МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Кемеровский государственный университет»

Институт фундаментальных наук

Кафедра ЮНЕСКО по информационным вычислительным технологиям

**ОТЧЁТ**

О выполненной работе за первый период по дисциплине “Учебная практика. Технологическая (проектно-технологическая) практика”.

Проект «Reaper Game»

**Выполняют:**

студенты направления подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем, направленности (профиля) подготовки «Информационные системы и базы данных»

| Перваков Александр Сергеевич |  |  |  |
| --- | --- | --- | --- |
| (ФИО) |  | (оценка) |  |
| Киселев Константин Евгеньевич |  |  |  |
| (ФИО) |  | (оценка) |  |
| Куракин Евгений Русланович |  |  |  |
| (ФИО) |  | (оценка) |  |

Кемерово 2022

**ИЗУЧЕНИЕ UNITY**

Выбор Unity как движка для разработки нашей игры был обоснован тем, что редактор Unity имеет простой Drag&Drop интерфейс, а также установкой плагинов KALI который легко настраивать, состоящий из различных окон, благодаря чему можно производить отладку игры прямо в редакторе.

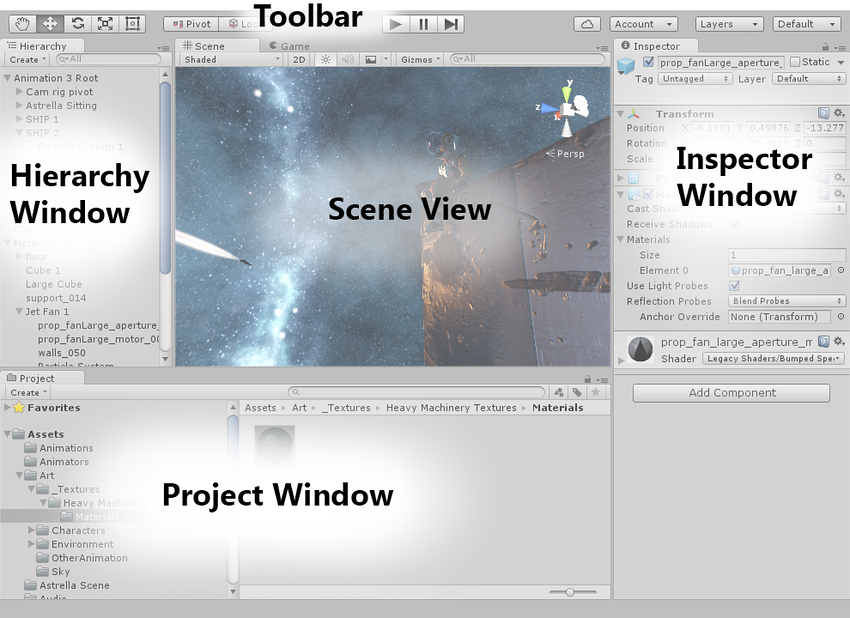
[https://docs.unity3d.com/Manual](https://docs.unity3d.com/Manual/UnityOverview.html) - документация к unity, на основании которой предствлена вся информация ниже.

В Unity есть множество вкладок, с которыми по началу может быть трудно разобраться.

Это такие вкладки, как:

* **Scene** – отвечает за игровую сцену (тут мы можем передвигать объект);
* **Game** – отвечает за конечную картинку, что увидит пользователь во время игры;
* **Hierarchy** – иерархия всех объектов, что есть у вас на сцене;
* **Inspector** – инспектор по каждому из объектов. Позволяет просмотреть характеристики каждого отдельного объекта;
* **Project** – папка, что содержит все материалы вашего проекта;
* **Console** – консоль, куда будут выводиться различные сообщения по ходу игры.

**Главный экран**



**Scene View** - это ваша интерактивная песочница. Вы будете использовать Scene View для выбора и расположения окружений, игрока, камеры, врагов, и всех остальных игровых объектов. Управление и манипулирование объектами с использованием окна Scene View является одной из наиболее важных функций Unity, поэтому важно уметь делать это быстро.

