

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
(ФГАОУ ВО «ЮФУ»)

Институт компьютерных технологий и информационной безопасности

ОТЧЁТ №3
(финальный)

о проделанной работе команды №6 – «IT Лапки»
на тему: **«Разработка игрового приложения на Unity»**
по дисциплине: **«Введение в инженерную деятельность»**

Выполнили:

студент КТбо 1-7 Бекезин С.А.
студент КТбо 1-7 Прокопович Н.А.
студент КТбо 1-7 Комов В.А.
студент КТбо 1-7 Никитин Н.С.
студент КТбо 1-7 Баженов Д.В.
студент КТбо 1-7 Фомин М.А.
студент КТбо 1-7 Каневская А.Э.
студент Ктбо 1-9 Щелокова А.В.

Проверил:

старший преподаватель кафедры ВТ
Переверзев В.А.

СОДЕРЖАНИЕ

| | |
|--------------------------------------|----|
| РЕЗУЛЬТАТЫ К ПОСЛЕДНЕМУ ОТЧЕТУ | 2 |
| 1.1 Работа с локацией | 2 |
| 1.2 Финальная головоломка | 3 |
| 1.3 Каменный монстр..... | 4 |
| 1.4 Дополнения к меню | 5 |
| 1.5 Планетарная карта..... | 6 |
| 1.6 Журнал прогресса | 8 |
| 1.7 Подсказки для новичков | 9 |
| 1.8 Работа художника | 9 |
| 1.9 Игровой код..... | 11 |
| ЗАКЛЮЧЕНИЕ | 19 |

1 РЕЗУЛЬТАТЫ К ПОСЛЕДНЕМУ ОТЧЕТУ

1.1 Работа с локацией

В прошлой версии игры локация была разделена на 2 части: стартовая часть (джунглевая) и подземная. Обе эти части были далеко не одинаковы в размере. Так, пещеры в прошлой версии занимали в 3 раза меньше места, чем поверхность. Такой расклад вещей не является удивительным, так как руины должны были быть куда важнее в плане геймплея с самого начала. Однако, несмотря на свои размеры, пещеры всё равно были достаточно пустыми и довольно слабыми с точки зрения геймплея, поэтому было решено увеличить их ещё больше.

Теперь пещеры пополнились двумя новыми комнатами: комнатой каменных монстров и глубинными развалинами.

