

ОТЧЕТ

по производственной (преддипломной) практике

по специальности 09.02.03 «Программирование в компьютерных системах»

Выполнил студент

группы П1-17

Лихторенко О. С.

\_\_\_\_\_\_\_\_\_\_ (подпись)

Принял преподаватель

Гусятинер Л.Б.

\_\_\_\_\_\_\_\_\_\_(подпись)

\_\_\_\_\_\_\_\_\_\_\_(оценка)

Королев, 2021

Оглавление

[Введение 3](#_Toc72098307)

[Глава 1. Характеристика объекта практики. 4](#_Toc72098308)

[1.1. Технико – экономическая характеристика. 4](#_Toc72098309)

[1.2. Аппаратное обеспечение 7](#_Toc72098310)

[1.3. Программное обеспечение 8](#_Toc72098311)

[Глава 2. Теоретическая часть. 12](#_Toc72098312)

[2.1. Методы проектирования. 12](#_Toc72098313)

[2.2. Математическая постановка задачи 15](#_Toc72098314)

[2.3. Программные решения 17](#_Toc72098315)

[Глава 3. Проектная часть. 25](#_Toc72098316)

[3.1. Технологии обработки данных 25](#_Toc72098317)

[3.2. Алгоритм решения 26](#_Toc72098318)

[3.3. Выбор инструментов 27](#_Toc72098319)

[3.4. Тестирование 31](#_Toc72098320)

[Глава 4. Организационно-экономическая часть 34](#_Toc72098321)

[4.1. Руководство оператора 34](#_Toc72098322)

[4.2. Раздел техники безопасности 34](#_Toc72098323)

[Дневник практики 40](#_Toc72098324)

[Источники. 41](#_Toc72098325)

# Введение

На 4 курсе обучения в ККМТ студентом группы П1-17 Лихторенко Олесей была пройдена преддипломная практика.

Студент получил и выполнил задание от руководителя, так же изучил организацию, для которой будет разработано данное приложение.

# Глава 1. Характеристика объекта практики.

## **Технико – экономическая характеристика.**

**Характеристика мастерской**

1. Полное название учебно-научной мастерской:

«Программные решения для бизнеса»

2. Базовое подразделение (кафедра): ККМТ

3. Год создания мастерской: 2019

4. Месторасположение мастерской: Московская область, г. Королёв, ул. Пионерская, д.8

5. Заведующий мастерской (ФИО, должность)

Попов Вячеслав Николаевич, зав. мастерской

6. Специализация мастерской: проведение занятий по закреплённым за лабораторией дисциплинам.

7. Кадровый потенциал лаборатории:

**Таблица 1.**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Ф.И.О. | Должность | Образование, ученая степень, ученое звание |
| 1. | Попов Вячеслав Николаевич | Заведующий мастерской | высшее |

8. Деятельность мастерской. Мастерской доступны следующие виды деятельности:

1) Образовательная. Проведение учебных дисциплин и курсов

2) Научная. Выполнение научных исследований студентами, аспирантами, докторантами, преподавателями и сотрудниками

9. Учебные дисциплины и учебные курсы, проводимые на базе мастерской:

**Таблица 2.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название дисциплины (учебного курса) | Направление подготовки | Кол-во часов | Преподаватель |
| 1. | МДК 03.01 Технология разработки программного обеспечения | 09.02.03 | 392 | Попов В.Н. |
| 2. | Операционные системы | 09.02.04 | 588 | Попов В.Н. |
| 3. | Архитектура компьютерных систем | 09.02.03 | 268 | Попов В.Н. |
| 4. | Технические средства информатизации | 09.02.03 | 74 | Коптилин Р.М. |
| 5. | МДК 03.03 Документирование и сертификация | 09.02.03 | 276 | Бобкова Н.Ю. |

10. Количество студентов, прошедших обучение на базе лаборатории:

**Таблица 3**.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Направление подготовки | Учебный год | Количество студентов |
| 1. | 09.02.03 Программирование в компьютерных системах | 2019 – 2020 | 114 |
| 2. | 09.02.04 Информационные системы (по отраслям) | 2019 - 2020 | 60 |
| 3. | 09.02.03 Программирование в компьютерных системах | 2020 - 2021 | 133 |

**Окончание таблицы 3.**

|  |  |  |  |
| --- | --- | --- | --- |
| 4. | 09.02.04 Информационные системы (по отраслям) | 2020 - 2021 | 60 |
| 5. | 10.02.04 «Обеспечение информационной безопасности телекоммуникационных систем» (по отраслям) | 2020-2021 | 60 |

11. Количество рабочих мест: 20

12. Среднее количество пар в неделю, проводимых в данной мастерской (период март-апрель 2021): 22

