**Частное учреждение образования «Минский колледж предпринимательства»**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | УТВЕРЖДАЮ |
|  |  |  |  | Заведующий отделением ПОИТ |
|  |  |  |  |  |
|  |  |  |  | М.В. Меньшикова |
|  |  |  |  | «\_\_\_» \_\_\_\_\_\_\_\_\_\_20\_\_ г. |

**Игра «Shop Town»**

**Пояснительная записка**

**МКП.3170814-01 81 15**

*technicaldocs.ru*

*Подпись и дата*

*Взам. инв. №*

*Инв. № дубл.*

*Инв. № подл.*

*Подпись и дата*

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  | Учащийся |
|  |  |  |
|  |  | Л.Л. Татур |
|  |  | «\_\_\_» \_\_\_\_\_\_\_\_\_\_20\_\_ г. |
|  |  |  |
|  |  | Руководитель |
|  |  |  |
|  |  | С.Н. Науменко |
|  |  | «\_\_\_» \_\_\_\_\_\_\_\_\_\_20\_\_ г. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

2021

**АННОТАЦИЯ**

Объектом и предметом исследования является телеграм бот, среда

разработки Visual studio 2019.

Цель проекта: создать программу для создания заметок в телеграме.

Название разрабатываемого проекта: телеграм бот “заметки”.

Исходные данные проектирования:

Ключевые слова: visual studio c#,

Дипломный проект: с, 24 рис., 1 табл., 6 источников, 3 приложения.

**МИНСКИЙ КОЛЛЕДЖ ПРЕДПРИНИМАТЕЛЬСТВА**

(название учреждения образования)

УТВЕРЖДАЮ

Председатель цикловой (предметной)   
комиссии «Программное обеспечение информационных технологий»

/ Ю.А. Доманова /

« » 2021 г.

**ЗАДАНИЕ**

**на курсовой проект**

Учащейся Татур Любови Леонидовне

(фамилия, имя, отчество)

курса 3 группы П-1807

по учебной дисциплине «Конструирование программ и языки программирования»

Тема курсового проекта (работы) «Игра «Town Shop»

Исходные данные: данные, получаемые от пользователя; данные, хранящиеся в сериализованных файлах.

**Состав проекта (работы)**

|  |  |
| --- | --- |
| **Пояснительная записка** | |
| Содержание разделов | Срок выполнения |
| Введение. Назначение и область применения | 01.02.21 – 14.02.21 |
| Постановка задачи. Описание организации данных | 15.02.21 – 28.02.21 |
| Описание программных средств. Проектирование интерфейса | 01.03.21 – 14.03.21 |
| Кодирование. Описание логической структуры программы | 01.03.21 – 04.04.21 |
| Тестирование и анализ полученных результатов | 05.04.21 – 11.04.21 |
| Руководство по использованию программного средства | 12.04.21 – 18.04.21 |
| Заключение. Источники, использованные при разработке | 19.04.21 – 25.04.21 |
| Приложения |  |

**Графическая часть проекта**

Лист – 1\_\_\_ER-диаграмма (если есть)

Лист – 2 Диаграмма вариантов использования

Лист – 3\_\_\_Диаграмма классов

Лист – 4 Диаграмма «Название диаграммы» любая на свой вкус

Дата выдачи «01» февраля 2021 г.

Срок сдачи «26 » апреля 2021 г.

**Преподаватель-руководитель** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**курсового проекта**

**Подпись учащегося**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ**

API – Application Programming Interface.

ПП – программный продукт.

ЛКМ – левая кнопка мыши.

ПКМ – правая кнопка мыши.

**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ……………………………………………………………………..

1. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ…………………………….

2. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ……………………………………

2.1 Постановка задачи…………………………………………………………..

2.2 Описание организации данных……………………………………………..

2.3 Описание программных средств……………………………………………

2.4 Проектирование интерфейса……………………………………………….

2.5 Описание логической структуры программы………………………………

3. ТЕСТИРОВАНИЕ И АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ……….