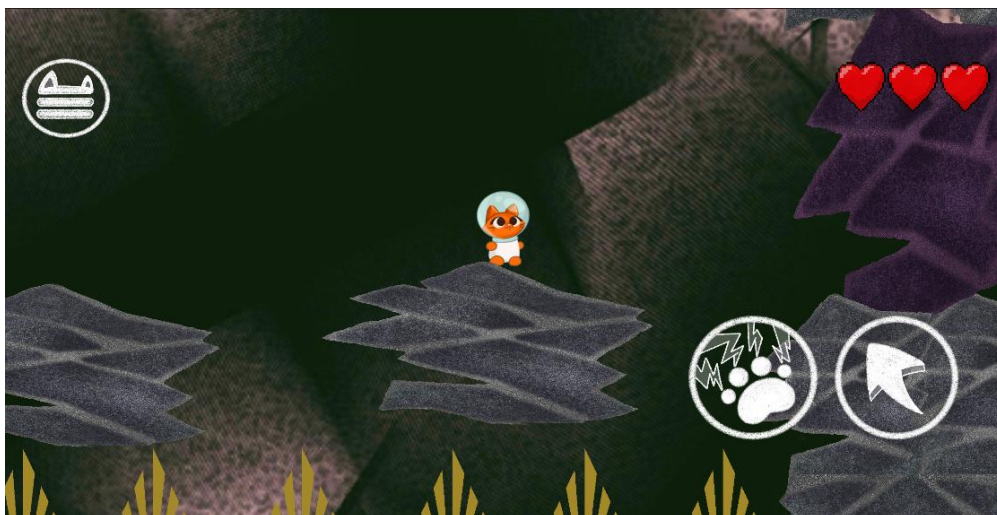


Рисунок 1.1 – Комната каменных монстров

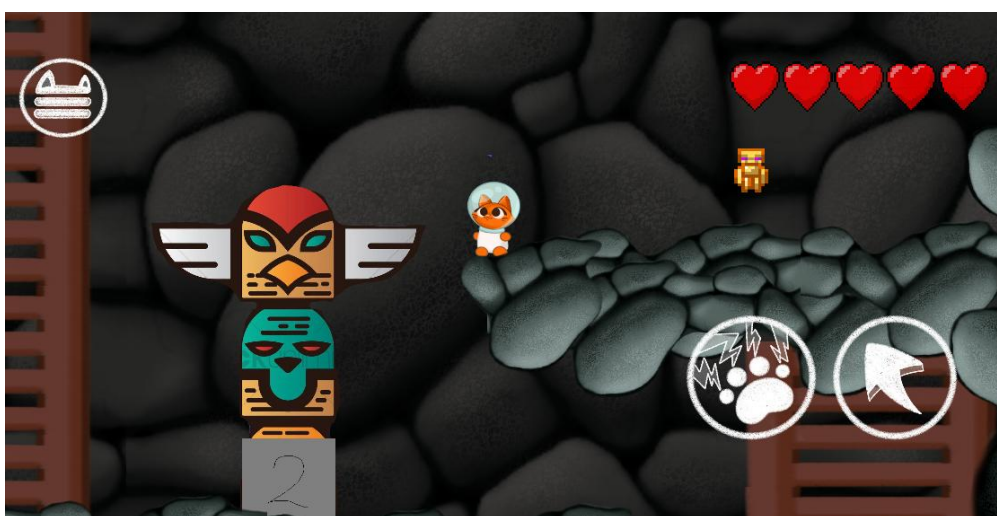


Рисунок 1.2 – Комната глубинных развалин

Самая последняя комната (комната глубинных развалин) является главной целью героя, так как в глубине этого места хранится то, что поможет ему сбежать с этой планеты.

1.2 Финальная головоломка

В комнате глубинных развалин расположена небольшая (и не сказал бы, что сложная) головоломка, при решении которой открывается путь к некой детали, которая может помочь герою восстановить свою ракету.

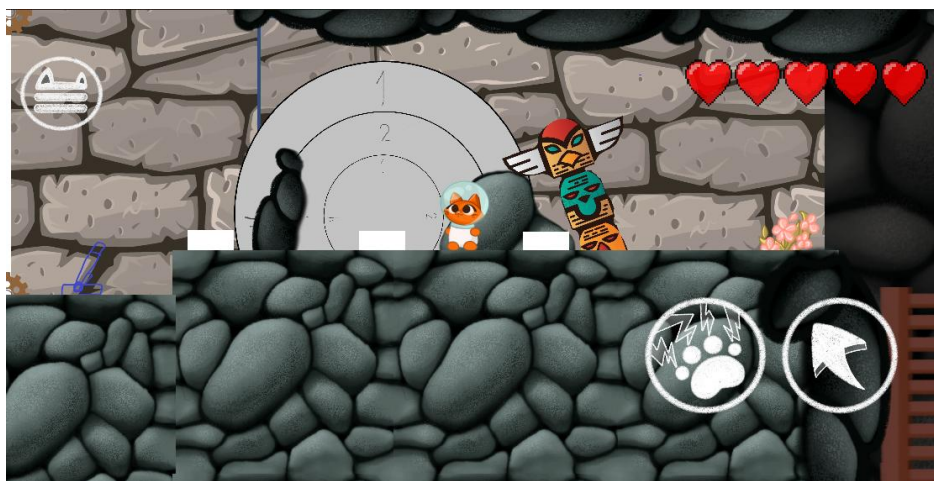


Рисунок 1.3 – Головоломка

Устройства головоломки достаточно простое. Нужно сделать так, чтобы каждая из частей барабана встала в правильную позицию (это что-то наподобие кодового замка). Ориентироваться нужно по тем значениям, которые находятся вертикально.

Кнопки на полу отвечают за вращение барабанов. Достаточно наступить на один, и соответствующий барабан прокрутится на 90 градусов. Как только на барабанах образуется правильный ряд рисунков, можно смело идти и нажимать на рычаг слева от барабана. Если код правильный, стена откроется.



Рисунок 1.4 – Головоломка решена

Несмотря на то, что головоломка кажется безобидной, лучше не стоит пробовать вводить случайные значения и нажимать на рычаг, так как за каждое нажатие рычага при нерешенной головоломке предусмотрено наказание, снимающее 1 сердечко за каждое нажатие.

