameObject.FindGameObjectWithTag("Point32");

point\_33 = GameObject.FindGameObjectWithTag("Point33");

point\_34 = GameObject.FindGameObjectWithTag("Point34");

point\_35 = GameObject.FindGameObjectWithTag("Point35");

point\_36 = GameObject.FindGameObjectWithTag("Point36");

point\_37 = GameObject.FindGameObjectWithTag("Point37");

point\_38 = GameObject.FindGameObjectWithTag("Point38");

point\_39 = GameObject.FindGameObjectWithTag("Point39");

point\_40 = GameObject.FindGameObjectWithTag("Point40");

point\_41 = GameObject.FindGameObjectWithTag("Point41");

point\_42 = GameObject.FindGameObjectWithTag("Point42");

point\_43 = GameObject.FindGameObjectWithTag("Point43");

point\_44 = GameObject.FindGameObjectWithTag("Point44");

point\_45 = GameObject.FindGameObjectWithTag("Point45");

point\_46 = GameObject.FindGameObjectWithTag("Point46");

point\_47 = GameObject.FindGameObjectWithTag("Point47");

point\_48 = GameObject.FindGameObjectWithTag("Point48");

point\_49 = GameObject.FindGameObjectWithTag("Point49");

point\_50 = GameObject.FindGameObjectWithTag("Point50");

point\_51 = GameObject.FindGameObjectWithTag("Point51");

point\_52 = GameObject.FindGameObjectWithTag("Point52");

point\_53 = GameObject.FindGameObjectWithTag("Point53");

point\_54 = GameObject.FindGameObjectWithTag("Point54");

point\_55 = GameObject.FindGameObjectWithTag("Point55");

point\_56 = GameObject.FindGameObjectWithTag("Point56");

point\_57 = GameObject.FindGameObjectWithTag("Point57");

point\_58 = GameObject.FindGameObjectWithTag("Point58");

point\_59 = GameObject.FindGameObjectWithTag("Point59");

point\_60 = GameObject.FindGameObjectWithTag("Point60");

point\_61 = GameObject.FindGameObjectWithTag("Point61");

point\_62 = GameObject.FindGameObjectWithTag("Point62");

point\_63 = GameObject.FindGameObjectWithTag("Point63");

point\_64 = GameObject.FindGameObjectWithTag("Point64");

point\_65 = GameObject.FindGameObjectWithTag("Point65");

point\_66 = GameObject.FindGameObjectWithTag("Point66");

point\_67 = GameObject.FindGameObjectWithTag("Point67");

point\_68 = GameObject.FindGameObjectWithTag("Point68");

point\_69 = GameObject.FindGameObjectWithTag("Point69");

point\_70 = GameObject.FindGameObjectWithTag("Point70");

point\_71 = GameObject.FindGameObjectWithTag("Point71");

point\_72 = GameObject.FindGameObjectWithTag("Point72");

point\_73 = GameObject.FindGameObjectWithTag("Point73");

point\_74 = GameObject.FindGameObjectWithTag("Point74");

point\_75 = GameObject.FindGameObjectWithTag("Point75");

point\_76 = GameObject.FindGameObjectWithTag("Point76");

point\_77 = GameObject.FindGameObjectWithTag("Point77");

point\_78 = GameObject.FindGameObjectWithTag("Point78");

point\_79 = GameObject.FindGameObjectWithTag("Point79");

point\_80 = GameObject.FindGameObjectWithTag("Point80");

point\_81 = GameObject.FindGameObjectWithTag("Point81");

point\_82 = GameObject.FindGameObjectWithTag("Point82");

point\_83 = GameObject.FindGameObjectWithTag("Point83");

point\_84 = GameObject.FindGameObjectWithTag("Point84");

point\_85 = GameObject.FindGameObjectWithTag("Point85");

point\_86 = GameObject.FindGameObjectWithTag("Point86");

point\_87 = GameObject.FindGameObjectWithTag("Point87");

point\_88 = GameObject.FindGameObjectWithTag("Point88");

point\_89 = GameObject.FindGameObjectWithTag("Point89");

point\_90 = GameObject.FindGameObjectWithTag("Point90");

point\_91 = GameObject.FindGameObjectWithTag("Point91");

point\_92 = GameObject.FindGameObjectWithTag("Point92");

point\_93 = GameObject.FindGameObjectWithTag("Point93");

point\_94 = GameObject.FindGameObjectWithTag("Point94");

point\_95 = GameObject.FindGameObjectWithTag("Point95");

point\_96 = GameObject.FindGameObjectWithTag("Point96");

point\_97 = GameObject.FindGameObjectWithTag("Point97");

point\_98 = GameObject.FindGameObjectWithTag("Point98");

point\_99 = GameObject.FindGameObjectWithTag("Point99");

point\_100 = GameObject.FindGameObjectWithTag("Point100");

point\_101 = GameObject.FindGameObjectWithTag("Point101");

all\_Point[0] = point\_0.transform;

all\_Point[1] = point\_1.transform;

all\_Point[2] = point\_2.transform;

all\_Point[3] = point\_3.transform;

all\_Point[4] = point\_4.transform;

all\_Point[5] = point\_5.transform;

all\_Point[6] = point\_6.transform;

all\_Point[7] = point\_7.transform;

all\_Point[8] = point\_8.transform;

all\_Point[9] = point\_9.transform;

all\_Point[10] = point\_10.transform;

all\_Point[11] = point\_11.transform;

all\_Point[12] = point\_12.transform;

all\_Point[13] = point\_13.transform;

