

*Приложение 2.33.1 к ОПОП по специальности
09.02.11 Разработка и управление программным обеспечением*

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет им. Г.И. Носова»

Многопрофильный колледж

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ДЛЯ ПРАКТИЧЕСКИХ ЗАНЯТИЙ
УЧЕБНОЙ ДИСЦИПЛИНЫ**

ОП.11 Разработка компьютерных игр

**для обучающихся специальности
09.02.11 Разработка и управление программным обеспечением**

СОДЕРЖАНИЕ

1 Введение	3
2 Методические указания	4
Практическое занятие 1 Установка и настройка Unity	
Практическое занятие 2 Создание игрока Space Shooter	5
Практическое занятие 3 Создание фона, эффект движения	6
Практическое занятие 4 Создание врагов	7
Практическое занятие 5 Создание волн. Генерация уровней	8
Практическое занятие 6 Создание бонусов	9
Практическое занятие 7 Создаем босса игры	10
Практическое занятие 8 Информационные панели	11
Практическое занятие 9 Анимация и системы частиц	12
Практическое занятие 10 Интерфейс: главное меню, магазин, панель апгрейда	13
Практическое занятие 11 Database	14
Практическое занятие 12 Логика интерфейса	15
Практическое занятие 13 Билд игрового продукта	16

1 ВВЕДЕНИЕ

Важную часть теоретической и профессиональной практической подготовки обучающихся составляют практические занятия.

Состав и содержание практических занятий направлены на реализацию Федерального государственного образовательного стандарта среднего профессионального образования.

Ведущей дидактической целью практических занятий является формирование профессиональных практических умений (умений выполнять определенные действия, операции, необходимые в последующем в профессиональной деятельности) или учебных практических умений (умений выполнять определенные действия, операции, необходимые в последующем в профессиональной деятельности), необходимых в последующей учебной деятельности.

В соответствии с рабочей программой учебной дисциплины «Разработка компьютерных игр» предусмотрено проведение практических занятий.

В результате их выполнения, обучающийся должен уметь:

Уд 1. Определять структуру и интерфейсы модулей;

Уд 2. выбирать подходящие языки программирования и технологии для реализации модуля;

Уд 3. проектировать интерфейсы программного обеспечения для взаимодействия с другими модулями и системами;

Уд 4. разрабатывать модули программного обеспечения с использованием различных языков программирования и технологий;

Уд 5. создавать интерфейсы для взаимодействия с другими модулями и системами;

Уд 6. работать с системой контроля версий.

Содержание практических занятий ориентировано на подготовку обучающихся к освоению профессионального модуля программы подготовки специалистов среднего звена по специальности и овладению **профессиональными компетенциями:**

ПК 2.1. Проектировать модули программного обеспечения;

ПК 2.2. Разрабатывать модули программного обеспечения.

А также формированию общих компетенций:

ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности.

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.

Практические занятия проводятся в рамках соответствующей темы, после освоения дидактических единиц, которые обеспечивают наличие знаний, необходимых для ее выполнения.

2 МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Тема 1.1 Знакомство со средой разработки Unity

Практическое занятие №1 Установка и настройка Unity

Цель: создать и настроить первый проект в Unity

Выполнив работу, вы будете уметь:

Уд 1. Определять структуру и интерфейсы модулей;

Уд 2. выбирать подходящие языки программирования и технологии для реализации модуля.

Материальное обеспечение:

Unity Hub, Unity Editor 2022+, LibreOffice, 7zip

Задание:

1. Настроить среду разработки Unity;
2. Создать папку со своей фамилией, в папке группы, создать новый проект и сохранить в свою папку.

Порядок выполнения работы:

- 1 Выполнить настройку среды Unity для режима 2D;
- 2 Создать проект в папку группы;
- 3 Добавить иерархию папок в проекте;
- 4 Оформить отчёт.

Форма представления результата:

Отчет о проделанной работе.

Критерии оценки:

«5» (отлично): выставляется студенту, если практическая работа в полном объеме, решение оформлено с соблюдением установленных правил; студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач.

«4» (хорошо): выставляется студенту, если при выполнении задания допущены незначительные ошибки, решение оформлено с соблюдением установленных правил; студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач;

«3» (удовлетворительно): выставляется студенту, если задание выполнено с «грубыми» ошибками, решение оформлено без соблюдения установленных правил;

«2» (неудовлетворительно): выставляется студенту, если работа не выполнена.

