

**ОТЧЕТ**

по производственной практике ПП.01.01 по модулю ПМ.01  
«Разработка программных модулей программного обеспечения   
для компьютерных систем»

по специальности 09.02.03 «Программирование в компьютерных системах»

Студент 3 курса группы П2-17

Форма обучения: очная

Гришаев Андрей Михайлович

Место прохождения практики

Государственном бюджетном образовательном учреждении высшего образования Московской области «Технологический университет»

(название организации)

Срок прохождения практики с 13 января 2020 г. по 15 марта 2020 г.

Руководители практики

От организации: заведующий мастерской \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Попов В.Н.

(Должность) (Подпись) (ФИО)

От колледжа: преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Родичкин П.Ф.

(подпись)

Итоговая оценка по практике \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Оглавление**

[**Введение** 3](#_Toc41566407)

[**1.** **Общие сведения о организации.** 4](#_Toc41566408)

[**1.1.** **Структура организации характеристика основных видов деятельности**. 4](#_Toc41566409)

[**1.2.** **Должностные обязанности оператора ЭВМ, техника – программиста, инженера – программиста.** 4](#_Toc41566410)

[**1.2.1.** **Должностные обязанности оператора ЭВМ.** 4](#_Toc41566411)

[**1.2.2.** **Должностные обязанности техника – программиста.** 5](#_Toc41566412)

[**1.2.3.** **Должностные обязанности инженера – программиста.** 5](#_Toc41566413)

[**1.3.** **Основные функции отдела.** 6](#_Toc41566414)

[**1.4.** **Документооборот предприятия, структурного подразделения.** 7](#_Toc41566415)

[**2.** **Содержание выполняемых видов работ** 9](#_Toc41566416)

[**2.1.** **Разработка спецификаций отдельный компонентов**. 9](#_Toc41566417)

[**2.2.** **Коды для игры.** 9](#_Toc41566418)

[**2.3. Отрисовка спрайтов персонажей, объектов и фонов**…………………………………1](#_Toc41566419)6

[**2.4** **Музыка** 17](#_Toc41566420)

[**2.5** **Часть разрабатываемая студентом Растопчиным А. Р.** 25](#_Toc41566421)

[**3** **Обращение с программой** 18](#_Toc41566422)

[**4.** **Выводы** 18](#_Toc41566423)

[**5.** **Заключение** 19](#_Toc41566424)

[**6.** **Дневник практики** 20](#_Toc41566425)

[**7.** **Список использованной литературы.** 20](#_Toc41566426)

[**8.** **Приложения.** 21](#_Toc41566427)

# 

# **Введение**

На 3 курсе обучения в ККМТ, студентом группы П2-17 Гришаевым Андреем была проведена производственная практика по модулю ПМ.01  
«Разработка программных модулей программного обеспечения   
для компьютерных систем». Студент получил задание разработать игру.

Во время прохождения практики я поставил для себя следующие цели:

* Приобрести опыт работы по специальности.
* Закрепить теоретические знания, полученные во время учебы.
* Проанализировать работы отдела.
* Закрепить навыки в разработке проектной и технической документации.
* Закрепить навыки отладки и тестирования программных модулей.

Для выполнения вышеупомянутых мной целей я выдвинул следующие задачи:

* Изучить специфику деятельности организации.
* Установить необходимые инструменты для работы.
* Найти подходящую литературу.

# **Общие сведения о организации.**

## **Структура организации характеристика основных видов деятельности**.

Данное предприятие работает в сфере образования. Университет образован 16 июля 1998 года в форме некоммерческой организации с названием: Негосударственное образовательное учреждение «Королевская академия управления, экономики и социологии».

Технологический университет (ранее Финансово-технологическая академия; Королевский институт управления, экономики и социологии) создан для подготовки кадров новой информации, воспроизводства интеллектуальных ресурсов, формирования инновационных проектов и технологий. Академия находится в наукограде Королеве Московской области – уникальном центре интеллектуальных ресурсов, которые используются для интеграции важнейших знаний и создания систем глобального масштаба.