4. РУКОВОДСТВО ПО ИСПОЛЬЗОВАНИЮ ПРОГРАММНОГО

СРЕДСТВА……………………………………………………………………….

ЗАКЛЮЧЕНИЕ…………………………………………………………………..

ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ…………………

ПРИЛОЖЕНИЯ…………………………………………………………………

**СОДЕРЖАНИЕ**

Используй автоматическое оглавление

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 5](#_Toc70466548)

[**1.** **НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ** 6](#_Toc70466549)

[**1.1** **Обоснование необходимости автоматизации** 6](#_Toc70466550)

[**1.2** **Определение данных и их представление** 7](#_Toc70466551)

[**2 ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ** 8](#_Toc70466552)

[**2.1 Постановка задачи** 8](#_Toc70466553)

[**2.2 Описание организации данных** 8](#_Toc70466554)

[**2.3 Описание программных средств** 9](#_Toc70466555)

[**2.4 Проектирование интерфейса** 9](#_Toc70466556)

[**2.5 Описание логической структуры программы** 17](#_Toc70466557)

[**3. ТЕСТИРОВАНИЕ И АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ** 22](#_Toc70466558)

[**4 РУКОВОДСТВО ПО ИСПОЛЬЗОВАНИЮ ПРОГРАММНОГО СРЕДСТВА** 30](#_Toc70466559)

[**Список используемых источников** 31](#_Toc70466560)

**ВВЕДЕНИЕ**

Видеоигра — игра с использованием изображений, сгенерированных электронной аппаратурой. Другими словами, видеоигра является электронной игрой, которая базируется на взаимодействии человека и устройства посредством визуального интерфейса, например, телевизора, монитора компьютера или телефона.

Сейчас известно 7 типов компьютерных игр:

1) Квесты — осуществление одного или нескольких персонажей к конечной цели, осуществляя преодоление препятствий;

2) Экшн — это игры от первого лица. Говоря простым языком: "бродишь и стреляешь";

3) Ролевые игры;

4) Стратегии — в игре такого жанра отсутствует отдельный персонаж. Там человек управляет определенными внутриигровыми процессами. Сначала используя логику, игрок выполняет свою логическую цепочку решения, а после наблюдает последствия своих действий;

5) Симуляторы — достаточно популярный жанр. Суть его в том, что он имитирует, например, управление на автомобиле, самолете, корабле, космическом аппарате. В таких играх нет сюжета, они не развивают какого то привыкания;

6) Логические игры;

7) Азартные игры — идет направленность на мозг человека. Такой вид игры развивает мыслительную деятельность, подводит человека к собственному логическому размышлению, заставляет креативно мыслить, и строить логические цепочки.

Основной целью выполнения курсового проекта является разработка программного средства для занимательного времяпровождения и отдыха, а также расширение, углубление знаний по дисциплине «конструирование программ и языки программирования». Получение навыков проектирования и создания программного обеспечения.

Задачи выполнения курсового проекта:

- систематизация полученных знаний;

- углубление уровня и расширение объема профессионально значимых знаний, умений и навыков;

- формирование умений и навыков самостоятельной работы;

- практическое использование сериализации для хранения информации;

- разработка интуитивно простого интерфейса для пользователя.

**1 НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ**

* 1. **Обоснование необходимости автоматизации**

Microsoft Visual Studio – это интегрированная среда разработки от Microsoft. Он используется для разработки компьютерных программ для Microsoft Windows, а также веб-сайтов, веб-приложений и веб-сервисов.

Visual Studio использует платформы разработки программного обеспечения Microsoft:

– Windows API;

– Windows Forms;

– Windows Presentation Foundation;

– Windows Store;

– Microsoft Silverlight.

Visual Studio включает в себя редактор кода, поддерживающий IntelliSense, а также рефакторинг кода. Интегрированный отладчик работает как отладчик уровня источника и отладчик уровня машины.

