Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Саратовский государственный технический университет

имени Гагарина Ю.А.»

Энгельсский технологический институт

Кафедра «Естественные и математические науки»

Дисциплина: Разработка интерактивных приложений

Отчет

Тема: “Разработка игры “Мишка на луне””

|  |  |
| --- | --- |
|  | Выполнил: студент 4 курса  учебной группы ПИНЖ-41  очной формы обучения  Козлов А.А. |

Энгельс 2025

**Создание графики**

Используется Gimp для создания растровых изображений.

Для двухмерного платформера созданы такие текстуры как:

* Земля;
* Небо;
* Панелька;
* Враг;
* Кнопка выход;
* Земля на подставке;
* Кнопка вверх;
* Кол;
* Медведь;
* Меню;
* Монета;
* Кнопка начать кампанию;
* Ракета;
* Кнопка творческий режим.

Все текстуры представлены на рисунке 1.

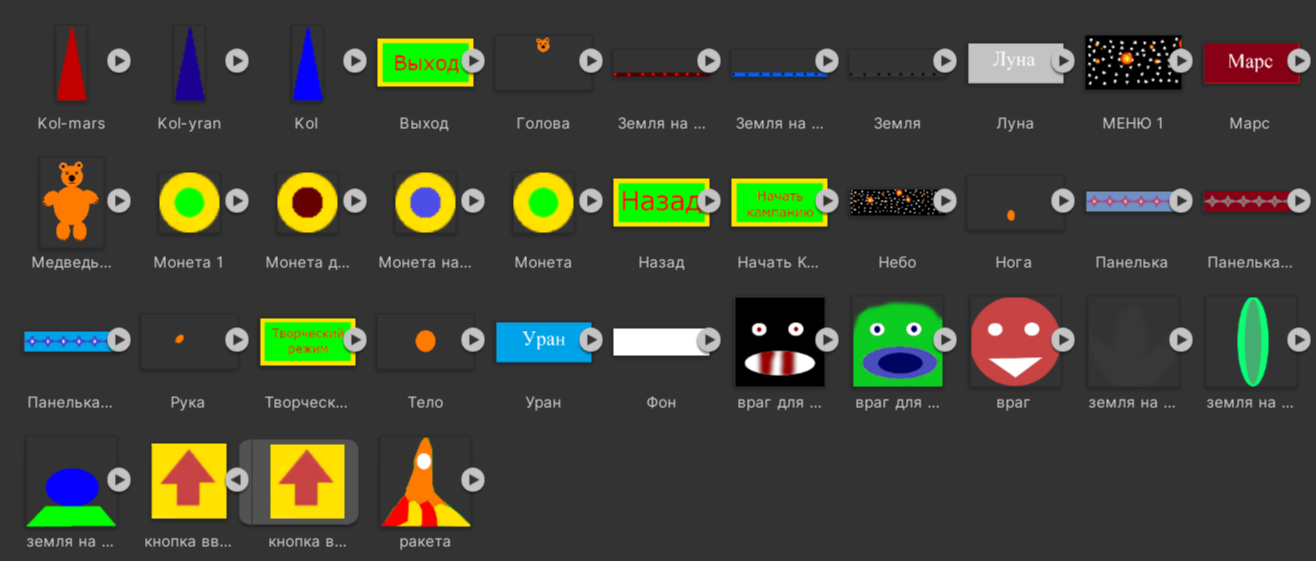


Рисунок 1 - Готовые текстуры

**Создание сцен**

Для двухмерного платформера было создано 7 сцены:

* Меню;
* Кампания;
* Редактор уровня;
* Уровень после редактора;
* Луна;
* Марс;
* Уран.

Сцена “Меню” представлена на рисунке 2.

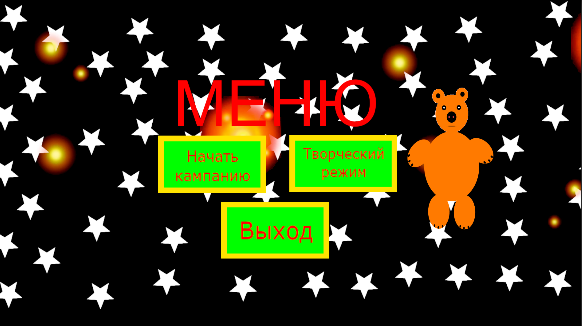


Рисунок 2 - Сцена “Меню”

Сцена “Кампания” представлена на рисунке 3.



Рисунок 3 - Сцена “Кампания”

Сцена “Редактор уровня” представлена на рисунке 4.



Рисунок 4 - Сцена “Редактор уровня”

Сцена “ Уровень после редактора” представлена на рисунке 5.



Рисунок 5 - Сцена “ Уровень после редактора”

Сцена “ Луна” представлена на рисунке 6.



Рисунок 6 - Сцена Луна”

Сцена “Марс” представлена на рисунке 7.



Рисунок 7 - Сцена “Марс”

Сцена “Уран” представлена на рисунке 8.



Рисунок 8 - Сцена “ Уран”

Создание анимации для игрового персонажа

Для игрового персонажа были созданы анимации:

* Взлёт;
* Полёт;
* Приземление;
* Бег.

Все они были записаны с помощью Unity. Их взаимосвязь представлена на рисунке 9.