20 января 2015 года постановлением Правительства Московской области Академии присвоен статус «университета» и вуз переименован в Государственное бюджетное образовательное учреждение высшего образования Московской области «Технологический университет».

Организационная структура колледжа представлена на Рис. 6.1 в Приложении 1.

## **Должностные обязанности оператора ЭВМ, техника – программиста, инженера – программиста.**

### **Должностные обязанности оператора ЭВМ.**

* осуществляет техническую подготовку документации, необходимой в процессе работы компании. Выполняет копирование документов на ксероксе;
* выполняет набор различных текстов с соблюдением правил орфографии и пунктуации, а также стандартов оформления организационно-распорядительной документации;
* осуществляет работу с электронной почтой, принимает входящие электронные письма и следит за своевременной отправкой исходящих;
* распечатывает и систематизирует нужные документы;
* заносит в компьютерные базы данных различную информацию, важную и необходимую для работы компании;
* следит за состоянием компьютера и копировальной техники;
* своевременно информирует руководство о необходимости приобретения материалов, непосредственно относящихся к производственному процессу.

### **Должностные обязанности техника – программиста.**

* выполняет работу по обеспечению механизированной и автоматизированной обработки, поступающей в ВЦ (ИВЦ) информации, разработки технологии решения экономических и других задач производственного и научно-исследовательского характера;
* принимает участие в проектировании систем обработки данных и систем математического обеспечения машины;
* выполняет подготовительные операции, связанные с осуществлением вычислительного процесса, ведет наблюдение за работой машин;
* составляет простые схемы технологического процесса обработки информации, алгоритмы решения задач, схемы коммутации, макеты, рабочие инструкции и необходимые пояснения к ним;
* разрабатывает программы решения простых задач, проводит их отладку и экспериментальную проверку отдельных этапов работ;
* выполняет работу по подготовке технических носителей информации, обеспечивающих автоматический ввод данных в вычислительную машину, по накоплению и систематизации показателей нормативного и справочного фонда, разработке форм исходящих документов, внесению необходимых изменений и своевременному корректированию рабочих программ;
* участвует в выполнении различных операций технологического процесса обработки информации (прием и контроль входной информации, подготовка исходных данных, обработка информации, выпуск исходящей документации и передача ее заказчику);
* ведет учет использования машинного времени, объемов выполненных работ;
* выполняет отдельные служебные поручения своего непосредственного руководителя.

### **Должностные обязанности инженера – программиста.**

* на основе анализа математических моделей и алгоритмов решения экономических и других задач разрабатывает программы, обеспечивающие возможность выполнения алгоритма и соответственно поставленной задачи средствами вычислительной техники, проводит их тестирование и отладку;
* разрабатывает технологию решения задач по всем этапам обработки информации;
* осуществляет выбор языка программирования для описания алгоритмов и структур данных;
* определяет информацию, подлежащую обработке средствами вычислительной техники, ее объемы, структуру, макеты и схемы ввода, обработки, хранения и вывода, методы ее контроля;
* выполняет работу по подготовке программ к отладке и приводит отладку;
* определяет объем и содержание данных контрольных примеров, обеспечивающих наиболее полную проверку соответствия программ их функциональному назначению;
* осуществляет запуск отлаженных программ и ввод исходных данных, определяемых условиями поставленных задач;
* проводит корректировку разработанной программы на основе анализа выходных данных;
* разрабатывает инструкции по работе с программами, оформляет необходимую техническую документацию;
* определяет возможность использования готовых программных продуктов;
* осуществляет сопровождение внедрения программ и программных средств;
* разрабатывает и внедряет системы автоматической проверки правильности программ, типовые и стандартные программные средства, составляет технологию обработки информации;
* выполняет работу по унификации и типизации вычислительных процессов;
* принимает участие в создании каталогов и картотек стандартных программ, в разработке форм документов, подлежащих машинной обработке, в проектировании программ, позволяющих расширить область применения вычислительной техники.