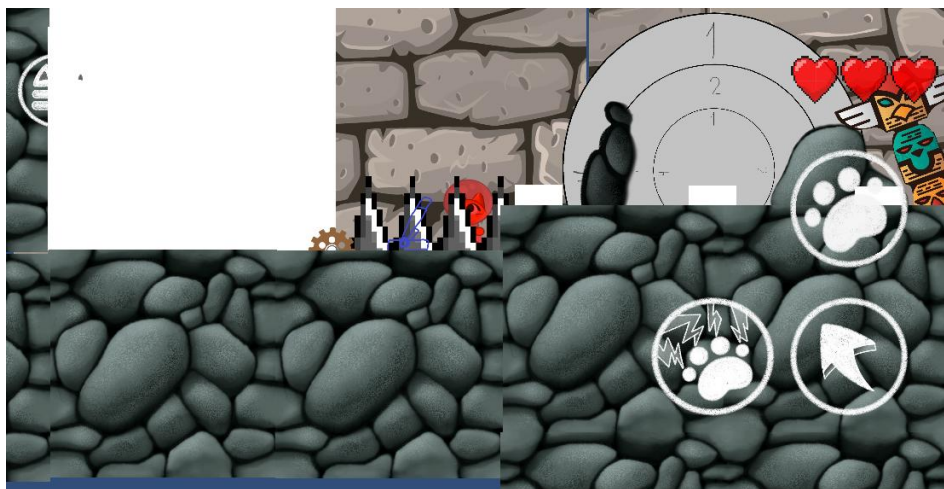


Рисунок 1.5 – Наказание за нажатие рычага

Но как же тогда определить правильный код? По всей комнате разбросано 3 тотемных столба, каждый из которых состоит из трёх сегментов. На каждом из них по одному сегменту занято символами, которые обозначают часть правильного кода. Нужно найти все 3 столба, чтобы открыть комнату с деталью.



Рисунок 1.6 – Тотемный столб с частью решения

1.3 Каменный монстр

С новыми комнатами в игру был добавлен новый противник с новым типом атаки. В отличие от уже присутствующих псов, каменные монстры имеют куда больше единиц здоровья, а также не сражаются самостоятельно. Псы атакуют вблизи врукопашную. В это же время каменные монстры проигрывают анимацию «наращивания» на себе камня и запускают его в героя. Проще говоря, новый тип противников – это противники дальнего боя.



Рисунок 1.7 – Каменный монстр

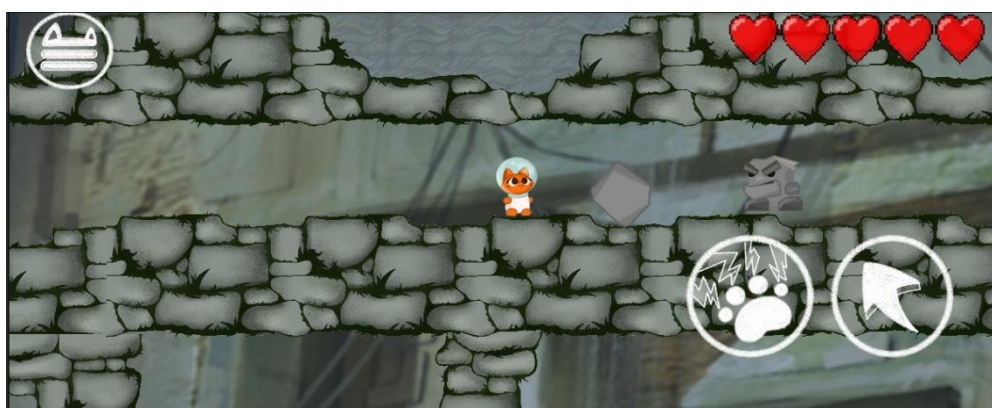


Рисунок 1.8 – Атака каменного монстра

1.4 Дополнения к меню

Было немного обновлено основное меню игры. Теперь при нажатии кнопки «играть» окно меняется. Игроку предлагается выбрать начало игры: начать сначала или выбрать уровень самому. При нажатии кнопки «играть» проигрывается небольшая анимация того, как герой прилетает на экран на своей ракете.