**Планировка мастерской 313б**

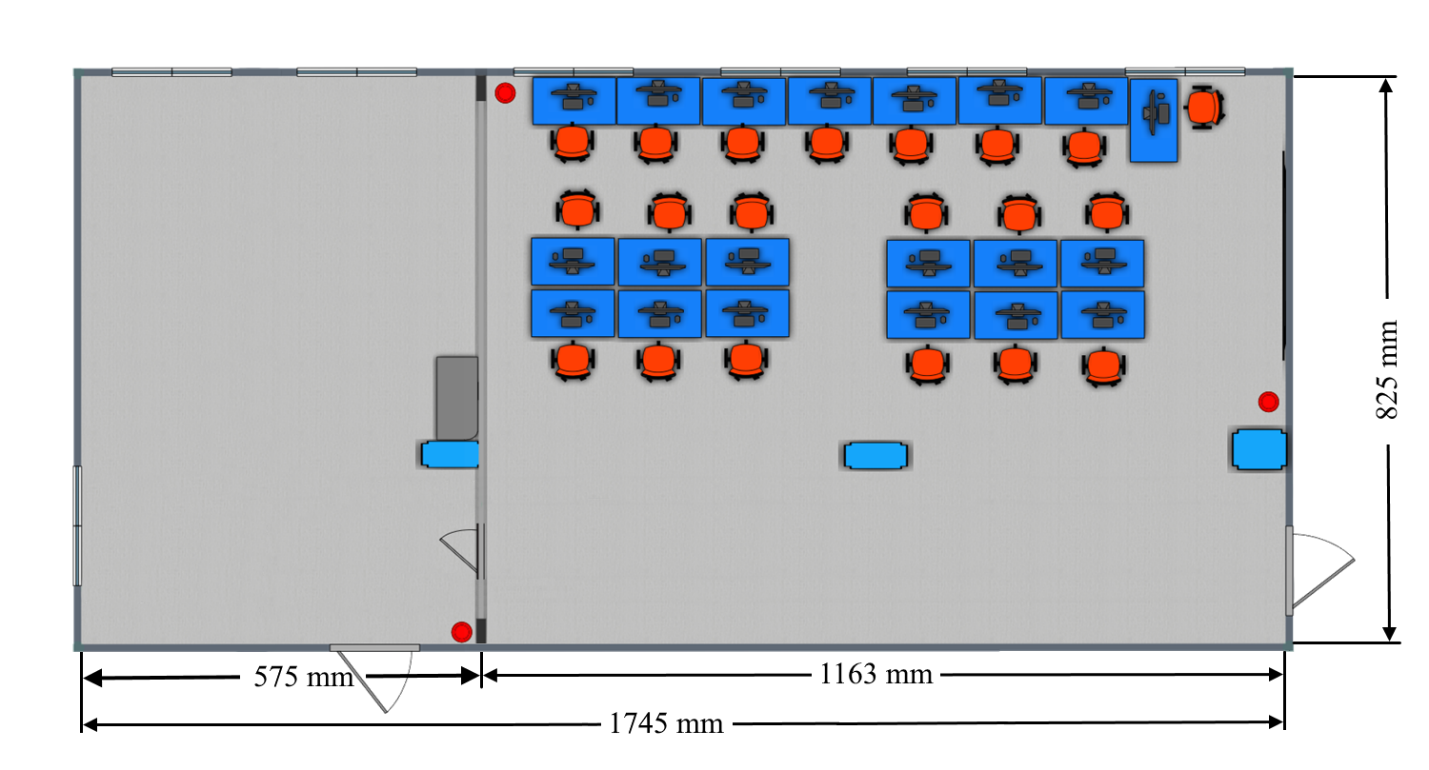


Рисунок 1. Общая схема

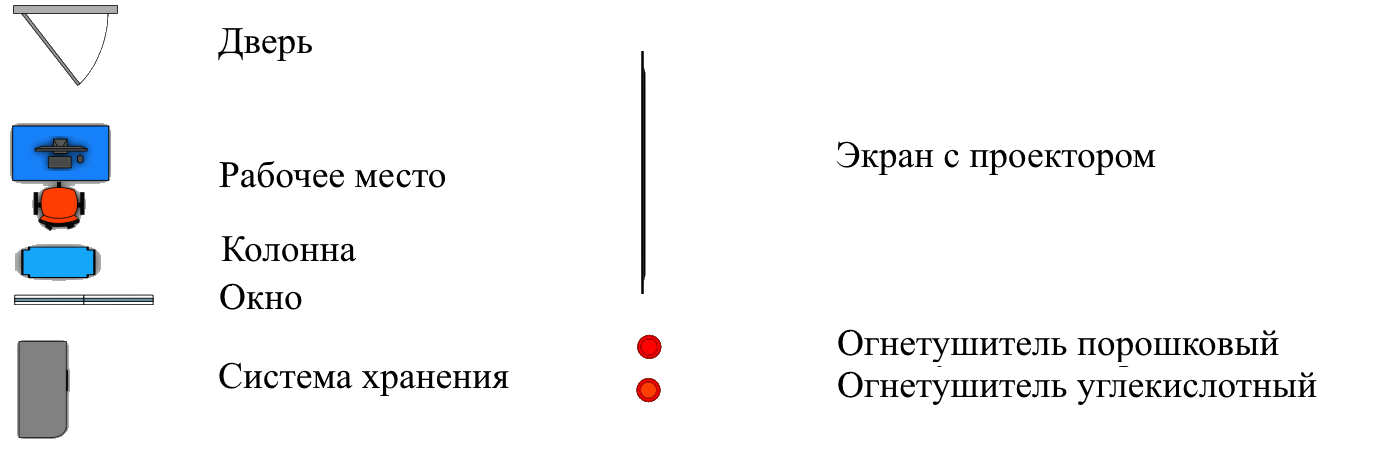


Рисунок 2. Легенды

## **1.2. Аппаратное обеспечение**

Аппаратное обеспечение мастерской состоит из:

1. Автоматизированного рабочего места (20 шт.)

ЦПУ:

- Intel Core i7-9700;

- базовая тактовая частота 3.0 ГГц;

- количество физических ядер 8;

- количество потоков 8;

ОЗУ:

- 16 Гб;

ПЗУ:

- SSD объемом 500 Гб, HDD объемом 1000 Гб;

сетевого адаптера:

- технология Ethernet стандарта 1000BASE-T.

монитора ЖКД Dell p2419h с диагональю 24"

2. Проектора Panasonic PT-VW360

3. Планшетного компьютера Lenovo TB-X605L (20 шт.)