## **Основные функции отдела.**

* Производственно-технологическая: разработка алгоритма решения задачи на основе предложенной модели; программная реализация алгоритма; отладка и тестирование программных продуктов; модификация программных продуктов; адаптация и настройка программных продуктов; сопровождение программных продуктов; разработка и эксплуатация баз данных; обеспечение достоверности информации при использовании баз данных;
* Организационно-управленческая: организация работы коллектива исполнителей; планирование и организация работ; выбор оптимальных решений при планировании работ в условиях нестандартных ситуаций; участие в оценке качества и экономической эффективности деятельности; обеспечение техники безопасности.

## **Документооборот предприятия, структурного подразделения.**

Документооборот Отдела в сфере поставленной мне на практике задачи состоит из нескольких этапов:

* получение приказа и распределение работы между сотрудниками;
* перечень существующих дел в Отделе;
* годовой план работ;
* годовой отчет по проделанной работе.

Вид построенной IDEF модели по плану документооборота представлен на рисунках 1 – 3:

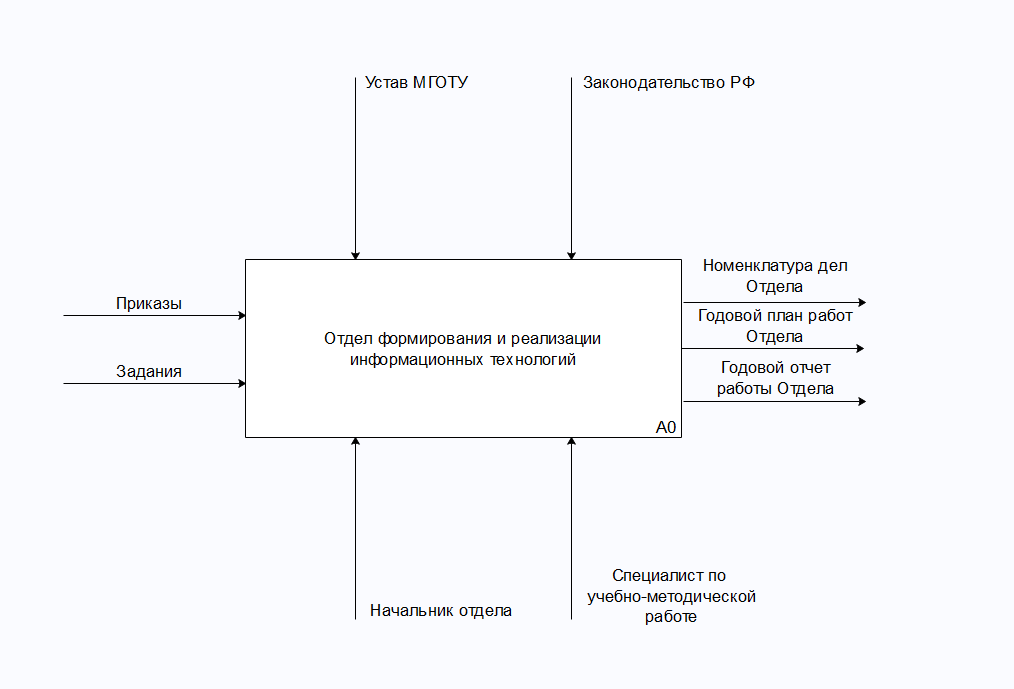


Рис. 1. - IDEF - модель 1 уровень

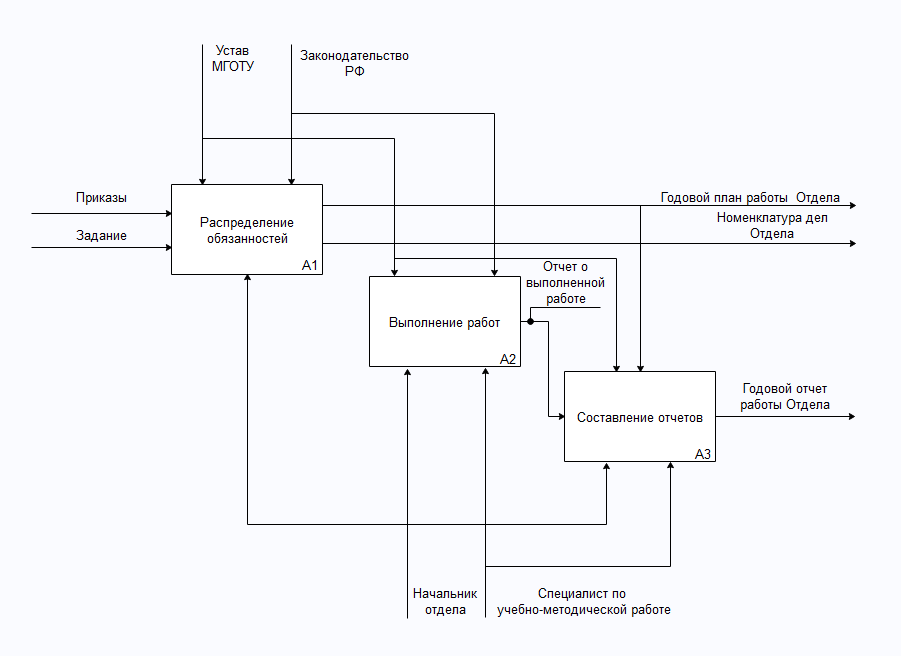


Рис. 2 - IDEF - модель 2 уровень

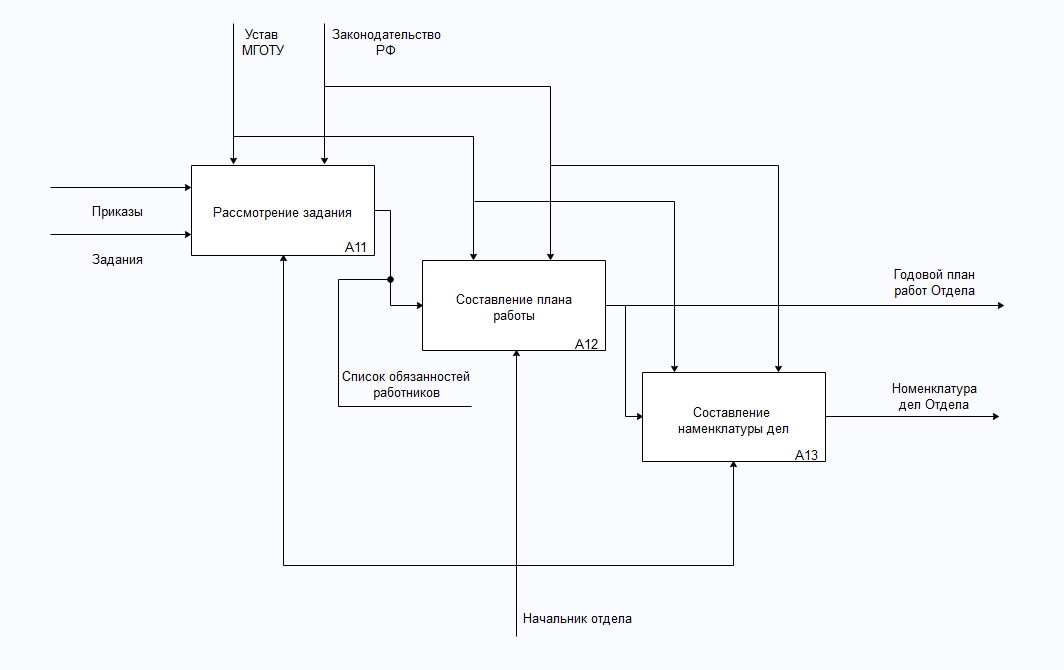


Рис. 3 – IDEF – модель подуровня блока «Распределение задания.

# **Содержание выполняемых видов работ**

## **Разработка спецификаций отдельный компонентов**.

Общее задание было разделено на 3 этапа:

1. Концепция. На основе выданного задания было принято решение писать игру в жанре top-down survival shooter на движке Game maker 2. .
2. Написание кода игры.
3. Отрисовка спрайтов для персонажей, объектов и фонов.
4. Написание музыки.