Другие встроенные инструменты включают конструктор форм для создания приложений с графическим интерфейсом, веб-дизайнер, дизайнер классов и конструктор схем базы данных. Он принимает плагины, которые расширяют функциональные возможности практически на каждом уровне, включая добавление поддержки для систем контроля версий (таких как Subversion) и добавление новых наборов инструментов, таких как редакторы и визуальные дизайнеры для языков, специфичных для предметной области, или наборов инструментов для других аспектов жизненного цикла разработки программного обеспечения.

Одним из аналогов Visual Studio является Eclipse. Eclipse, отстаёт от Visual Studio. Eclipse имеет ряд минусов:

* при недостаточном выделении памяти для приложения, он будет работать медленно.
* при запуске Eclipse в проекте, хранящемся в сетевой файловой системе, он будет работать медленнее.
* при запуске Eclipse из сетевой установки, она будет работать медленнее.

Unity — межплатформенная среда разработки компьютерных игр, разработанная американской компанией Unity Technologies. Unity позволяет создавать приложения, работающие на более чем 25 различных платформах, включающих персональные компьютеры, игровые консоли, мобильные устройства, интернет-приложения и другие. Выпуск Unity состоялся в 2005 году и с того времени идёт постоянное развитие.

Основными преимуществами Unity являются наличие визуальной среды разработки, межплатформенной поддержки и модульной системы компонентов. К недостаткам относят появление сложностей при работе с многокомпонентными схемами и затруднения при подключении внешних библиотек.

На Unity написаны тысячи игр, приложений, визуализации математических моделей, которые охватывают множество платформ и жанров. При этом Unity используется как крупными разработчиками, так и независимыми студиями.

**1.2 Определение данных и их представление**

Входными данными для разработки ПП являются:

* данные, получаемые от пользователя;
* данные, хранящиеся в сериализованных файлах;

Выходными данными разрабатываемого ПП являются:

* информация, сохраняемая в сериализованных файлах.

Разрабатываемый ПП должен создавать и модифицировать следующие данные и сериализованных файлах:

* количество игровой валюты;
* содержание инвентаря пользователя;
* положение расставленных в игровой комнате предметов.

**2 ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ**

**2.1 Постановка задачи**

Разработать проект, который будет интересен пользователям, в целях занимательного времяпровождения и получения положительных эмоций.

Приложение «Shop Town» предлагает пользователю получить игровую валюту, играя в интерактивную мини-игру, и тратить её на интересующие товары, которые пользователь сможет разместить в своем игровом доме.

На данный момент времени существуют подобные приложения, однако игра «Shop Town» имеет возможность дальнейшего совершенствования и добавления новых различных игр, предметов, иных функций.

Разработанное приложение содержит следующие функции: интерактивная мини-игра для получения игровой валюты, покупка предметов в магазинах, расставление предметов из инвентаря в игровую комнату, возврат предмета из комнаты в инвентарь, продажа предметов из инвентаря.

**2.2 Описание организации данных**

Данные, хранящиеся в сериализованном файле, диаграмма сущность-связь которой представлена в приложении 1:

1)Количество игровой валюты;

2) Информация о предметах, хранящихся в инвентаре пользователя;

3) Информация о предметах, расставленных в игровом доме.

**2.3 Описание программных средств**

Проект был разработан в программе Unity, которая имеет свои достоинства и недостатки.

Как правило, игровой движок предоставляет множество функциональных возможностей, позволяющих их задействовать в различных играх, в которые входят моделирование физических сред, карты нормалей, динамические тени и многое другое. В отличие от многих игровых движков, у Unity имеется два основных преимущества: наличие визуальной среды разработки и межплатформенная поддержка[3]. Первый фактор включает не только инструментарий визуального моделирования, но и интегрированную среду, цепочку сборки, что направлено на повышение производительности разработчиков, в частности, этапов создания прототипов и тестирования. Под межплатформенной поддержкой предоставляется не только места развертывания (установка на персональном компьютере, на мобильном устройстве, консоли и т. д.), но и наличие инструментария разработки (интегрированная среда может использоваться под Windows и Mac OS).