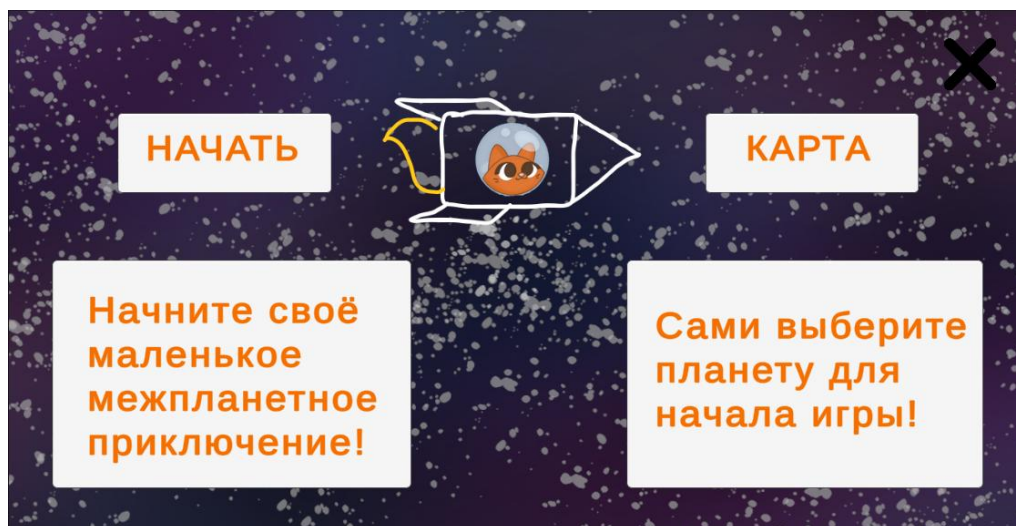


Рисунок 1.9 – Меню выбора начала игры

При выборе начала игры снова проигрывается анимация. На этот раз того, как герой улетает с экрана. Далее появляется экран загрузки, на котором герой летит дальше на своей ракете в качестве индикатора загрузки.

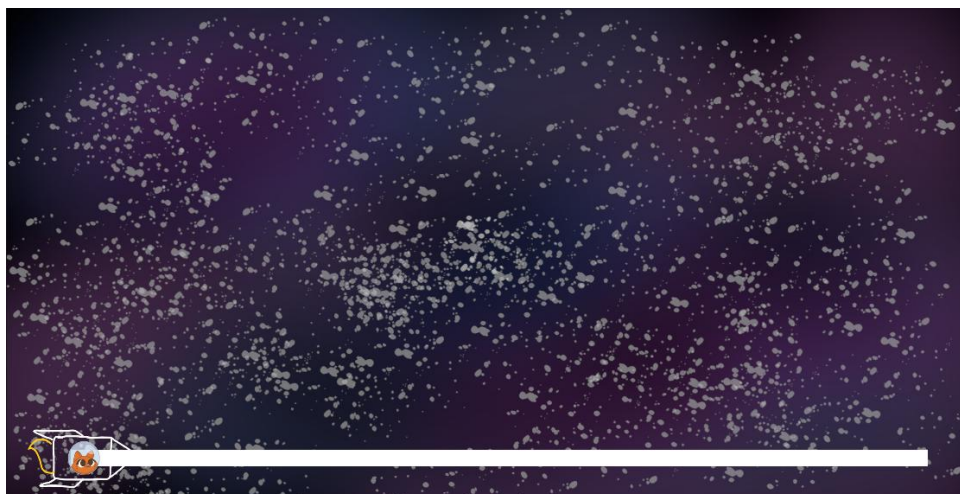


Рисунок 1.10 – Экран загрузки

Индикатор синхронизирован со степенью загрузки игровой сцены, поэтому то, что он показывает, всегда соответствует действительности.

1.5 Планетарная карта

Была добавлена планетарная карта, позволяющая начать игру сразу на любой из заранее открытых планет. Сама по себе карта представляет собой три планеты, медленно перемещающиеся по экрану, при нажатии на которые появляется окно с более подробной информацией, а также функциями закрытия окна или начала игры на выбранной планете.

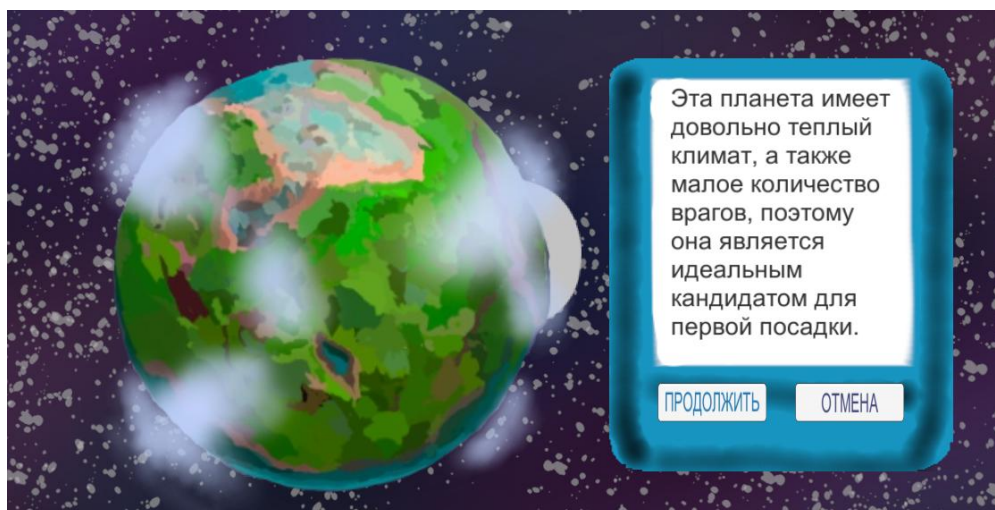


Рисунок 1.11 – Первая планета

Если планета еще не была открыта, то её описание и кнопка начала игры будут отсутствовать.