## **Коды для игры.**

**Листинг 1. Персонаж, передвижение, анимация, выстрел.**

**Объект playerosn.**

Это невидимая основа главного героя, которая отвечает за передвижение, и к которой прикрепляется объект player1, смотрящий за мышкой.

//Событие step. Перемещение объекта. Переменная объекта player1 Speed1 отвечает за прокачиваемую скорость.

if (keyboard\_check(ord("D"))){

x += 2 + player1.Speed1

}

if (keyboard\_check(ord("A"))){

x -= 2 + player1.Speed1

}

if (keyboard\_check(ord("W"))){

y -= 2 + player1.Speed1

}

if (keyboard\_check(ord("S"))){

y += 2 + player1.Speed1

}

**Объкт player1.**



Видимая часть гг. Прикреплена к объекту playerosn. Смотрит за мышкой.

// Событие Create(то есть, что происходит при создании).

str = 0; //Переменная для стрельбы.

//Дальше переменные для прокачки.

Uron1 = 0;

Hp1 = 0;

Speed1 = 0;

//Событие step(то есть, что происходит всегда.)

x = playerosn.x;

y = playerosn.y;

image\_angle = point\_direction(x, y, mouse\_x, mouse\_y);

//код для стрельбы, завязанный на анимации.

if ((sprite\_index == ggstr) and (image\_index == 2)){

sprite\_index = gg;

str = 0;

}

//Событие Global Left Mouse Pressed(при нажатии кнопки мыши).

if (str == 0){

instance\_create\_depth(x, y, 0, buletpl); //Создание пулию

instance\_create\_depth(x, y, 0, vystrell); //Создание вспышки от выстрела

sprite\_index = ggstr;

str = 1

}

**Объект картинки мыши mysh.**

x = mouse\_x;

y = mouse\_y;

**Объект buletpl. Пуля.**

//Событие create. Поворот пули при создании, учитывая разброс, полет.

move\_towards\_point(mouse\_x-50+random(100),mouse\_y-50+random(100),50);

image\_angle = point\_direction(x, y, mouse\_x, mouse\_y);

**Объект vystrel. Вспышка от выстрела.**

//Событие Step. При окончании анимации уничтожается.

image\_angle = point\_direction(x, y, mouse\_x, mouse\_y);

if image\_index > 1 {

instance\_destroy();}

**Объект obj\_hppp. Полоска жизней.**

//Событие step. Полоска жизней привязана к игроку. При их окончании перенос в комнату с концом игры.

x = player1.x;

y = player1.y;

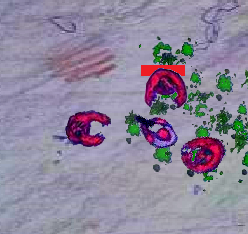
if (image\_xscale <= 0){

room = gameover;

}

**Листинг 2. Монстры, волны, прокачка.**

**Объект monster. Вражеские npc.**



Идут в сторону игрока, имеют два поведения, наносят урон.

//Событие create. Задаем переменные.

hp = 4;

sprite\_index = mn1;

spawn.exist = 1; //Переменная, означающая, что монстр существует на экране.

tolkuchka = 3; //Переменная для того, чтоб монстры не накладывались друг на друга, превращаясь в один объект.

//Событие destroy. Создает объект большой зеленой лужи при уничтожении.

instance\_create\_depth(x, y, 0 , monsterd);

//Событие step.

//ИИ монстра. Двигается за героем. По-разному ведет себя в зависимости от растояния до игрока.

if (obj\_hppp.image\_xscale > 0){

if (sprite\_index == mn1){

if (distance\_to\_object(player1) < 200){

move\_towards\_point( player1.x, player1.y, 2 );

image\_angle = point\_direction(x, y, player1.x, player1.y);

}

else{

move\_towards\_point( player1.x - 400 + random(800), player1.y - 400 + random(800), 4 );

image\_angle = point\_direction(x, y, player1.x, player1.y);

}

}

}

//Переключение анимаций.

if ((sprite\_index == mn1pain) and (image\_index == 2)){

sprite\_index = mn1;

}

spawn.exist = 1;

//Код для того чтоб монстры не накладывались друг на друга.

if tolkuchka <= 0 {

tolkuchka = 3;

x = x - random(5) + random(10);

y = y - random(5) + random(10);

}

//Событие Collision with buletpl.

sprite\_index = mn1pain;

move\_towards\_point( player1.x, player1.y, random(3) );

image\_angle = point\_direction(x, y, player1.x, player1.y);

hp = hp - (1 + player1.Uron1); //Из жизней вычетается урон + урон от прокачки.