Третьим преимуществом называется модульная система компонентов Unity, с помощью которой происходит конструирование игровых объектов, когда последние представляют собой комбинируемые пакеты функциональных элементов. В отличие от механизмов наследования, объекты в Unity создаются посредством объединения функциональных блоков, а не помещения в узлы дерева наследования. Такой подход облегчает создание прототипов, что актуально при разработке игр.

В качестве недостатков приводятся ограничение визуального редактора при работе с многокомпонентными схемами, когда в сложных сценах визуальная работа затрудняется. Вторым недостатком называется отсутствие поддержки Unity ссылок на внешние библиотеки, работу с которыми программистам приходится настраивать самостоятельно, и это также затрудняет командную работу. Ещё один недостаток связан с использованием шаблонов экземпляров (англ. prefabs). С одной стороны, эта концепция Unity предлагает гибкий подход визуального редактирования объектов, но с другой стороны, редактирование таких шаблонов является сложным. Также, WebGL-версия движка, в силу специфики своей архитектуры (трансляция кода из C# в С++ и далее в JavaScript), имеет ряд нерешённых проблем с производительностью, потреблением памяти и работоспособностью на мобильных устройствах.

Проект был разработан с помощью Visual Studio на языке С#, который имеет свои достоинства и недостатки.

Плюсы Visual Studio:

* бесплатно распространяется Visual Studio Community, с достаточным набором возможностей.
* платные версии могут предоставляться учебным заведениям и студентам бесплатно.
* удобная система умного автодополнения.
* большое количество настроек среды разработки под «себя», благодаря встроенным механизмам и доступным дополнениям.

Минусы Visual Studio:

* версии Visual Studio Professional и Visual Studio Enterprise являются платными.
* тяжеловесная IDE, достаточно требовательная к железу.
* ряд разработчиков считают ее избыточной для создания мелких проектов.
* достаточно маленькая скорость запуска проектов и приложений.
* не работает на Linux.
* платные версии с расширенным функционалом распространяются по подписке: Professional – от 45$ в месяц, Enterprise – от 250$ в месяц.

**2.4 Проектирование интерфейса**

Курсовой проект состоит из 6 сцен:

Сцена 1 – комната пользователя (рис. 2.1);

Сцена 2 – улица с магазинами (рис. 2.2);

Сцена 3 – магазин мебели (рис. 2.3);

Сцена 4 – магазин ванных принадлежностей (рис. 2.4);

Сцена 5 – магазин стикеров (рис. 2.5);

Сцена 6 – центр игровых автоматов (рис. 2.6).

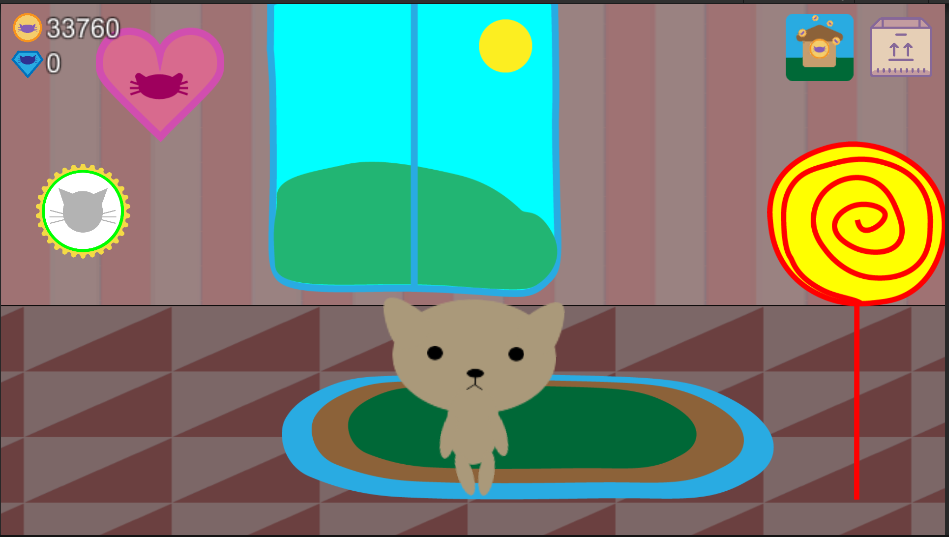


Рис. 2.1. Комната пользователя

Таблица 2.1 – Компоненты сцены «Название формы»:

