**Слайд 1**

Тема моей работы – Создание игры в С#/Unity. Работу выполнил студент 3 курса 2 группы Суражев Кирилл. Научный руководитель – Кравчук Анжелика Ивановна

**Слайд 2**

Актуальность работы заключается в том, что в наши дни огромное количество самых разных по интересам людей частенько поигрывают в компьютерные игры, причем это не только скучающие школьники или прогульщики-студенты. Среди игроков встречаются и бизнесмены, и политики, и домохозяйки, и инженеры, и художники - в целом абсолютно разные люди. Всех их объединяет одно - желание испытать в виртуальных мирах что-то новое, неизведанное, попытать удачу и получить наслаждение как от игрового процесса, так и от достигнутых в игре результатов.

Актуальность создания состоит в том, что создатели игр могут создавать свои миры, людей, флору и фауну, различных существ и связь всего этого друг с другом. И эти творения способны развлекать, развивать и обладают силой затрагивать человеческие сердца

**Слайд 3 - 4**

Целью данной курсовой работы является написание 2D игры в программе Unity.Название игры: Flappy Bird

Задачи:

* Изучить функции игр
* Изучить классификацию игр по жанрам
* Изучить процесс создания игр
* Реализовать главное меню с названием игры (Flappy Bird) и с кнопками Play (при её нажатии игра начинается) и Quit (при её нажатии происходит выход из игры)
* Реализовать игровой уровень, а именно:
  + *Нижнюю границу уровня, пол (когда птичка касается её, она умирает)*
  + *Механику движения птицы (прыжок, гравитация, движение вправо)*
  + *Механику столкновений объектов друг с другом (трубы и пол не сталкиваются друг с другом, а птичка с трубами или полом – сталкивается)*
  + *Механику генерации новых препятствий и удаления старых, а также механику увеличения сложности по мере продвижения по уровню*
  + *Механику появления новых облаков и удаления старых*
  + *Механику смерти птички (когда она сталкивается с трубой или полом, она умирает, и появляется меню с надписью Game Over, рекордом и текущим результатом, а также кнопки Retry (при нажатии на неё уровень перезапускается) и Main Menu (при нажатии на неё происходит возврат на главное меню)*
  + *Механику звука (при совершении следующих действий: нажатии на кнопку, прыжке, обновлении текущего счёта, смерти должен появляться определённый звук)*
  + *Механику текущего счёта и сохранения рекорда (когда птичка пролетает через отверстие в трубе, счёт увеличивается на 1, и если счёт становится больше, чем текущий рекорд, он записывается в рекорд, причём рекорд хранится в отдельном файле и сохраняется даже после выхода из игры*
  + *Механику ожидания игрока (время как бы остановлено, пока игрок не нажмёт на кнопку прыжка)*

**Слайд 5**

В человеческой практике игровая деятельность выполняет такие функции:

* развлекательную;
* коммуникативную: освоение диалектики общения;
* самореализации в игре как полигоне человеческой практики;
* игротерапевтическую: преодоление различных трудностей, возникающих в других видах жизнедеятельности;
* диагностическую: выявление отклонений от нормативного поведения, самопознание в процессе игры;
* функцию коррекции: внесение позитивных изменений в структуру личностных показателей;
* межнациональной коммуникации: усвоение единых для всех людей социально-культурных ценностей;
* социализации: включение в систему общественных отношений, усвоение норм человеческого общежития

**Слайд 6**

Жанров придумано очень много. Перечислим основные, по которым на сайтах в Интернете сортируются игры.

* Action
* MMORPG
* RPG
* Аркады
* Головоломки
* Гонки
* Приключения
* Симуляторы
* Спортивные
* Стратегии
* Файтинги

**Слайд 7**

Рассмотрим подробнее этапы разработки:

1. Концептирование (Concept)

На этом первом шаге команда придумывает концепцию игры, и проводит начальную проработку игрового дизайна. Главная цель данного этапа – это геймдизайнерская документация, включающая в себя Vision (развернутый документ, описывающий игру, как конечный бизнес-продукт) и Concept Document (начальную проработку всех аспектов игры).

2. Прототипирование (Prototyping)

Важный этап проектирования любой игры – это создание прототипа. Прототип реализуется для оценки основного игрового процесса, проверки различных гипотез, проведения тестов игровых механик, для проверки ключевых технических моментов.

3. Вертикальный срез (Vertical Slice)

Цель Вертикального среза – получить минимально возможную полноценную версию игры, включающую в себя полностью реализованный основной игровой процесс. При этом высокое качество проработки обязательно нужно воплотить только для тех игровых элементов, которые существенно влияют на восприятие продукта. При этом все базовые фичи игры присутствуют как минимум в черновом качестве.

4. Производство контента (Content production)

На этом этапе производится достаточное количество контента для первого запуска на внешнюю аудиторию. Реализуются все фичи, запланированные к закрытому бета-тестированию. Это наиболее продолжительный этап, который может занимать, для крупных клиентских проектов год и более.

5. Friends & Family / CBT (закрытое бета-тестирование)

На этапе CBT продукт впервые демонстрируется достаточно широкой публике, хотя и лояльной продукту или компании. Среди наиболее важных задач на этом этапе выступают: поиск и исправление гейм-дизайнерских ошибок, проблем игровой логики и устранение критических багов.

6. Soft Launch / OBT (открытый бета-тест)

На этом этапе продолжается тестирование игры, но уже на широкой аудитории. Идет оптимизация под большие нагрузки. Игра должна быть готова для приема большого трафика. В игре реализован биллинг и принимаются платежи.