//Создание брызгов и зеленых луж.

repeat (5){

instance\_create\_depth(x, y, 0, obj\_blood);

}

instance\_create\_depth(x,y, 0, meat);

//Жизни меньше или равны нулю.

if (hp <= 0){

instance\_destroy()

spawn.exist = 0;

}

//Событие On Collision With monster. Изменение переменной для того чтоб монстры не накладывались друг на друга.

tolkuchka = tolkuchka - 0.02;

//Событие On Collision With player1

//Включает анимацию, при которой он не наносит урон и бежит вперед.

//Так-же отнимает жизни у героя и создает объекты зеленых брызгов и луж.

if (obj\_hppp.image\_xscale > 0){

if (sprite\_index == mn1){

sprite\_index = mn1pain;

move\_towards\_point( player1.x, player1.y, random(3) );

image\_angle = point\_direction(x, y, player1.x, player1.y);

obj\_hppp.image\_xscale -= 0.1;

repeat (5){

instance\_create\_depth(x, y, 0, obj\_blood);

}

instance\_create\_depth(x,y, 0, meat);

}

}

**Объект monserd. Большая лужа после смерти монстра.**

//Событие step. Нерозрачность постоянно уменьшается. Когда она на нуле - объект уничтожается.

image\_alpha = image\_alpha - 0.0005;

if image\_alpha = 0{

instance\_destroy();}

**Объект obj\_blood. Зеленые лужи разных размеров.**

//Событие Create. Рандомные переменные для спрайтов и растяжения объекта

rand = random(0.9);

x = x - 30 + random(60)

y = y - 30 + random(60)

image\_index = random(7);

image\_xscale -= rand;

image\_yscale -= rand;

image\_alpha = 1;

//Событие step. Нерозрачность постоянно уменьшается. Когда она на нуле - объект уничтожается.

image\_alpha = image\_alpha - 0.0005;

if image\_alpha = 0{

instance\_destroy();}

**Объект spawn. Спавнеры монстров.**

//Событие Create. Переменные, для рандомного создания волн врагов, которые в процессе становятся сильнее.

wait = 0;

spaw = 1;

spawred = 3;

spawreda = spawred;

wave = 1;

exist = 0;

//Событие Step. Время игры, когда идут волны.

if wait = 0 {

HP.visible = false;

URON.visible = false;

SPEED.visible = false;

if spaw > 0 {

spawreda = spawreda - wave \* (random(0.222)-random(0.202));

if spawreda <= 0 {

instance\_create\_depth(x,y,-1,monster);

spaw = spaw - 1;

spawreda = spawred;

}

}

}

//Когда волна кончилась. Восстановление hp, и появление кнопок для прокачки.

if (spaw <= 0 && exist = 0) {

wait = 1;}

if wait = 1{

obj\_hppp.image\_xscale = 1 + player1.Hp1;

HP.visible = true;

URON.visible = true;

SPEED.visible = true;

}

**Объект HP. Кнопка прокачки между волнами.**



//Событие Left Mouse Down. Прокачка максимальной полоски жизней.

if HP.visible = true{

spawn.wait = 0;

spawn.wave = spawn.wave + 1;

spawn.spaw = spawn.wave;

spawn.spawred = spawn.spawred - 0.01;

spawn.spawreda = spawn.spawred;

player1.Hp1 = player1.Hp1 + 0.5;

obj\_hppp.image\_xscale = 1 + player1.Hp1;

HP.visible = false;

URON.visible = false;

SPEED.visible = false;

}

**Объект URON. Кнопка прокачки между волнами.**

//Событие Left Mouse Down. Прокачка урона.