|  |  |
| --- | --- |
| Название компонента | Назначение компонента |
| guna2TextBox1 | ввод минимальной цены |
| guna2TextBox2 | ввод максимальной цены |
| guna2ComboBox1 | выбор условия для сортировки |
| label2 | сброс сортировки |
| guna2TextBox3 | ввод города отправления |
| guna2TextBox4 | ввод города прибытия |
| guna2DateTimePicker1 | выбор даты отправления |
| guna2DateTimePicker2 | выбор даты обратно |
| guna2NumericUpDown1 | выбор кол-ва мест |

*Продолжение таблицы 2.1 (если есть разрыв)*

|  |  |
| --- | --- |
| guna2ComboBox2 | Выбор класса |
| guna2ChekBox1 | Вывод билетов туда-обартно |

**2.5 Описание логической структуры программы**

**ССЫЛКИ НА ДИАГРАММЫ**

**Для чего класс нужен и его модель**

**Написать таблицу с методы и их назначением**

* 1. **ТЕСТИРОВАНИЕ И АНАЛИЗ ПОЛУЧЕННЫЗ РЕЗУЛЬТАТОВ**

10 тестов

**4 РУКОВОДСТВО ПО ИСПОЛЬЗОВАНИЮ ПРОГРАММНОГО СРЕДСТВА**

При запуске игры персонаж пользователя появляется в пустом игровом доме, пустым инвентарем и имеет некоторое количество игровой валюты.

Пользователь имеет возможность посетить магазины и мини-игру, выйдя на улицу. Чтобы выйти на улицу необходимо нажать на иконку, на которой изображена улица.

Чтобы попасть в определенный магазин, необходимо, находясь на улице, нажать на нужное здание, после чего пользователь попадет в магазин.

Чтобы приобрести вещь в магазине нужно нажать на неё, выбрать необходимое количество и нажать «Да». Если у пользователя хватает валюты на приобретение вещи, они появятся в его инвентаре.

Чтобы открыть инвентарь, необходимо, находясь в игровом доме, нажать на иконку коробки.

Для того, чтобы разместить предметы в игровом доме, необходимо нажать ПКМ на предмет в инвентаре, перетащить предмет на нужное место, и нажать на ЛКМ.

Для того, чтобы вернуть предмет обратно в инвентарь, необходимо, в режиме инвентаря, нажать на предмет в комнате.

Чтобы продать предмет, который находится в инвентаре, необходимо нажать на иконку крестика около нужного предмета. В появившемся окошке выбрать необходимое количество и нажать «Да».

Добавить скринов и ссылаться на них

**ЗАКЛЮЧЕНИЕ**

В ходе курсового проекта была разработана игра «Shop Town». В приложении пользователь имеет возможность заработать игровую валюту, играя в интерактивную мини-игру, чтобы потратить её на предметы из магазина. Игрок может расставить купленные предметы в игровом доме по своему усмотрению.

В ходе курсового проекта был использован язык программирования С# и программы Visual Studio и Unity.

Программное средство было протестировано. Тестирование прошло успешно. В дальнейшем может быть модифицировано следующим образом: добавление новых магазинов и новых предметов, бонусы за ежедневный вход в игру, возможность изменять внешний вид персонажа, добавление новых мини-игр для зарабатывания игровой валюты, возможность существования нескольких комнат в доме.

**ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ**

**Стырить гденить!!**

**ПРИЛОЖЕНИЕ 1 – UML – ДИАГРАММЫ**

Use Case Diagram: Use case View (Система хранения информации об оружии)

Activity diagram (Система хранения информации)