7. Release

Ключевая цель – это получение прибыли. На этом этапе должно быть полностью отлажено оперирование продукта (техническая поддержка, работа с комьюнити), соблюдаются маркетинговые и финансовые планы, ведутся работы по улучшению финансовых показателей, активно отрабатываются каналы по привлечению трафика.

**Слайд 8 -10**

1.Описание системных требований

Игра запустится на любом современном компьютере или Android-е

2. Описание пользовательского интерфейса

В главном меню игры есть два пункта: Play и Quit. Play позволяет начать игру, Quit – выйти из неё. После нажатия на кнопку Start игра сразу же начинается. В игре вы можете нажать Space, чтобы прыгнуть, и Esc, чтобы выйти в главное меню. Также в игре есть интерфейс, показывающий результаты и текущий рекорд (он сохраняется при выходе из игры) (см. рисунок 1.1)

Также в игре есть интерфейсное меню, появляющееся после смерти персонажа (см рисунок 1.2), с вариантами:

1)Retry (перезапуск уровня)

2)Main menu (вернуться на главное меню)

3. Руководство к использованию и эксплуатации

После того как вы установите на компьютер игру, на рабочем столе запустите ярлык с игрой (см рисунок 1.3)

После запуска файла вы увидите окно (см рисунок 1.4)

4. Результаты экспериментальной проверки

При проверке проекта никаких отклонений от поставленной задачи не наблюдалось. В целом игра готова к эксплуатации. В результате экспериментальной проверки никаких ошибок выявлено не было. Игра работает без сбоев и нарушений.

**Слайд 11 – 13**

**2.1 Вступление**

Игра была разработана с помощью туториала CodeMonkey [4].

**2.2 Базовая настройка**

Для того, чтобы создать игровой проект, необходимо: находясь в главном окне Unity, создать новый проект, назвать его Flappy Bird (название игры), из Templates выбрать 2D, выбрать местоположение и нажать Create (см. Рисунок 2.1)

Далее был создан объект GameHandler и к нему был привязан скрипт GameHandler.cs (cм. Приложение А), отвечающий за отладочную загрузку в начале

Далее была выполнены импорт и настройка текстур объектов, использующихся в игре (см. Рисунок 2.2)

Также на GameScene был добавлен спрайт птицы (см. Рисунок 2.3)

Далее был создан скрипт GameAssets.cs (см. Приложение А) для хранения всех объектов для лёгкого доступа, привязанный к новому пустому объекту GameAssets

**2.3 Управление птицей, её физика, анимация**

Был создан и привязан к объекту Bird скрипт управления и физики птицы Bird.cs (см. Приложение 3), отвечающий за перемещение птицы, физику и логику столкновений (если птица сталкивается с полом или препятствием, то появляется интерфейс Game Over, с результатом и меню с вариантами: 1) Restart, 2) Return to the main menu:

Также был разработан скрипт SpriteAnimator.cs (см. Приложение А), отвечающий за анимацию птицы

**2.4 Генерация уровня, сложность**

Был создан объект Level в GameScene и к нему был привязан скрипт Level.cs (см. Приложение А), отвечающий за генерацию новых препятствий, облаков в поле зрения камеры, а также удаление тех, что находятся вне поля зрения камеры

**2.5 Интерфейс в игре**

В редакторе Unity в GameScene были созданы следующие объекты (см. рисунки 2.4, 2.5, 2.6)

На объект ScoreWindow был наложен скрипт ScoreWindow.cs (см. Приложение А), который отвечает за изменение числа очков вверху экрана, а если оно становится выше, чем HIGHSCORE, то HIGHSCORE меняется после смерти персонажа на это число.

Также был создан файл Score.cs (см. Приложение А), который содержит основную логику ScoreWindow.cs

**2.6 Экран загрузки**

Была создана сцена Loading (для того, чтобы было видно, когда игра грузится) (см. Рисунок 2.7)

Далее были разработаны скрипты Loader.cs (см. Приложение А) и LoaderUpdate.cs (см. Приложение А), отвечающие за логику экрана загрузки

Скрипт LoaderUpdate.cs был привязан к объекту LoaderUpdate.cs

**2.7 Главное меню**

Была создана сцена MainMenu (см. Рисунок 2.8)

К объекту MainMenuWindow был привязан скрипт MainMenuWindow.cs (см. Приложение А), отвечающий за логику главного меню (если нажимаешь Play, игра начинается, если Quit, происходит выход из игры)

**2.8 Звук**

Также был создан скрипт SoundManager.cs (см. Приложение А), который отвечает за весь звук в игре

**2.9 Ожидание игрока**

Для большего удобства был добавлен экран ожидания игрока и был разработан скрипт WaitingToStartWindow.cs (см. Приложение А) (время в игре как бы остановлено, пока игрок не нажмёт кнопку прыжка

**Слайд 14**

Итак, заключение:

В настоящей курсовой работе были:

* Изучены функции игр
* Изучена классификация игр по жанрам
* Изучен процесс создания игр
* Реализовано главное меню с названием игры (Flappy Bird) и с кнопками Play (при её нажатии игра начинается) и Quit (при её нажатии происходит выход из игры)
* Реализован игровой уровень, а именно:
  + Нижняя границу уровня, пол
  + Механика движения птицы
  + Механика столкновений объектов друг с другом
  + Механика генерации новых препятствий и удаления старых, а также механику увеличения сложности по мере продвижения по уровню
  + Механика появления новых облаков и удаления старых
  + Механика смерти птички и Main Menu
  + Механика звука
  + Механика текущего счёта и сохранения рекорда
  + Механика ожидания игрока

**Слайд 15**

Спасибо за внимание

Выполнил студент 3 курса 2 группы

Суражев Кирилл

Научный руководитель – Кравчук А.И.