Тема 2.1 Разработка компьютерной игры

Практическое занятие №2 Создание игрока Space Shooter

Цель: создать и настроить движение игрока.

Выполнив работу, вы будете уметь:

Уд 1. Определять структуру и интерфейсы модулей;

Уд 2. выбирать подходящие языки программирования и технологии для реализации модуля;

Уд 4. разрабатывать модули программного обеспечения с использованием различных языков программирования и технологий;

Уд 6. работать с системой контроля версий.

Материальное обеспечение:

Unity Hub, Unity Editor 2022+, LibreOffice, 7zip

Задание:

1. Создать объект, контролируемый игроком;
2. Создать управление, для этого объекта.

Порядок выполнения работы:

- 1 Создать объект для управления игроком;
- 2 Добавить BoxCollider;
- 3 Добавить Rigidbody;
- 4 Написать скрипт Player_Moving;
- 5 Оформить отчёт.

Форма представления результата:

Отчет о проделанной работе.

Критерии оценки:

«5» (отлично): выставляется студенту, если практическая работа в полном объеме, решение оформлено с соблюдением установленных правил; студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач.

«4» (хорошо): выставляется студенту, если при выполнении задания допущены незначительные ошибки, решение оформлено с соблюдением установленных правил; студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач;

«3» (удовлетворительно): выставляется студенту, если задание выполнено с «грубыми» ошибками, решение оформлено без соблюдения установленных правил;

«2» (неудовлетворительно): выставляется студенту, если работа не выполнена.

Практическое занятие №3

Создание фона, эффект движения

Цель: научиться создавать динамические сцены в 2D проекте

Выполнив работу, вы будете уметь:

Уд 1. Определять структуру и интерфейсы модулей;

Уд 3. проектировать интерфейсы программного обеспечения для взаимодействия с другими модулями и системами;

Уд 5. создавать интерфейсы для взаимодействия с другими модулями и системами.

Материальное обеспечение:

Unity Hub, Unity Editor 2022+, LibreOffice, 7zip

Задание:

1. Создать пустые объекты для разделения слоев сцены игры;
2. Добавить на сцену фон;
3. Разделить слои на сцене;
4. Добавить звезды и планету на сцену.

Порядок выполнения работы:

- 1 Создать пустые объекты на сцене;
- 2 Добавить текстуру фона на сцену;
- 3 Распределить слои на сцене по пустым объектам;
- 4 Создать слои платформ на сцене;
- 5 Оформить отчет.

Форма представления результата:

Отчет о проделанной работе.

Критерии оценки:

«5» (отлично): выставляется студенту, если практическая работа в полном объеме, решение оформлено с соблюдением установленных правил; студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач.

«4» (хорошо): выставляется студенту, если при выполнении задания допущены незначительные ошибки, решение оформлено с соблюдением установленных правил; студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач;

«3» (удовлетворительно): выставляется студенту, если задание выполнено с «грубыми» ошибками, решение оформлено без соблюдения установленных правил;

«2» (неудовлетворительно): выставляется студенту, если работа не выполнена.

Практическое занятие №4

Создание врагов

Цель: создать вражеский объект с механикой передвижения и стрельбы

Выполнив работу, вы будете уметь:

Уд 2. выбирать подходящие языки программирования и технологии для реализации модуля;

Уд 3. проектировать интерфейсы программного обеспечения для взаимодействия с другими модулями и системами;

Уд 4. разрабатывать модули программного обеспечения с использованием различных языков программирования и технологий.

Материальное обеспечение:

Unity Hub, Unity Editor 2022+, LibreOffice, 7zip

Задание:

1. Создать объект для будущего врага;
2. Добавить объект врага на сцену;
3. Создать управление этого объекта;
4. Создать метод стрельбы для этого объекта.

Порядок выполнения работы:

- 1 Создать объект для управления врагом;
- 2 Добавить BoxCollider;
- 3 Добавить Rigidbody;
- 4 Написать скрипт ObgMoving;
- 5 Написать скрипт Enemy;
- 6 Оформить отчёт.

Форма представления результата:

Отчет о проделанной работе.

Критерии оценки:

«5» (отлично): выставляется студенту, если практическая работа в полном объеме, решение оформлено с соблюдением установленных правил; студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач.