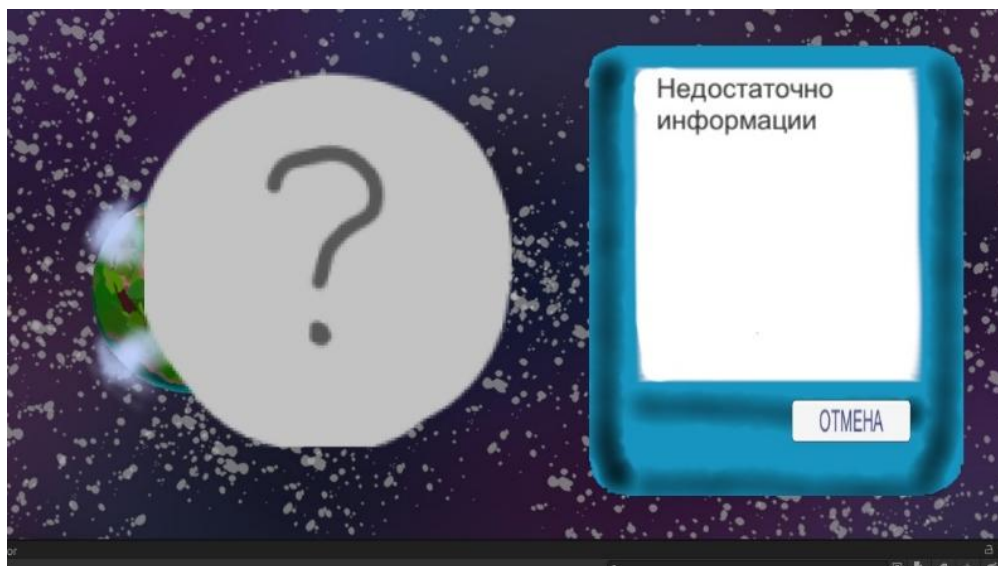


Рисунок 1.12 – Закрытая вторая планета



Рисунок 1.13 – Открытая вторая планета

1.6 Журнал прогресса

За время, прошедшее с предыдущего отчета, на уровень была добавлена кнопка в верхней левой части экрана. При её нажатии появляется меню, в котором можно изменить настройки игры, выйти с уровня или посмотреть журнал прогресса.

Журнал прогресса – это небольшое окно, в котором в зависимости от того, насколько далеко игрок прошел уровень, появляются подсказки по прохождению, если что-то становится слишком непонятным или трудным. Также в этом окне присутствует функция подсчета собранных по уровню компонентов (т.е. показывается прогресс игрока).



Рисунок 1.14 – Меню уровня

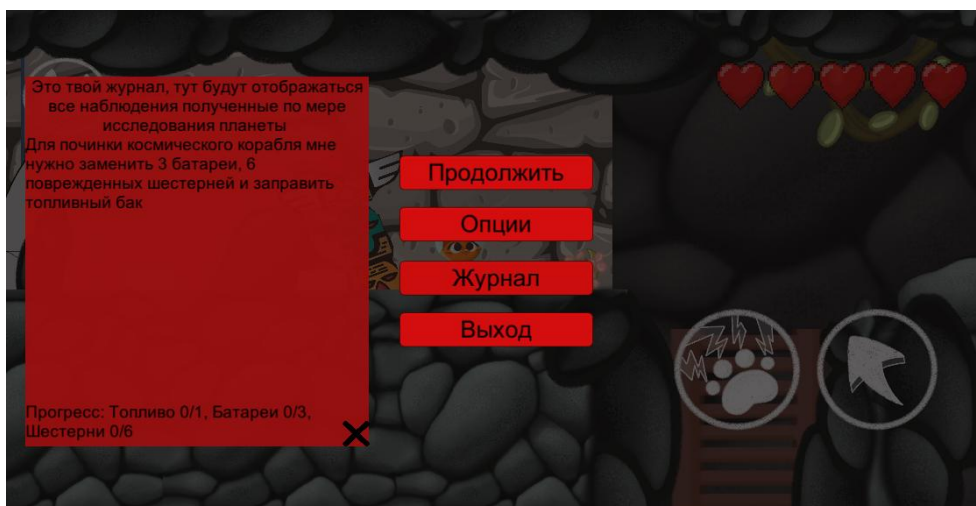


Рисунок 1.15 – Журнал прогресса

1.7 Подсказки для новичков

Самое, пожалуй, небольшое нововведение – это подсказки к элементам управления, которые появляются на некоторое время при первом заходе в игру. Это поможет потенциальным игрокам, которые не играли в платформеры при помощи джойстика или которые вообще не играли в платформеры.