**Hierarchy Window** - содержит все объекты(GameObject) в текущей сцене. Некоторые из них являются прямыми экземплярами файлов ассетов, таких как 3D-модели, а другие - экземпляры префабов, пользовательских объектов, из которых состоит большая часть вашей игры. Вы можете выбрать объекты в иерархии, и перетащить один объект на другой, для создания родительской связи(Parenting) (см. ниже). При добавлении и удалении объектов в сцене, они также будут появляться и исчезать из Иерархии.

**Project Window** - В этом окне вы можете работать с ассетами, которые находятся в вашем проекте.

**Inspector Window** - Панель Inspector используется для того, чтобы просматривать и редактировать свойства объектов различными способами.

**Toolbar** - Панель инструментов, обеспечивает доступ к наиболее важным рабочим функциям.

Во время изучения, мы смотрели видео по Unity, после чего пробовали на практике, рассказанное в ролике. Также перед применением на практике методов или классов unity, изучали документацию, которая понятно изложена на сайте юнити,+

по этим аспектам

**Основные компоненты**

**Assets** - это представление любого предмета, который может быть использован в вашей игре или проекте. Ресурс может быть получен из файла, созданного вне Unity, такого как 3D-модель, аудиофайл, изображение или любой из других типов файлов, поддерживаемых Unity.

**Scenes** - Сцены содержат объекты вашей игры. Они могут использоваться для создания главного меню, отдельных уровней и для других целей. Можно считать каждый файл сцены отдельным игровым уровнем. В каждой сцене можно разместить объекты окружения, заграждения, декорации, по кусочкам создавая дизайн и саму игру.

**GameObject***.* Каждый объект в игре — это GameObject. Однако, GameObject’ы ничего не делают сами по себе. Они требуют специальной настройки, прежде чем стать персонажами, предметами окружения или специальными эффектами. Если каждый объект — это GameObject, то как мы сможем различить интерактивные объекты и статичную комнату? Что отличает GameObject’ы друг от друга?

GameObject’ы являются контейнерами. Пустая коробка, которая может содержать внутри различные элементы, такие как остров с запечёнными тенями или физически корректный автомобиль. Чтобы действительно понять GameObject’ы, нужно понять их составляющие, который называются компонентами (**Components**). В зависимости от того, какой вы хотите создать объект, вы будете добавлять различные комбинации компонентов к GameObject’у. Представляйте GameObject как пустой кулинарный горшок, а компоненты как различные ингредиенты, которые составят ваш рецепт игрового процесса. Вы можете также создать ваши собственные компоненты, используя скрипты.

**Scripts** – пользовательские компоненты.

**Prefabs** - это особый тип ассетов, позволяющий хранить весь GameObject со всеми компонентами и значениями свойств. Префаб выступает в роли шаблона для создания экземпляров хранимого объекта в сцене. Любые изменения в префабе немедленно отражаются и на всех его экземплярах, при этом вы можете переопределять компоненты и настройки для каждого экземпляра в отдельности.

**Sprites** - это 2D-графические объекты. Если вы привыкли работать в 3D, спрайты - это, по сути, просто стандартные текстуры, но существуют специальные методы для объединения и управления текстурами спрайтов для повышения эффективности и удобства во время разработки.

После ознакомления с основами с помощью документации юнити, мы посмотрели ролик - <https://www.youtube.com/watch?v=nRGOW9O7ARk>, в котором рассказывалось о взаимодействии с юнити, некоторых полезных приемах и началом разработки.

**Изучение языка программирования C#**

В Unity используется производительный язык программирования C#. Благодаря C# и кроссплатформенной реализации .NET, – Unity-проекты могут собираться и работать одинаково стабильно на разных операционных системах.

Для того, чтобы создать игру в Unity, необходимо знать основы языка C#, его синтаксис.