«4» (хорошо): выставляется студенту, если при выполнении задания допущены незначительные ошибки, решение оформлено с соблюдением установленных правил; студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач;

«3» (удовлетворительно): выставляется студенту, если задание выполнено с «грубыми» ошибками, решение оформлено без соблюдения установленных правил;

«2» (неудовлетворительно): выставляется студенту, если работа не выполнена.

Практическое занятие №5

Создание волн. Генерация уровней

Цель: реализовать генерацию уровней с волнами врагов

Выполнив работу, вы будете уметь:

Уд 2. выбирать подходящие языки программирования и технологии для реализации модуля;

Уд 4. разрабатывать модули программного обеспечения с использованием различных языков программирования и технологий;

Уд 6. работать с системой контроля версий.

Материальное обеспечение:

Unity Hub, Unity Editor 2022+, LibreOffice, 7zip

Задание:

1. Создать префабы различных типов бонусов (здоровье, скорость, урон, щит);
2. Реализовать систему подбора и применения бонусов;
3. Добавить визуальные и звуковые эффекты активации бонусов.

Порядок выполнения работы:

- 1 Создать объект LevelController;
- 2 Создать объект Wave;
- 3 Добавить врагов в объект Wave;
- 4 Написать скрипт LevelController;
- 5 Написать скрипт Wave;
- 6 Оформить отчёт.

Форма представления результата:

Отчет о проделанной работе.

Критерии оценки:

«5» (отлично): выставляется студенту, если практическая работа в полном объеме, решение оформлено с соблюдением установленных правил; студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач.

«4» (хорошо): выставляется студенту, если при выполнении задания допущены незначительные ошибки, решение оформлено с соблюдением установленных правил; студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач;

«3» (удовлетворительно): выставляется студенту, если задание выполнено с «грубыми» ошибками, решение оформлено без соблюдения установленных правил;

«2» (неудовлетворительно): выставляется студенту, если работа не выполнена.

Практическое занятие №6

Создание бонусов

Цель: реализовать систему бонусов и дополнительных возможностей для игрока

Выполнив работу, вы будете уметь:

Уд 1. Определять структуру и интерфейсы модулей;

Уд 3. проектировать интерфейсы программного обеспечения для взаимодействия с другими модулями и системами;

Уд 5. создавать интерфейсы для взаимодействия с другими модулями и системами.

Материальное обеспечение:

Unity Hub, Unity Editor 2022+, LibreOffice, 7zip

Задание:

1. Создать префабы различных типов бонусов (здоровье, скорость, урон, щит);
2. Реализовать систему подбора и применения бонусов.

Порядок выполнения работы:

1. Создать объект BonusManager для управления системой бонусов;
2. Создать базовый скрипт Bonus с виртуальным методом применения эффекта;
3. Разработать префабы бонусов с спрайтами и коллайдерами (IsTrigger);
4. Реализовать систему случайной генерации бонусов на уровне;
5. Оформить отчёт.

Форма представления результата:

Отчет о проделанной работе.

Критерии оценки:

«5» (отлично): выставляется студенту, если практическая работа в полном объеме, решение оформлено с соблюдением установленных правил; студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач.

«4» (хорошо): выставляется студенту, если при выполнении задания допущены незначительные ошибки, решение оформлено с соблюдением установленных правил; студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач;

«3» (удовлетворительно): выставляется студенту, если задание выполнено с «грубыми» ошибками, решение оформлено без соблюдения установленных правил;

«2» (неудовлетворительно): выставляется студенту, если работа не выполнена.

Практическое занятие №7

Создаем босса игры

Цель: реализовать игрового босса с уникальными механиками, фазами боя и искусственным интеллектом

Выполнив работу, вы будете уметь:

Уд 2. выбирать подходящие языки программирования и технологии для реализации модуля;

Уд 3. проектировать интерфейсы программного обеспечения для взаимодействия с другими модулями и системами;

Уд 4. разрабатывать модули программного обеспечения с использованием различных языков программирования и технологий.

Материальное обеспечение:

Unity Hub, Unity Editor 2022+, LibreOffice, 7zip

Задание:

1. Создать префаб босса с уникальными характеристиками;
2. Реализовать искусственный интеллект босса с различными атаками;
3. Добавить систему фаз боя и изменение поведения в зависимости от здоровья.