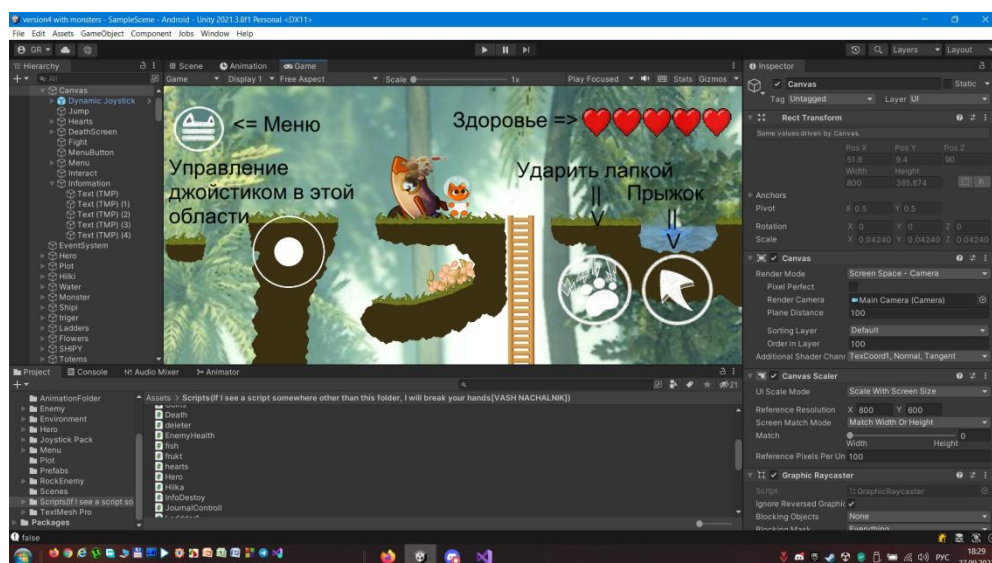


Рисунок 1.16 – Подсказки к элементам геймплея

1.8 Работа художника

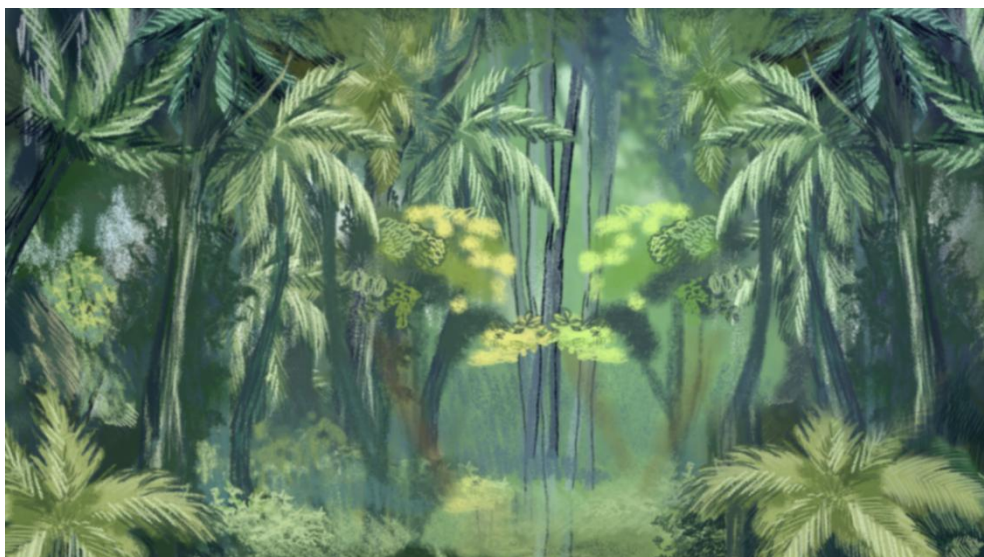


Рисунок 1.17 – Новый фон поверхности



Рисунок 1.18 – Спрайт угря (собираемые объекты)