4. Коммутатора catalyst 2960 si

5. Коммутатора AT-GS950/16

6. Сервера Team Server R2-E52 в комплектации:

- Процессор -  2хIntel Xeon Gold 5218 LGA 3647 22Mb 2.3Ghz

- Оперативная память - 256Gb RAM

- Жесткий диск - 6Tb SSD

Схема локальной сети мастерской соответствует топологии «звезда».

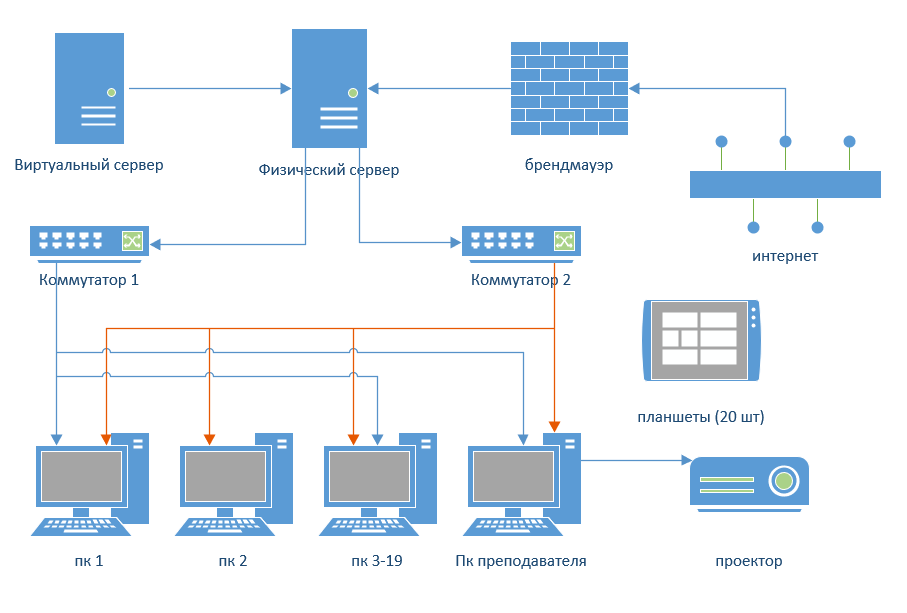


Рисунок 3. Схема локальной сети мастерской 313б.

## **1.3. Программное обеспечение**

Программное обеспечение мастерской «Программные решения для бизнеса» состоит из перечня продуктов, приведенных в Таблице 4 (Программное обеспечение мастерской).

**Таблица 4**

Программное обеспечение мастерской:

|  |  |  |  |
| --- | --- | --- | --- |
| № | Наименование | Количество лицензий | Вебсайт продукта |
| 1. 1 | MS Windows 10 Pro Edu | 20 | https://www.microsoft.com/ |
| 1. 2 | MS Office 2013 Pro Plus | 20 | https://www.microsoft.com/ |
| 1. 3 | 7-Zip 19.00 (x64) | 20 | https://www.7-zip.org/ |
| 1. 4 | Adobe Acrobat Reader DC - Russian | 20 | https://www.adobe.com/ru/ |
| 1. 5 | Anaconda3 2019.10 (Python 3.7.4 64-bit) | 20 | https://docs.anaconda.com/ |
| 1. 6 | Android Studio | 20 | https://developer.android.com/studio |
| 1. 7 | Apache NetBeans IDE 11.2 | 20 | https://netbeans.apache.org/ |
| 1. 10 | Git version 2.24.1.2 | 20 | https://git-scm.com/ |
| 1. 11 | Google Chrome | 20 | https://www.google.com/intl/ru/chrome/ |

**Продолжение Таблицы 4**

|  |  |  |  |
| --- | --- | --- | --- |
| 1. 12 | IntelliJ IDEA Community Edition 2019.3 | 20 | https://www.jetbrains.com/ |
| 1. 13 | Java 8 Update 231 (64-bit) | 20 | https://www.java.com/ |
| 1. 14 | Java (TM) SE Development Kit 15 (64-bit) | 20 | https://www.oracle.com/ |
| 1. 15 | JetBrains PyCharm Community Edition 2019.2.1 | 20 | https://www.jetbrains.com/ |
| 1. 16 | Lightshot-5.5.0.4 | 20 | https://lightshot.ru.uptodown.com/ |
| 1. 17 | Microsoft .NET Framework 3.5 Targeting Pack (rus) | 20 | https://www.microsoft.com/ |
| 1. 18 | Microsoft ODBC Driver 17 for SQL Server | 20 | https://www.microsoft.com/ |
| 1. 19 | Microsoft SQL Server Management Studio - 18.4 | 20 | https://www.microsoft.com/ |
| 1. 20 | Microsoft System CLR Types для SQL Server 2019 CTP2.2 | 20 | https://www.microsoft.com/ |
| 1. 21 | Microsoft Visio - ru-ru | 20 | https://www.microsoft.com/ |
| 1. 22 | Microsoft Visual Studio 2010 Tools for Office Runtime (x64) | 20 | https://www.microsoft.com/ |
| 1. 23 | Microsoft Visual Studio Installer | 20 | https://visualstudio.microsoft.com/ru/ |
| 1. 25 | MySQL Installer - Community | 20 | https://dev.mysql.com/downloads/ |
| 1. 26 | MySQL Workbench 8.0 CE | 20 | https://dev.mysql.com/downloads/ |

**Окончание Таблицы 4**

|  |  |  |  |
| --- | --- | --- | --- |
| 1. 27 | NetBeans IDE 8.2 RC | 20 | https://netbeans.apache.org/ |
| 1. 28 | Notepad++ (64-bit x64) | 20 | https://notepad-plus-plus.org/ |
| 1. 29 | OBS Studio | 20 | https://obsproject.com/ru |
| 1. 30 | Oracle VM VirtualBox 6.1.18 | 20 | https://www.virtualbox.org/ |
| 1. 31 | Python Launcher | 20 | https://www.python.org/ |
| 1. 32 | Ramus | 20 | http://ramussoftware.com/ |
| 1. 33 | Visual Studio Community 2017 | 20 | https://visualstudio.microsoft.com/ru/vs/ |
| 1. 34 | Visual Studio Professional 2019 | 20 | https://visualstudio.microsoft.com/ru/vs/ |

# Глава 2. Теоретическая часть.

## **2.1. Методы проектирования.**

В качестве метода проектирования приложения была выбрана архитектура «сверху вниз».