Порядок выполнения работы:

1. Создать объект Boss с базовыми характеристиками (здоровье, урон, скорость);
2. Разработать скрипт BossAI с состояниями (поиск цели, атака, отступление, специальная атака);
3. Создать систему фаз боя (BossPhaseManager) с изменением поведения при определённых порогах здоровья;
4. Реализовать минимум 3 различных типа атак босса (ближняя, дальняя, специальная);
5. Добавить индикатор здоровья босса на UI с визуальными эффектами при получении урона;
6. Оформить отчёт.

Форма представления результата:

Отчет о проделанной работе.

Критерии оценки:

«5» (отлично): выставляется студенту, если практическая работа в полном объеме, решение оформлено с соблюдением установленных правил; студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач.

«4» (хорошо): выставляется студенту, если при выполнении задания допущены незначительные ошибки, решение оформлено с соблюдением установленных правил; студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач;

«3» (удовлетворительно): выставляется студенту, если задание выполнено с «грубыми» ошибками, решение оформлено без соблюдения установленных правил;

«2» (неудовлетворительно): выставляется студенту, если работа не выполнена.

Практическое занятие №8

Информационные панели

Цель: реализовать пользовательский интерфейс для отображения здоровья и щита игрока

Выполнив работу, вы будете уметь:

Уд 1. Определять структуру и интерфейсы модулей;

Уд 3. проектировать интерфейсы программного обеспечения для взаимодействия с другими модулями и системами;

Уд 5. создавать интерфейсы для взаимодействия с другими модулями и системами.

Материальное обеспечение:

Unity Hub, Unity Editor 2022+, LibreOffice, 7zip

Задание:

1. Создать UI-панель для отображения здоровья (HP) игрока;
2. Реализовать индикатор щита (Shield) с визуальным разделением от здоровья;
3. Настроить динамическое обновление показателей в реальном времени.

Порядок выполнения работы:

1. Создать Canvas для UI-элементов с настройкой Screen Space - Overlay;
2. Создать панель HealthPanel с использованием UI Image или Slider для отображения HP;
3. Создать панель ShieldPanel с отдельным индикатором для щита;
4. Написать скрипт HealthBarController для обновления UI при изменении здоровья;
5. Протестировать корректность отображения при различных сценариях (получение урона, лечение, активация щита);
6. Оформить отчёт.

Форма представления результата:

Отчет о проделанной работе.

Критерии оценки:

«5» (отлично): выставляется студенту, если практическая работа в полном объеме, решение оформлено с соблюдением установленных правил; студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач.

«4» (хорошо): выставляется студенту, если при выполнении задания допущены незначительные ошибки, решение оформлено с соблюдением установленных правил; студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач;

«3» (удовлетворительно): выставляется студенту, если задание выполнено с «грубыми» ошибками, решение оформлено без соблюдения установленных правил;

«2» (неудовлетворительно): выставляется студенту, если работа не выполнена.

Практическое занятие №9

Анимация и системы частиц

Цель: реализовать анимационные состояния игровых объектов и создать визуальные эффекты с использованием систем частиц

Выполнив работу, вы будете уметь:

- Уд 2. выбирать подходящие языки программирования и технологии для реализации модуля;
- Уд 4. разрабатывать модули программного обеспечения с использованием различных языков программирования и технологий;
- Уд 5. создавать интерфейсы для взаимодействия с другими модулями и системами;
- Уд 6. работать с системой контроля версий.

Материальное обеспечение:

Unity Hub, Unity Editor 2022+, LibreOffice, 7zip

Задание:

1. Настроить контроллер анимаций (Animator Controller) для игрового персонажа;
2. Создать базовые анимационные клипы (полёт);
3. Реализовать визуальный эффект (выстрела, двигателя) с помощью модуля Particle System.

Порядок выполнения работы:

1. Добавить компонент Animator к игровому объекту и создать файл Animator Controller;
2. Открыть окно Animation, создать клипы (anim) для состояний Idle (покой) и Run (движение);
3. Создать объект ParticleSystem и настроить модули Emission, Shape, Renderer и Color over Lifetime для желаемого эффекта;
4. Написать скрипт управления анимацией;
5. Реализовать логику активации эффекта частиц по событию через скрипт;
6. Протестировать корректность срабатывания анимаций и эффектов в игровой сцене;
7. Оформить отчёт.