Рисунок 1.19 – Спрайт шестеренок (собираемые объекты)



Рисунок 1.20 – Спрайт фрукта (собираемые объекты)



Рисунок 1.21 – Подбитая ракета

1.9 Игровой код

Скрипт взаимодействия с планетами на планетарной карте **PlanetAnim.cs**:

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



```

public class PlanetAnim : MonoBehaviour, IPointerDownHandler
{
    [SerializeField] Vector2 scaleMax;
    [SerializeField] Vector2 movePos;
    [SerializeField] float timeToMax = 0.1f;
    [SerializeField] float RandValue, t = 0, startInfo;
    [SerializeField] Vector2 startScale;
    [SerializeField] Vector2 startPos;

    public short NumOfPlanet;
    public GameObject ButtonFuthher;
    public Sprite NewPlanet;

    private SpriteRenderer render;

    private bool IsSlided = false;
    public string description = "";
    public Animator Info;
    public Text InfoBox;

    private void Start()
    {
        startScale = transform.localScale;
        startPos = transform.localPosition;
        scaleMax = new Vector2(startScale.x + 2, startScale.y + 2);
        movePos.x = -4;
        RandValue = Random.Range(1f, 2f);
        render = transform.GetComponent<SpriteRenderer>();
        if (NumOfPlanet == 2 && PlanetsGlobalData.PlanetTwoUnlocked)

```

```

        transform.GetComponent<SpriteRenderer>().sprite = NewPlanet;
    }

    public void OnPointerDown(PointerEventData eventData)
    {
        if (!IsSlided & InfoBox.transform.position.x >= 1200)
        {
            StartCoroutine(c_ScaleUp());
            IsSlided = true;
        }
    }

    public void ExitPressed()
    {
        if (IsSlided)
        {
            StartCoroutine(c_ScaleDown());
            IsSlided = false;
        }
    }

    private void FixedUpdate()
    {
        t += Time.fixedDeltaTime;

        transform.position = new Vector2(transform.position.x, 0.3f*Mathf.Sin(RandValue * t));
    }

    IEnumerator c_ScaleUp()
    {
        if (NumOfPlanet == 1 || (NumOfPlanet == 2 && PlanetsGlobalData.PlanetTwoUnlocked))
        {
            ButtonFuthher.SetActive(true);
            InfoBox.text = description;
        }
    }

```

```

        PlanetsGlobalData.PlanetChosen = NumOfPlanet;
    }
    else
    {
        ButtonFuthher.SetActive(false);
        InfoBox.text = "Недостаточно информации";
    }
    render.sortingOrder = 10;
    Info.SetTrigger("IsTriggered");
    float elapsedTime = 0.0f;
    while ((elapsedTime += Time.fixedDeltaTime) <= timeToMax)
    {
        transform.localScale = Vector2.Lerp(startScale, scaleMax, elapsedTime / timeToMax);
        transform.localPosition = Vector2.Lerp(startPos, movePos, elapsedTime / timeToMax);
        yield return null;
    }
}
IEnumerator c_ScaleDown()
{
    Info.SetTrigger("IsTriggered");
    InfoBox.text = description;
    float elapsedTime = 0.0f;
    while ((elapsedTime += Time.fixedDeltaTime) <= timeToMax)
    {
        transform.localScale = Vector2.Lerp(scaleMax, startScale, elapsedTime / timeToMax);
        transform.localPosition = Vector2.Lerp(movePos, startPos, elapsedTime / timeToMax);
        yield return null;
    }
    render.sortingOrder = 2;
}

```

```
}
```

Скрипт загрузки новой локации, запуска экрана загрузки и управления индикатором загрузки **Blacken.cs**:

```
using System.Collections;
```

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

```
using UnityEngine;
```

```
using UnityEngine.SceneManagement;
```

```
using UnityEngine.UI;
```

```
public class Blacken : MonoBehaviour
```

```
{
```

```
    private Animator NoScreen;
```

```
    public Slider Catslider;
```

```
    public GameObject Screen;
```

```
    void Start()
```

```
    {
```

```
        NoScreen = GetComponent<Animator>();
```

```
    }
```

```
    public void TurnToBlack()
```

```
    {
```

```
        NoScreen.SetTrigger("Blacken");
```

```
    }
```

```
    public void BlackFinished()
```

```
    {
```

```
        int ChosenPlanet = PlanetsGlobalData.PlanetChosen;
```

```
        SceneManager.LoadScene(ChosenPlanet);
```

```
        StartCoroutine>LoadingScreen(ChosenPlanet));
```

```
    }
```



```
IEnumerator LoadingScreen(int ChosenPlanet)
{
    AsyncOperation operation = SceneManager.LoadSceneAsync(ChosenPlanet + 1);
    Screen.SetActive(true);
    while (!operation.isDone)
    {
        float LoadProgress = Mathf.Clamp01(operation.progress / .9f);
        Catslider.value = LoadProgress;
        yield return null;
    }
}
}
```