При использовании подхода сверху-вниз изначальная задача делится на части, которые каждая в свою очередь может состоять из подзадач. Таким образом, делятся полученные задачи на ещё меньшие подзадачи до тех пор, пока основная задача не будет состоять из простых и однозначно понятных подзадач. Недостатком такого подхода является то, что в случае больших программных решений его нецелесообразно использовать, поскольку основная задача в таком случае делится на огромное количество подзадач, в случае чего возникают ситуации, когда одна и та же проблема решается несколько раз (потому что некоторые подзадачи требуют решения похожих проблем). В дополнение к этому инженер ПО должен довольно рано начинать думать над конкретными алгоритмами, при помощи которых поставленные задачи могут быть решены.

Плюсы и минусы использования метода проектирования «сверху вниз»

Плюсы:

* Организация целенаправленно внедряет решение для ресурсов, связанных с отдельным управляемым приложением.
* Первая стадия реализации сразу демонстрирует весь спектр возможностей решения по управлению идентификационной информацией.
* По завершении всех этапов реализации для управляемого приложения вы получите более глубокое и зрелое решение по управлению идентификационной информацией.
* На начальных этапах этот метод не так сильно затрагивает рабочие операции и технический персонал, как в случае реализации "снизу-вверх".

Минусы:

* На первых этапах решение охватывает только ограниченную область.
* На первых этапах управление охватывает небольшой процент учетных записей пользователей.
* Возможно, вам придется разработать пользовательские адаптеры на ранних этапах.
* Служба поддержки и предприятие в целом не смогут быстро ощутить преимущества решения.
* Вероятно, стоимость реализации окажется выше.

На основании всех перечисленных плюсов и незначительных для нашего проекта минусов, мы решили выбрать этот метод.

**Метод разработки**

В качестве метода разработки приложения была выбрана модель жизненного цикла «итеративная» или «инкрементальная».

Итеративная или инкрементальная модель предполагает разбиение создаваемой системы на набор кусков, которые разрабатываются с помощью нескольких последовательных проходов всех работ или их части.

На первой итерации разрабатывается фрагмент системы, не зависящий от других. При этом большая часть или даже полный цикл работ осуществляются на нем, затем оцениваются результаты, и на следующей итерации либо первый фрагмент переделывается, либо разрабатывается следующий, который может зависеть от первого, либо как-то совмещается доработка первого фрагмента с добавлением новых функций. В результате на каждой итерации можно анализировать промежуточные результаты работ и реакцию на них всех заинтересованных лиц, включая пользователей, и вносить корректирующие изменения на следующих итерациях. Каждая итерация может содержать полный набор видов деятельности от анализа требований до ввода в эксплуатацию очередной части ПО.

Итеративный процесс предполагает, что разные виды деятельности не привязаны намертво к определенным этапам разработки, а выполняются по мере необходимости, иногда повторяются до тех пор, пока не будет получен нужный результат.

Плюсы и минусы использования итеративного метода разработки

Плюсы:

* Снижение рисков — раннее обнаружение конфликтов между требованиями, моделями и реализацией проекта; большая фокусировка на основных задачах; динамическое формирование требований и управление ими.
* Организация эффективной обратной связи проектной команды с потребителем, создание продукта, реально отвечающего его потребностям.
* Быстрый выпуск минимально ценного продукта (MVP) и возможность вывести продукт на рынок и начать эксплуатацию гораздо раньше.

Минусы:

* Проблемы с архитектурой и накладные расходы — при работе с хаотичными требованиями и без проработанного глобального плана архитектура приложения может пострадать, а на её приведение к адекватному виду могут потребоваться дополнительные ресурсы. По сути, за возможность менять требования в ходе создания продукта, приходится так или иначе расплачиваться.
* Нет фиксированного бюджета и сроков, а также нужна сильная вовлеченность заказчика в процесс — для некоторых заказчиков это неприемлемые условия сотрудничества с разработчиком, им лучше подойдёт водопадная модель.

При выборе метода разработки, наиболее подходящего для нашего проекта, мы опирались на следующие пункты:

* Эффективная обратная связь с игроками (потребителями).
* Динамическое формирование требований и управление ими.
* Создание продукта, отвечающего ожиданиям игрока.
* Быстрый выпуск демоверсии.

Возможность вывести продукт на рынок как можно раньше.

## **2.2. Математическая постановка задачи**

**Формула передвижения игрового персонажа:**

**d = v \* s \* deltatime \* r**

где

* **d** – расстояние пройденное игровым объектом,
* **v** – вектор направления,
* **s** – скорость объекта,
* **deltatime** – время между текущим в предыдущим кадром.
* **r** – специальный коэффициент учета укореняя

## 

## **2.3. Программные решения**

**Описание существующих разработок.**

В игровой индустрии существует огромное количество разнообразных жанров. В качестве примера рассмотрим несколько игр:

1. Flappy Bird

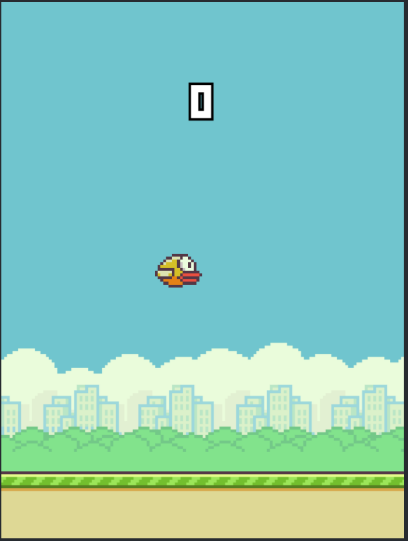


Рисунок 2.1. Запуск игры.