Форма представления результата:

Отчет о проделанной работе.

Критерии оценки:

«5» (отлично): выставляется студенту, если практическая работа в полном объеме, решение оформлено с соблюдением установленных правил; студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач.

«4» (хорошо): выставляется студенту, если при выполнении задания допущены незначительные ошибки, решение оформлено с соблюдением установленных правил; студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач;

«3» (удовлетворительно): выставляется студенту, если задание выполнено с «грубыми» ошибками, решение оформлено без соблюдения установленных правил;

«2» (неудовлетворительно): выставляется студенту, если работа не выполнена.

Практическое занятие №10

Интерфейс: главное меню, магазин, панель апгрейда

Цель: реализовать многоэкранную систему пользовательского интерфейса с навигацией между главным меню, магазином и панелью улучшения характеристик

Выполнив работу, вы будете уметь:

- Уд 1. Определять структуру и интерфейсы модулей;
- Уд 3. проектировать интерфейсы программного обеспечения для взаимодействия с другими модулями и системами;
- Уд 5. создавать интерфейсы для взаимодействия с другими модулями и системами.

Материальное обеспечение:

Unity Hub, Unity Editor 2022+, LibreOffice, 7zip

Задание:

1. Спроектировать и создать UI-панели главного меню, магазина и улучшений;
2. Реализовать систему навигации и переключения между экранами.

Порядок выполнения работы:

1. Создать основной Canvas и настроить параметры масштабирования (Canvas Scaler, Screen Match Mode);
2. Разработать структуру UI для главного меню: фон, заголовок, кнопки «Играть», «Магазин», «Улучшения», «Выход»;
3. Создать панель магазина с сеткой товаров, отображением баланса игрока и кнопками покупки;
4. Спроектировать панель апгрейда с индикаторами характеристик, кнопками улучшения и визуализацией стоимости/уровней;
5. Оформить отчёт.

Форма представления результата:

Отчет о проделанной работе.

Критерии оценки:

«5» (отлично): выставляется студенту, если практическая работа в полном объеме, решение оформлено с соблюдением установленных правил; студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач.

«4» (хорошо): выставляется студенту, если при выполнении задания допущены незначительные ошибки, решение оформлено с соблюдением установленных правил; студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач;

«3» (удовлетворительно): выставляется студенту, если задание выполнено с «грубыми» ошибками, решение оформлено без соблюдения установленных правил;

«2» (неудовлетворительно): выставляется студенту, если работа не выполнена.

Практическое занятие №11

Database

Цель: реализовать систему сохранения и загрузки игровых данных (локальная БД) для обеспечения персистентности состояния игры

Выполнив работу, вы будете уметь:

- Уд 1. Определять структуру и интерфейсы модулей;
- Уд 2. выбирать подходящие языки программирования и технологии для реализации модуля;
- Уд 4. разрабатывать модули программного обеспечения с использованием различных языков программирования и технологий;
- Уд 6. работать с системой контроля версий.

Материальное обеспечение:

Unity Hub, Unity Editor 2022+, LibreOffice, 7zip

Задание:

1. Создать структуру данных для хранения состояния игрока (валюта, открытые уровни, настройки);
2. Реализовать класс-менеджер базы данных с методами сериализации и десериализации;
3. Интегрировать систему сохранения в моменты завершения уровня и запуска игры.

Порядок выполнения работы:

1. Создать скрипт PlayerData (или SaveData), содержащий публичные поля для сохраняемых параметров (например, int gold, int unlockedLevel, float highScore);
2. Добавить атрибут [System.Serializable] к классу данных для возможности конвертации в JSON;
3. Создать скрипт DatabaseManager (или SaveSystem);
4. Реализовать метод Save(), использующий JsonUtility.ToJson для преобразования объекта данных в строку и File.WriteAllText для записи в Application.persistentDataPath;
5. Реализовать метод Load(), использующий File.ReadAllText и JsonUtility.FromJson для восстановления состояния;
6. Оформить отчёт.

Форма представления результата:

Отчет о проделанной работе.

Критерии оценки:

«5» (отлично): выставляется студенту, если практическая работа в полном объеме, решение оформлено с соблюдением установленных правил; студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач.