Для начала мы изучили **синтаксис** языка. Узнали и запомнили, как создаются программы, переменные, типы данных. Нами были изучены общие правила написания кода на данном языке программирования, ключевые особенности.

* базовые конструкции:
  + условные конструкции
  + циклы
  + функции
* объекты, типы и ссылки;
* объектно-ориентированное программирование;

Наше ознакомление проходило, так, что мы уже знаем основы и все описанное выше, мы, по сути, изучали не полностью, а только различия в синтаксисе и работе. Главных, на уровне основ, различия языка C# и уже знакомого нам C++ два:

Автоматический сбор мусора, то есть решена проблема утечки памяти и необходимости освобождения памяти при работе с динамическими типами.

В языке C# указатели редко используются, код, применяющий указатели, еще называют небезопасным кодом, вместо него используются ссылки.

Видео по основам C# для юнити, с которыми ознакомилась наша команда:

1. Переменные и основные функции - <https://www.youtube.com/watch?v=blBc45DBFJE&list=PL0lO_mIqDDFVNOKquWCHh4n-Ird5HRB_1&index=2>
2. Обьекты, компоненты, условные операции и циклы - <https://www.youtube.com/watch?v=AMPZ5zVjxyo&list=PL0lO_mIqDDFVNOKquWCHh4n-Ird5HRB_1&index=3>
3. Отслеживание нажатий пользователя - https://www.youtube.com/watch?v=HuDDJfQK0m8&list=PL0lO\_mIqDDFVNOKquWCHh4n-Ird5HRB\_1&index=4
4. Создание обьектов - <https://www.youtube.com/watch?v=2CeedUmODOI&list=PL0lO_mIqDDFVNOKquWCHh4n-Ird5HRB_1&index=5>
5. Корутины - <https://www.youtube.com/watch?v=u-7zWOJmHpg&list=PL0lO_mIqDDFVNOKquWCHh4n-Ird5HRB_1&index=6>
6. OnCollision и тригерры - <https://www.youtube.com/watch?v=Xamh-TaeXiY&list=PL0lO_mIqDDFVNOKquWCHh4n-Ird5HRB_1&index=7>

**Разработка игры**

После знакомства с unity и основами C#, мы смотрели ролики по разработке 2D игры на юнити, одновременно пробуя некоторые вещи на практике, обращаясь к документации unity, так как там подробно и понятно описываются все нужные классы и их методы, типы данных и все, что используется при разработке.

<https://www.youtube.com/playlist?list=PLRHtm1zQx-f_iPJNssrmZFSj9H5a00e5I>

https://www.youtube.com/playlist?list=PL3V36b1NObb-s4rJxSnitYV\_SINVj2O\_9

**Источники**

Основные источники информации в том порядке, котором наша команда изучала их(за исключением документации, к которой обращались на протяжении всего времени):

<https://www.youtube.com/watch?v=nRGOW9O7ARk> - основы работы с unity и начало разработки.

[https://docs.unity3d.com/Manual](https://docs.unity3d.com/Manual/UnityOverview.html) - документация к unity.

<https://www.youtube.com/playlist?list=PL0lO_mIqDDFVNOKquWCHh4n-Ird5HRB_1> - плейлист с видео по C# для unity.

<https://www.youtube.com/playlist?list=PLRHtm1zQx-f_iPJNssrmZFSj9H5a00e5I> - плейлист с видео по созданию 2D игры на unity.

<https://www.youtube.com/playlist?list=PL3V36b1NObb-s4rJxSnitYV_SINVj2O_9> - плейлист с видео по созданию платформера на unity.

<https://www.youtube.com/watch?v=Mwvkt4UqW-Q> - видео по созданию главного меню для игры

**ЗАКЛЮЧЕНИЕ**

На основе изученной нами информации, уже получится создать игру, но по мере продвижения, придется неоднократно обращаться к различным источникам информации, за получением новой.