Flappy Bird - игра-платформер для мобильных устройств, разработанная вьетнамским разработчиком Донгом Нгуеном, в которой Была реализована на платформах [iOS](https://ru.wikipedia.org/wiki/IOS" \o "IOS) и [Android](https://ru.wikipedia.org/wiki/Android" \o "Android).

Игра была выпущена [24 мая](https://ru.wikipedia.org/wiki/24_%D0%BC%D0%B0%D1%8F) [2013 года](https://ru.wikipedia.org/wiki/2013_%D0%B3%D0%BE%D0%B4_%D0%B2_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D1%85_%D0%B8%D0%B3%D1%80%D0%B0%D1%85) и получила внезапный рост популярности в начале 2014 года. Была подвергнута критике за сложный игровой процесс, плагиат графики и игровой механики, другие обозреватели посчитали, что она вызывает зависимость. В конце января 2014 стала самой скачиваемой бесплатной игрой в [App Store](https://ru.wikipedia.org/wiki/App_Store" \o "App Store). В течение этого времени разработчик зарабатывал $50 000 в день за рекламу внутри приложения.

[10 февраля](https://ru.wikipedia.org/wiki/10_%D1%84%D0%B5%D0%B2%D1%80%D0%B0%D0%BB%D1%8F) [2014 года](https://ru.wikipedia.org/wiki/2014_%D0%B3%D0%BE%D0%B4) Flappy Bird была удалена разработчиком из [App Store](https://ru.wikipedia.org/wiki/App_Store" \o "App Store) и [Google Play](https://ru.wikipedia.org/wiki/Google_Play" \o "). После этого появилась тенденция продавать смартфоны с заранее установленной игрой, чаще всего за крупную сумму. Это также вызвало появление множества клонов игры, создаваемых разработчиками в надежде повторить успех оригинала.

Сюжет игры:

Игрок с помощью касаний экрана должен контролировать полёт птицы между рядами зелёных труб, не задевая их.

Плюсы игры:

* простота и понятность;
* одна попытка у большинства не занимает более 5-15 секунд;
* вес, пойдет на любом телефоне;
* 8битный стиль, а именно: графика, звуки, атмосфера. Трубы и звук взяты из марио, всеми любимому еще с далекой нинтендо, создается эффект ностальгии.

Ссылка: https://flappybird.io/

# Глава 3. Проектная часть.

## **3.1. Технологии обработки данных**

Для понимания проходящих процессов по разработке компьютерных игр была разработана функциональная модель бизнеc-процесса с использованием нотации IDEF0 с помощью программного продукта MS Visio (см. Рисунок 3.1. Бизнес-процесс разработки игрового приложения). Диаграмма позволяет наглядно увидеть, какие исходные данные необходимы для выполнения каждого фукнкционального блока и что является результатом выполнения тех или иных процедур.

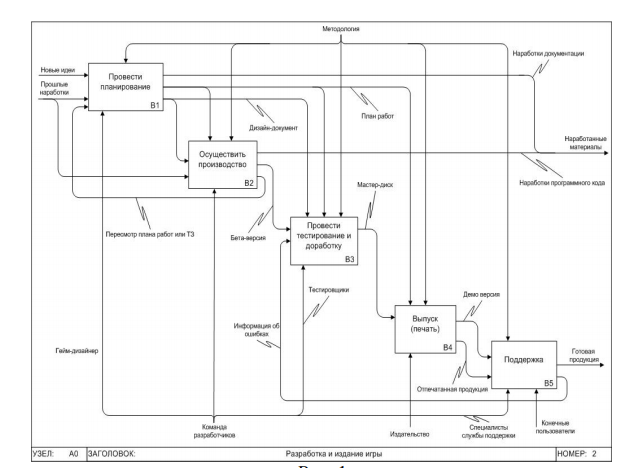
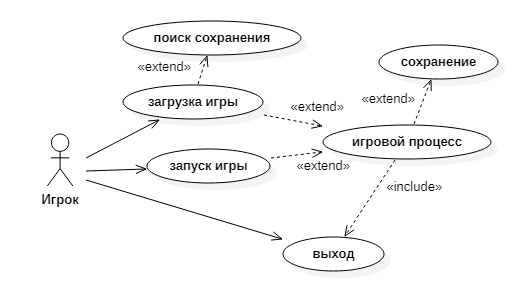


Рисунок 3.1. Бизнес-процесс разработки игрового приложения.

## **3.2. Алгоритм решения**

Для более наглядного представления алгоритма решения программы была создана диаграмма прецедентов, которая представлена ниже на рисунке 3.2(Диаграмма прецедентов), отражающая отношения между акторами (в данном случае это игроки) и прецедентами (действия игроков) и являющаяся составной частью модели прецедентов.

Рисунок 3.2. Диаграмма прецедентов.