all\_Point[14] = point\_14.transform;

all\_Point[15] = point\_15.transform;

all\_Point[16] = point\_16.transform;

all\_Point[17] = point\_17.transform;

all\_Point[18] = point\_18.transform;

all\_Point[19] = point\_19.transform;

all\_Point[20] = point\_20.transform;

all\_Point[21] = point\_21.transform;

all\_Point[22] = point\_22.transform;

all\_Point[23] = point\_23.transform;

all\_Point[24] = point\_24.transform;

all\_Point[25] = point\_25.transform;

all\_Point[26] = point\_26.transform;

all\_Point[27] = point\_27.transform;

all\_Point[28] = point\_28.transform;

all\_Point[29] = point\_29.transform;

all\_Point[30] = point\_30.transform;

all\_Point[31] = point\_31.transform;

all\_Point[32] = point\_32.transform;

all\_Point[33] = point\_33.transform;

all\_Point[34] = point\_34.transform;

all\_Point[35] = point\_35.transform;

all\_Point[36] = point\_36.transform;

all\_Point[37] = point\_37.transform;

all\_Point[38] = point\_38.transform;

all\_Point[39] = point\_39.transform;

all\_Point[40] = point\_40.transform;

all\_Point[41] = point\_41.transform;

all\_Point[42] = point\_42.transform;

all\_Point[43] = point\_43.transform;

all\_Point[44] = point\_44.transform;

all\_Point[45] = point\_45.transform;

all\_Point[46] = point\_46.transform;

all\_Point[47] = point\_47.transform;

all\_Point[48] = point\_48.transform;

all\_Point[49] = point\_49.transform;

all\_Point[50] = point\_50.transform;

all\_Point[51] = point\_51.transform;

all\_Point[52] = point\_52.transform;

all\_Point[53] = point\_53.transform;

all\_Point[54] = point\_54.transform;

all\_Point[55] = point\_55.transform;

all\_Point[56] = point\_56.transform;

all\_Point[57] = point\_57.transform;

all\_Point[58] = point\_58.transform;

all\_Point[59] = point\_59.transform;

all\_Point[60] = point\_60.transform;

all\_Point[61] = point\_61.transform;

all\_Point[62] = point\_62.transform;

all\_Point[63] = point\_63.transform;

all\_Point[64] = point\_64.transform;

all\_Point[65] = point\_65.transform;

all\_Point[66] = point\_66.transform;

all\_Point[67] = point\_67.transform;

all\_Point[68] = point\_68.transform;

all\_Point[69] = point\_69.transform;

all\_Point[70] = point\_70.transform;

all\_Point[71] = point\_71.transform;

all\_Point[72] = point\_72.transform;

all\_Point[73] = point\_73.transform;

all\_Point[74] = point\_74.transform;

all\_Point[75] = point\_75.transform;

all\_Point[76] = point\_76.transform;

all\_Point[77] = point\_77.transform;

all\_Point[78] = point\_78.transform;

all\_Point[79] = point\_79.transform;

all\_Point[80] = point\_80.transform;

all\_Point[81] = point\_81.transform;

all\_Point[82] = point\_82.transform;

all\_Point[83] = point\_83.transform;

all\_Point[84] = point\_84.transform;

all\_Point[85] = point\_85.transform;

all\_Point[86] = point\_86.transform;

all\_Point[87] = point\_87.transform;

all\_Point[88] = point\_88.transform;

all\_Point[89] = point\_89.transform;

all\_Point[90] = point\_90.transform;

all\_Point[91] = point\_91.transform;

all\_Point[92] = point\_92.transform;

all\_Point[93] = point\_93.transform;

all\_Point[94] = point\_94.transform;

all\_Point[95] = point\_95.transform;

all\_Point[96] = point\_96.transform;

all\_Point[97] = point\_97.transform;

all\_Point[98] = point\_98.transform;

all\_Point[99] = point\_99.transform;

all\_Point[100] = point\_100.transform;

all\_Point[101] = point\_101.transform;

}

void Update ()

{

target\_Pos = all\_Point[i].transform.position;

transform.Translate(Vector3.Normalize(target\_Pos - transform.position)\*Time.deltaTime\*speed\_move);

float distance = Vector3.Distance(target\_Pos, transform.position);

if(distance < 0.5f)

{

if(i < all\_Point.Length - 1)

{

i++;

}

else

{

i = 0;

//Destroy(gameObject);

}

}

}

/\*void OnTriggerEnter2D(Collider2D other)

{

if (i == 2)

{

anim.SetBool("Right", true);

anim.SetBool("Left", false);

anim.SetBool("Up", false);

anim.SetBool("Down", false);

}

if (i == 1)

{

anim.SetBool("Up", true);

anim.SetBool("Right", false);

anim.SetBool("Left", false);

anim.SetBool("Down", false);

}

if (i == 0)

{

anim.SetBool("Left", true);

anim.SetBool("Right", false);

anim.SetBool("Up", false);

anim.SetBool("Down", false);

}

if (i == 3)

{

anim.SetBool("Down", true);

anim.SetBool("Right", false);

anim.SetBool("Up", false);

anim.SetBool("Left", false);

}

//print(i);

}\*/

}

WinMenu.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class WinMenu : MonoBehaviour {

public Transform target = null;

int distance = -10;

float lift = 1.5f;

void Update()

{

transform.position = new Vector3(0, lift, distance) + target.position;

transform.LookAt(target);

if (Input.GetKeyDown(KeyCode.Return))

{

Application.LoadLevel("MainMenu");

}

}

}