«4» (хорошо): выставляется студенту, если при выполнении задания допущены незначительные ошибки, решение оформлено с соблюдением установленных правил; студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач;

«3» (удовлетворительно): выставляется студенту, если задание выполнено с «грубыми» ошибками, решение оформлено без соблюдения установленных правил;

«2» (неудовлетворительно): выставляется студенту, если работа не выполнена.

Тема 3.1 Перенос игры на различные платформы

Практическое занятие №12

Логика интерфейса

Цель: реализовать программную логику пользовательского интерфейса, обеспечивающую взаимодействие элементов UI с игровой механикой через систему событий и контроллеры

Выполнив работу, вы будете уметь:

- Уд 1. Определять структуру и интерфейсы модулей;
- Уд 2. выбирать подходящие языки программирования и технологии для реализации модуля;
- Уд 4. разрабатывать модули программного обеспечения с использованием различных языков программирования и технологий;
- Уд 6. работать с системой контроля версий.

Материальное обеспечение:

Unity Hub, Unity Editor 2022+, LibreOffice, 7zip

Задание:

1. Разработать архитектуру управления состояниями интерфейса (открытие, закрытие, блокировка панелей);
2. Реализовать обработку событий от интерактивных UI-элементов и привязку их к игровой логике;
3. Настроить двустороннюю синхронизацию данных между игровыми системами и визуальным представлением интерфейса.

Порядок выполнения работы:

1. Создать скрипт UIManager для централизованного управления активностью панелей и очередью отображения;
2. Разработать систему событий на базе C# event или UnityEvent;
3. Написать контроллеры для отдельных экранов (PauseMenuController, SettingsController, HUDController);
4. Привязать UI-компоненты (Button, Toggle, Slider, Dropdown) к публичным методам контроллеров через Inspector или программно через AddListener;
5. Реализовать логику обновления данных: подписать текстовые поля и индикаторы на игровые события, обеспечив автоматическое обновление при изменении состояния;
6. Оформить отчёт.

Форма представления результата:

Отчет о проделанной работе.

Критерии оценки:

«5» (отлично): выставляется студенту, если практическая работа в полном объеме, решение оформлено с соблюдением установленных правил; студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач.

«4» (хорошо): выставляется студенту, если при выполнении задания допущены незначительные ошибки, решение оформлено с соблюдением установленных правил; студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач;

- «3» (удовлетворительно): выставляется студенту, если задание выполнено с «грубыми» ошибками, решение оформлено без соблюдения установленных правил;
- «2» (неудовлетворительно): выставляется студенту, если работа не выполнена.

Практическое занятие №13

Билд игрового продукта

Цель: выполнить сборку проекта в исполняемый файл и подготовить готовый дистрибутив игрового продукта для целевой платформы.

Выполнив работу, вы будете уметь:

Уд 6. работать с системой контроля версий.

Материальное обеспечение:

Unity Hub, Unity Editor 2022+, LibreOffice, 7zip

Задание:

1. Настроить параметры проекта и целевой платформы в Player Settings;
2. Выполнить сборку (билд) игрового проекта в исполняемый файл;
3. Протестировать работоспособность собранного приложения и упаковать дистрибутив.

Порядок выполнения работы:

1. Открыть File → Build Settings, добавить все необходимые сцены в список Scenes In Build;
2. Выбрать целевую платформу;
3. Открыть Player Settings, настроить название продукта, имя компании, иконку приложения и разрешение по умолчанию;
4. В разделе Resolution and Presentation настроить параметры окна (Windowed/Fullscreen, масштабирование);
5. Нажать Build, выбрать целевую директорию и дождаться завершения компиляции;
6. Проверить консоль Unity на отсутствие ошибок, запустить сгенерированный .exe файл;
7. Упаковать папку с билдом в архив (7zip) для передачи или публикации;
8. Оформить отчёт.

Форма представления результата:

Отчет о проделанной работе.

Критерии оценки:

«5» (отлично): выставляется студенту, если практическая работа в полном объеме, решение оформлено с соблюдением установленных правил; студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач.

«4» (хорошо): выставляется студенту, если при выполнении задания допущены незначительные ошибки, решение оформлено с соблюдением установленных правил; студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач;

«3» (удовлетворительно): выставляется студенту, если задание выполнено с «грубыми» ошибками, решение оформлено без соблюдения установленных правил;

«2» (неудовлетворительно): выставляется студенту, если работа не выполнена.