**Таблица 3.1.**

Описание прецедентов

|  |  |
| --- | --- |
| Прецедент | Краткое описание |
| Загрузка игры | Запускается игроком. Включает в себя расширяемый прецедент «поиск сохранения», «игровой процесс». |

**Окончание таблицы 3.1.**

|  |  |
| --- | --- |
| Запуск игры | Запускается игроком. Включает в себя расширяемый прецедент «игровой процесс». |
| Выход | Запускается игроком. |

## **3.3. Выбор инструментов**

Выбор инструментов для разработки игрового приложения «Passing», в первую очередь, руководствовался наличием уже установленного и подходящего под условия разработки программного обеспечения на компьютере мастерской «Программные решения для бизнеса», а именно:

1) Среда разработки Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов. Сравнительный анализ сред разработки приведен в Таблице 3.2. (Сравнение сред разработки).

Преимущества IDE-среды Visual Studio:

* Встроенный Web-сервер;
* Поддержка множества языков при разработке;
* Меньше кода для написания;
* Интуитивный стиль кодирования;
* Более высокая скорость разработки;
* Возможности отладки;

2) Язык программирования – С#. Это объектно-ориентированный язык программирования, разработанный группой инженеров компании Microsoft как язык разработки приложений для платформы Microsoft.NET Framework, а также поддерживается средой разработки Microsoft Visual Studio. Сравнительный анализ языков программирования приведен в Таблице 3.3.(Сравнение языков программирования).

Преимущества:

* это объектно-ориентированный, простой и в то же время мощный язык программирования, который позволяет разработчикам создавать многофункциональные приложения.
* C# относится к языкам компилируемого типа, поэтому он обладает всеми преимуществами таких языков.
* Из-за большого разнообразия синтаксических конструкций и возможности работать с платформой .Net, C# позволяет быстрее, чем любой другой язык, разрабатывать программные решения.
* C# отличается надежностью и элегантностью из-за большого разнообразия синтаксических конструкций

3) Игровой движок – Unity. На основании выбора среды разработки и языка программирования, далее был выбран и в последствии установлен игровой движок. Unity – это среда для разработки компьютерных игр, в которой объединены различные программные средства, используемые при создании программного обеспечения – текстовый редактор, компилятор, отладчик и так далее. При этом, благодаря удобству использования, Unity делает создание игр максимально простым и комфортным, а мультиплатформенность движка позволяет охватить как можно большее количество игровых платформ и операционных систем. Сравнительный анализ игровых движков приведен в Таблице 3.4 (Сравнение игровых движков).

Преимущества:

- возможность разрабатывать игры, не требуя для этого каких-то особых знаний. Здесь используется компонентно-ориентированный подход, в рамках которого разработчик создает объекты (например, главного героя) и к ним добавляет различные компоненты (например, визуальное отображение персонажа и способы управления им). Благодаря удобному Drag & Drop интерфейсу и функциональному графическому редактору движок позволяет рисовать карты и расставлять объекты в реальном времени и сразу же тестировать получившийся результат.

- наличие огромной библиотеки ассетов и плагинов, с помощью которых можно значительно ускорить процесс разработки игры

- поддержка огромного количества платформ, технологий, API. Созданные на движке игры можно легко портировать между ОС Windows, Linux, OS X, Android, iOS, на консоли семейств PlayStation, Xbox, Nintendo, на VR- и AR-устройства. Unity поддерживает DirectX и OpenGL, работает со всеми современными эффектами рендеринга, включая новейшую технологию трассировки лучей в реальном времени.

- имеет бесплатную версию движка

**Таблица 3.2**

Сравнение сред разработки

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Критерий | MS Visual Studio | Xcode | Code::Blocks | NetBeans |
| 1. | Функциональность | 9 | 9 | 6 | 6 |
| 2. | Удобство использования | 10 | 9 | 7 | 8 |
| 3. | Системные ресурсы | 5 | 7 | 8 | 7 |
| 4. | Доступность | 9 | 9 | 10 | 10 |
| 5. | Портативность | 8 | 5 | 7 | 8 |
| 6. | Надежность | 8 | 7 | 7 | 7 |
| Итого: | | 49 | 46 | 45 | 46 |

По результатам анализа была выбрана среда разработки MS Visual Studio, как наиболее многофункциональная удобная IDE для командной разработки.

Вебсайт разработчика: https://visualstudio.microsoft.com/ru/

**Таблица 3.3.**

Сравнение языков программирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Критерий | С | С++ | С# | Java |
| 1. | Скорость обработки | 8 | 9 | 7 | 6 |
| 2. | Читабельность | 5 | 6 | 7 | 7 |
| 3. | Графика (3D) | 5 | 6 | 8 | 6 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4. | Кросс-платформенность | 6 | 6 | 4 | 7 |
| 5. | Простота | 2 | 3 | 6 | 5 |
| Итого: | | 26 | 30 | 32 | 31 |

По результатам анализа был выбран язык программирования С#, как наиболее подходящий для разработки игрового приложения.

**Таблица 3.4**

Сравнение игровых движков

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Критерий | Unity 3D | Unreal Engine 4 | Source Engine SDK | Cry Engine 3 |
| 1. | Стоимость | 10(  Indie-версия: бесплатно  Pro-версия: 1500$) | 7(Бесплатный для некоммерческого  Использования) | 5(Лицензия проприетарная) | 7(Бесплатный для некоммерческого  Использования) |
| 2. | Поддержка платформ | 10 | 8 | 6 | 6 |

**Продолжение таблицы 3.4.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 3. | Поддержка языков | 9 (С#, Java script, Boo) | 0 (Собственный скриптовый язык UnrealScript) | 3 (С++) | 5 (С++ и скриптовый язык Lua) |
| 4. | Результат | 10 (Прекрасная картинка, не требующая при создании больших познаний в коде) | 8 (Выигрывает в визуальном плане (частицы, реакция объектов)) | 7 (Хорошая анимация персонажей и транспорта) | 6 (Продвинутые технологии, визуального впечатления не производит) |
| Итого: | | 39 | 23 | 21 | 24 |

По результатам анализа был выбран игровой движок Unity 3D как наиболее подходящий мультиплатформенный игрвоой движок для разработки игрового приложения.