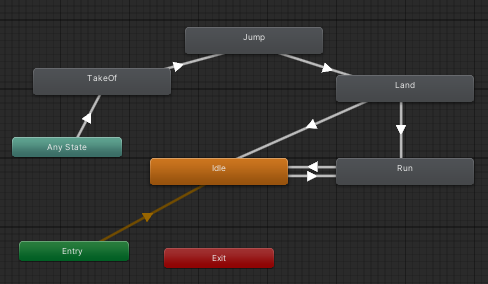


Рисунок 9 - Взаимосвязь анимации

Создание скриптов для взаимодействия игровых объектов

Для таких игровых объектов, как камера, игрок, враг были созданы скрипты. Благодаря им:

* Камера движется за игроком, знает границы локации;
* Враг способен видеть игрока на определенном расстоянии и следовать за ним;
* Игрок способен двигаться, собирать монеты, умереть, завершить уровень;
* Игрок имеет анимацию.

Также был создан скрипт для игрового объекта “Кол”, благодаря которому он появляется на сцене “Уровень после редактирования”.

**Создание пользовательского интерфейса**

Пользовательский интерфейс на сцене “Редактор уровня” (рисунок 10) представляет собой:

* Джойстик;
* Кнопку “Создать”, которая позволяет создать игровой объект “Кол”, передать ему управление, остановить камеру;
* Кнопку “Установить”, которая передает управление от игрового объекта “Кол” назад к камере;
* Кнопку “Готово”, которая позволяет перейти на следующую сцену;
* Кнопку “Выход”, возвращающую на сцену “Меню”.



Рисунок 10 - Пользовательский интерфейс сцены “Редактор уровня”

Пользовательский интерфейс на сценах “Луна”, “Марс”, “Уран”, “Уровень после редактора” (рисунок 11) представлять собой джойстик, кнопку для прыжка, кнопку выход, возвращающую на сцену “Меню”.

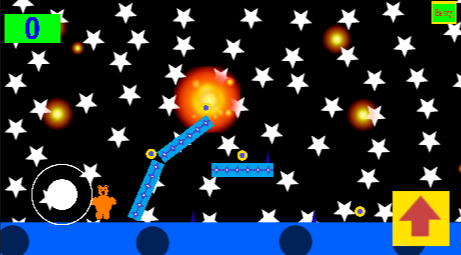


Рисунок 11 - Пользовательский интерфейс сцен “Кампания” и “Уровень после редактора”

Пользовательский интерфейс на сцене “Меню” (рисунок 12) представляет собой:

* Кнопку “Начать кампанию”, которая позволяет перейти на сцену “Кампания”;
* Кнопку “Творческий режим”, которая позволяет перейти на сцену “Редактор уровня”;
* Кнопку “Выход”, которая позволяет выйти из игры.

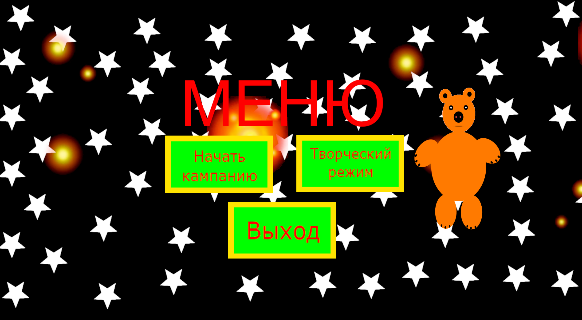


Рисунок 12 - Пользовательский интерфейс сцены “Меню”

Пользовательский интерфейс после смерти (рисунок 13) представляет собой:

* Кнопку “Реванш”, которая позволяет начать уровень заново;
* Кнопку “Выход”, которая позволяет перейти на сцену “Меню”.

****

Рисунок 13 - Пользовательский интерфейс после смерти

Пользовательский интерфейс после завершения игры (рисунок 14) представлен кнопкой выход, которая позволяет перейти на сцену “Меню”.

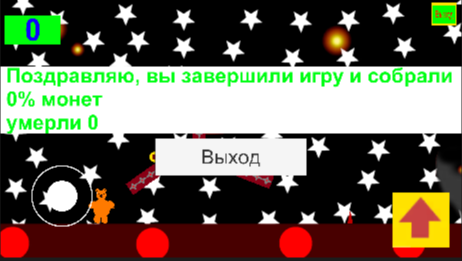
****

Рисунок 14 - Пользовательский интерфейс после завершения игры

**Адаптивность**

Игровой движок Unity автоматически подстраивает под экран смартфона только ширину изображения, находящегося в камере. Поэтому был использован скрипт, который подстраивает изображение под экран по высоте.

С помощью встроенных функций Unity все объекты пользовательского интерфейса привязаны к определённым углам экрана. Благодаря этому на разных устройствах, интерфейс выглядит одинаково. Для имитации разрешения смартфонов, планшетов используется Unity. При адаптации под экран используется метод Shrink. Поэтому чем меньше разрешение, тем меньше видно пользователю. Пример на рисунках 15-16.



Рисунок 15 - Изображение на планшете Apple iPad (5th gen)

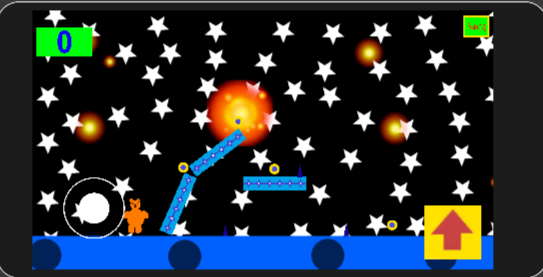


Рисунок 16 - Изображение на смартфоне LG G3