Скрипт, отвечающий за поведение финальной головоломки (генерации и проверки правильности кода, вращения сегментов барабана при взаимодействии с напольными кнопками и т.д.) **ButtonInteraction.cs**:

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

public class ButtonInteraction : MonoBehaviour
{
    public GameObject TargetObj, Hint;

    public int Solution, i;
    private int[] mass = { 0, 1, 2, 3 };

    private bool Spinning = false;
    public bool PartIsSolved = false;

    Sprite[] NumHint;
```

Coroutine doit;

```
private void Start()
```

```
{
```

```
    NumHint = transform.GetComponentInParent<SpritesMesh>().HintSprite;
```

```
    Solution = Random.Range(0, 3);
```

```
    Hint.GetComponent<SpriteRenderer>().sprite = NumHint[Solution];
```

```
    i = Random.Range(0, 3);
```

```
    TargetObj.transform.rotation = Quaternion.Euler(0, 0, i * 90);
```

```
    if (mass[i] == Solution)
```

```
        PartIsSolved = true;
```

```
}
```

```
private void Update()
```

```
{
```

```
}
```

```
public void OnTriggerEnter2D(Collider2D collision)
```

```
{
```

```
    if (collision.tag == "Player")
```

```
    {
```

```
        transform.position = new Vector2(transform.position.x, transform.position.y - 0.5f);
```

```
        if (!Spinning)
```

```
            StartCoroutine(CrutiteBaraban(90, 10));
```

```
    }
```

```
}
```

```
public void OnTriggerExit2D(Collider2D collision)
```

```
{
```

```
    if (collision.tag == "Player")
```

```
    {
```

```
        transform.position = new Vector2(transform.position.x, transform.position.y + 0.5f);
```

```
}
```

```

}

IEnumerator CrutiteBaraban(float angle, float time)
{
    var it = TargetObj.transform;
    var rot = it.rotation * Quaternion.Euler(0.0f, 0.0f, angle);
    Spinning = true;

    while (true)
    {
        it.rotation = Quaternion.Lerp(it.rotation, rot, time * Time.deltaTime);

        if (Quaternion.Angle(it.rotation, rot) < 0.5f)
        {
            doit = null;
            it.rotation = rot;
            Spinning = false;
            i = (i + 1) % 4;
            if (mass[i] == Solution)
                PartIsSolved = true;
            else
                PartIsSolved = false;
            yield break;
        }
        yield return null;
    }
}
}

```

ЗАКЛЮЧЕНИЕ

На данном этапе проекта должны были быть выполнены следующие работы:

- 1) Закончить создание первой (и единственной) локации игры;
- 2) Расширить возможности геймплея;
- 3) Добавить головоломку в конец локации;
- 4) Произвести косметические доработки уровня;
- 5) Исправить найденные баги.

Первая локация игры была дополнительно увеличена для удобства размещения на ней новых монстров и других объектов. Теперь данная локация закончена (у игрока есть цель на уровне и возможности к этой цели добраться).

Геймплей был доработан. Добавлены журнал прогресса и подсказки для новичков, расширены возможности меню. Добавлена возможность выбора локации.

В последней комнате подземелья была добавлена и проверена головоломка, за решение которой игрока вознаграждают необходимым для продвижения по уровню предметом.

Было нарисовано и размещено много новых спрайтов. Одни из них необходимы (спрайт для каменного монстра и его атаки, собираемые объекты), а другие сугубо декоративные, необходимые лишь для улучшения визуальной составляющей игры.

Было исправлено много багов. Каменные монстры раньше не могли атаковать, если их было много; прикосновение героя к шипам приводило к его исчезновению и невозможности дальнейшей игры; головоломка была невидимой; меню на телефонах имело некорректный вид; боевая анимация часто давала сбои и не проигрывалась и т.д. Всё это было исправлено и доработано к периоду сдачи данного отчета.

Игра сейчас находится на стадии завершения. Осталось заменить плейсхолдерные спрайты на игровые, добавить последние дополнения к геймплею, а также исправить последние баги игры и несовершенство локации. Несмотря на всё это, план на данный период работы был выполнен.