Вебсайт разработчика: https://unity.com/ru

## **3.4. Тестирование**

**Тестирование программы**

Тестовый набор №1:

Действие: после запуска игры пользователь взаимодействует с сенсором.

Ожидаемый результат: персонаж в игре не залипает в текстурах и движется с определённым интервалом.

Тестовый набор №2:

Действие: игрок нажимает на клавишу «Space».

Ожидаемый результат: при нажатии клавиши «Space» персонаж совершает прыжок на определенное расстояние.

Тестовый набор №3:

Действие: пользователь собирает по игровой локации «очки» за прохождение определенной дистанции.

Ожидаемый результат: счетчик очков засчитывает «очки» в соответствии с их собранным количеством.

Тестовый набор №4:

Действие: пользователь нажимает клавишу «Esc».

Ожидаемый результат: при нажатии пользователем клавиши «Esc» появляется главное меню запуска игры

**3.5. Главный модуль программы**

Описание работы формы главного модуля игрового приложения «Passing» и ее компонентов схематично отображено ниже (Рисунок 3.5. Главный модуль игрового приложения).

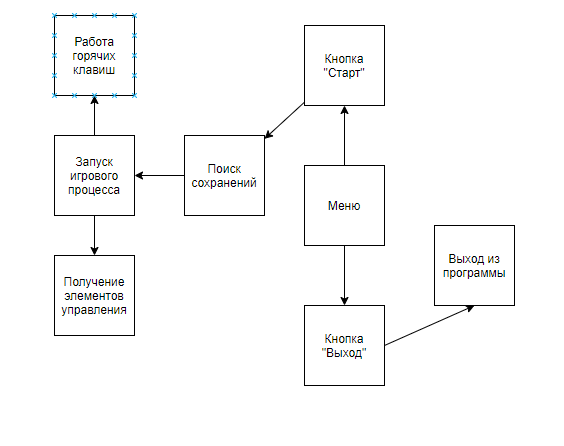


Рисунок 3.5. Главный модуль игрового приложения.

# Глава 4. Организационно-экономическая часть

## **4.1. Руководство оператора**

Руководство оператора для приложения находится в файле «руководство оператора.docx»

## **4.2. Раздел техники безопасности**

1. Общие требования безопасности

1.1. К работе на персональном компьютере допускаются лица, прошедшие обучение безопасным методам труда, вводный инструктаж, первичный инструктаж на рабочем месте.

1.2. При эксплуатации персонального компьютера на работника могут оказывать действие следующие опасные и вредные производственные факторы:

- повышенный уровень электромагнитных излучений;

- повышенный уровень статического электричества;

- пониженная ионизация воздуха;

- статические физические перегрузки;

- перенапряжение зрительных анализаторов.

1.3. Работник обязан:

1.3.1. Выполнять только ту работу, которая определена его должностной инструкцией.

1.3.2. Содержать в чистоте рабочее место.

1.3.3. Соблюдать режим труда и отдыха в зависимости от продолжительности, вида и категории трудовой деятельности (Приложение 1).

1.3.3. Соблюдать меры пожарной безопасности.

1.4. Рабочие места с компьютерами должны размещаться таким образом, чтобы расстояние от экрана одного видеомонитора до тыла другого было не менее 2,0 м, а расстояние между боковыми поверхностями видеомониторов - не менее 1,2 м.

1.5. Рабочие места с персональными компьютерами по отношению к световым проемам должны располагаться так, чтобы естественный свет падал сбоку, преимущественно слева.

1.6. Оконные проемы в помещениях, где используются персональные компьютеры, должны быть оборудованы регулируемыми устройствами типа: жалюзи, занавесей, внешних козырьков и др.

1.7. Рабочая мебель для пользователей компьютерной техникой должна отвечать следующим требованиям:

- высота рабочей поверхности стола должна регулироваться в пределах 680 - 800 мм; при отсутствии такой возможности высота рабочей поверхности стола должна составлять 725 мм;

- рабочий стол должен иметь пространство для ног высотой не менее 600 мм, глубиной на уровне колен не менее 450 мм и на уровне вытянутых ног не менее 650 мм;

- рабочий стул (кресло) должен быть подъемно - поворотным и регулируемым по высоте и углам наклона сиденья и спинки, а также - расстоянию спинки от переднего края сиденья;

- рабочее место должно быть оборудовано подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20 градусов; поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм;

- рабочее место с персональным компьютером должно быть оснащено легко перемещаемым пюпитром для документов.

1.8. Для нормализации аэроионного фактора помещений с компьютерами необходимо использовать устройства автоматического регулирования ионного режима воздушной среды (например, аэроионизатор стабилизирующий "Москва-СА1").

1.9. Женщины со времени установления беременности и в период кормления грудью к выполнению всех видов работ, связанных с использованием компьютеров, не допускаются.

1.10. За невыполнение данной Инструкции виновные привлекаются к ответственности согласно правилам внутреннего трудового распорядка или взысканиям, определенным Кодексом законов о труде Российской Федерации.

2. Требования безопасности перед началом работы

2.1. Подготовить рабочее место.

2.2. Отрегулировать освещение на рабочем месте, убедиться в отсутствии бликов на экране.

2.3. Проверить правильность подключения оборудования к электросети.

2.4. Проверить исправность проводов питания и отсутствие оголенных участков проводов.

2.5. Убедиться в наличии заземления системного блока, монитора и защитного экрана.

2.6. Протереть антистатической салфеткой поверхность экрана монитора и защитного экрана.

2.7. Проверить правильность установки стола, стула, подставки для ног, пюпитра, угла наклона экрана, положение клавиатуры, положение "мыши" на специальном коврике, при необходимости произвести регулировку рабочего стола и кресла, а также расположение элементов компьютера в соответствии с требованиями эргономики и в целях исключения неудобных поз и длительных напряжений тела.

3. Требования безопасности во время работы

3.1. Работнику при работе на ПК запрещается:

- прикасаться к задней панели системного блока (процессора) при включенном питании;

- переключать разъемы интерфейсных кабелей периферийных устройств при включенном питании;

- допускать попадание влаги на поверхность системного блока (процессора), монитора, рабочую поверхность клавиатуры, дисководов, принтеров и других устройств;

- производить самостоятельное вскрытие и ремонт оборудования;

- работать на компьютере при снятых кожухах;

- отключать оборудование от электросети и выдергивать электровилку, держась за шнур.

3.2. Продолжительность непрерывной работы с компьютером без регламентированного перерыва не должна превышать 2-х часов.

3.3. Во время регламентированных перерывов с целью снижения нервно - эмоционального напряжения, утомления зрительного анализатора, устранения влияния гиподинамии и гипокинезии, предотвращения развития познотонического утомления выполнять комплексы упражнений.

4. Требования безопасности в аварийных ситуациях

4.1. Во всех случаях обрыва проводов питания, неисправности заземления и других повреждений, появления гари, немедленно отключить питание и сообщить об аварийной ситуации руководителю.

4.2. Не приступать к работе до устранения неисправностей.

4.3. При получении травм или внезапном заболевании немедленно известить своего руководителя, организовать первую доврачебную помощь или вызвать скорую медицинскую помощь.

5. Требования безопасности по окончании работы

5.1. Отключить питание компьютера.

5.2. Привести в порядок рабочее место.

5.3. Выполнить упражнения для глаз и пальцев рук на расслабление.

Приложение 1

**ВРЕМЯ РЕГЛАМЕНТИРОВАННЫХ ПЕРЕРЫВОВ  
В ЗАВИСИМОСТИ ОТ ПРОДОЛЖИТЕЛЬНОСТИ РАБОЧЕЙ СМЕНЫ,  
ВИДА И КАТЕГОРИИ ТРУДОВОЙ ДЕЯТЕЛЬНОСТИ  
С ПЕРСОНАЛЬНЫМ КОМПЬЮТЕРОМ**

1. Виды трудовой деятельности разделяются на 3 группы: группа А - работа по считыванию информации с экрана компьютера с предварительным запросом; группа Б - работа по вводу информации; группа В - творческая работа в режиме диалога с компьютером. При выполнении в течение рабочей смены работ, относящихся к различным видам трудовой деятельности, за основную работу с компьютером следует принимать такую, которая занимает не менее 50% времени в течение рабочей смены или рабочего дня.

2. Для видов трудовой деятельности устанавливается 3 категории тяжести и напряженности работы с компьютером, которые определяются: для группы А - по суммарному числу считываемых знаков за рабочую смену (не более 60000 знаков за смену); для группы Б - по суммарному числу считываемых или вводимых знаков за рабочую смену (не более 40000 знаков за смену); для группы В - по суммарному времени непосредственной работы с компьютером за рабочую смену (не более 6 часов за смену).

3. При 8-часовой рабочей смене и работе на компьютере регламентированные перерывы следует устанавливать:

- для I категории работ через 2 часа от начала рабочей смены и через 2 часа после обеденного перерыва продолжительностью 15 минут каждый;

- для II категории работ через 2 часа от начала рабочей смены и через 1,5 - 2,0 часа после обеденного перерыва продолжительностью 15 минут каждый или продолжительностью 10 минут через каждый час работы;

- для III категории работ - через 1,5 - 2,0 часа от начала рабочей смены и через 1,5 - 2,0 часа после обеденного перерыва продолжительностью 20 минут каждый или продолжительностью 15 минут через каждый час работы.

4. При 12-часовой рабочей смене регламентированные перерывы должны устанавливаться в первые 8 часов работы аналогично перерывам при 8-часовой рабочей смене, а в течение последних 4 часов работы, независимо от категории и вида работ, каждый час продолжительностью 15 минут.

Сайт: https://mvf.klerk.ru/spr/spr89.htm

# Дневник практики

|  |  |  |
| --- | --- | --- |
| Дата | Содержание работ | Отметка о выполнении |
| 20.04.2021 | Изучение и обсуждение официальных документов |  |
| 21.04.2021 | Выдача заданий на преддипломную практику |  |
| 22.04.2021 – 24.04.2020 | Обсуждение задания на преддипломную практику и начало его выполнения  Написание игрового сценария |  |
| 25.04.2021 – 27.04.2021 | Разработка элементов пользовательского интерфейса |  |
| 28.04.2021 – 03.05.2021 | Разработка графических изображений – заднего фона, персонажей и препятствий |  |
| 04.05.2021 – 06.05.2021 | Нахождение звукового и музыкального сопровождения |  |
| 07.05.2021 – 09.05.2021 | Проведена доработка и усовершенствование основный игровых механик |  |
| 10.05.2021 – 13.05.2021 | Проведена доработка и усовершенствование пользовательского интерфейса с учетом появившихся графических изображений |  |
| 14.05.2021 – 16.05.2021 | Разработаны руководство оператора и отчет по преддипломной практике |  |
| 17.05.2020 | Защита отчётов по практике |  |

# Источники.

1. https://docs.microsoft.com/ru-ru/dotnet/csharp/
2. https://visualstudio.microsoft.com/ru/vs/getting-started/
3. https://planerka.info/item/diagramma-precedentov-variantov-ispolzovaniya-uml/
4. https://intuit.ru/studies/courses/2195/55/lecture/15043