if URON.visible = true{

spawn.wait = 0;

spawn.wave = spawn.wave + 1;

spawn.spaw = spawn.wave;

spawn.spawred = spawn.spawred - 0.01;

spawn.spawreda = spawn.spawred;

player1.Uron1 = player1.Uron1 + 0.5;

HP.visible = false;

URON.visible = false;

SPEED.visible = false;

}

**Объект SPEED. Кнопка прокачки между волнами.**

//Событие Left Mouse Down. Прокачка скорости.

if SPEED.visible = true{

spawn.wait = 0;

spawn.wave = spawn.wave + 1;

spawn.spaw = spawn.wave;

spawn.spawred = spawn.spawred - 0.01;

spawn.spawreda = spawn.spawred;

player1.Speed1 = player1.Speed1 + 0.5;

HP.visible = false;

URON.visible = false;

SPEED.visible = false;

}

## **Отрисовка спрайтов персонажей, объектов и фонов.**

Было принято решение рисовать все спрайты на бумаге, после этого обрезать белый фон и вставлять в игру.



Вот так выглядят **игровые объекты** на бумаге.



Так-же были нарисованы **стартовый экран** (мной).



И **экран, высвечивающийся при проигрыше** (Екатериной Андреевной Дворецкой).



## **Музыка.**

Для написания главной композиции игры решено было выбрать жанр, подходящий под атмосферу, то есть блэкметалл. Вокалистом выступил Василий Юрьевич Сидоров, я сыграл партии на гитаре и прописал барабанные партии в программе FL studio.

1. **Обращение с программой**

Ввиду особенностей бесплатной версии программы Game Maker 2, конвертация в формат .exe не представляется возможной.

Для запуска программы необходимо выполнить следующие действия:

Иметь в наличии на компьютере:

* Game maker 2.

Необходимо выполнить следующие действия:

* Зарегистрироваться на сайте yoyogames.com
* Скачать бесплатную 30-дневную версию программы.
* Открыть файл “Pounce v okno.yyz”

# **Выводы**

Полученные навыки:

* Работа с программой Game Maker 2.
* Проектирование и создание основных механик.

Полученные умения:

* Перенесение спрайтов с бумаги непосредственно в игру, с помощью программы paint.net.

# **Заключение**

Перед прохождением производственной практики в Государственном бюджетном образовательномучреждении высшего образования Московской области «Технологический университет» мной были поставлены следующие основные цели:

* Приобрести опыт работы по специальности.
* Закрепить теоретические знания, полученные во время учебы.
* Выполнение требований и действий, предусмотренных программой производственной практики и заданий руководителя.
* Закрепить навыки в разработке проектной и технической документации.
* Закрепить навыки отладки и тестирования программных модулей.

Во время прохождения практики я приобрел опыт работы по специальности. Также был закреплен навык разработки проектной и технической документации и навык отладки и тестирования программных модулей.

По окончанию практики был составлен отчёт.

# **Дневник практики**

Таблица 1. Дневник практики.

|  |  |  |  |
| --- | --- | --- | --- |
| Дата | Содержание работы | Отметка о выполнении  работы | Подпись руководителя практики |
| 14.01 – 21.01 | Выбор языка программирования, выбор среды разработки, выбор программного обеспечения. |  |  |
| 22.01 – 5.02 | Cоздание базовых механик для игры. |  |  |
| 6.02 – 21.02 | Работа с Game Maker 2. Создание игрового пространства. Отрисовка спрайтов |  |  |
| 22.02 – 28.02 | Создание игровых объектов и наложение на них спрайтов. |  |  |
| 29.02 – 11.03 | Создание меню и написание музыки. |  |  |
| 12.03 – 16.03 | Оформление отчёта. Обработка листингов, группировка отчёта. |  |  |

# **Список использованной литературы.**

**1.** [**gamemaker.ucoz.com**›index/0-9](https://gamemaker.ucoz.com/index/0-9)

**2.** [**gcup.ru**](https://gcup.ru/publ/gamedev/game_maker_studio_fizika_urok_1_osnovy/1-1-0-549)

**3.** [**gamedevelopment.tutsplus.com**](https://gamedevelopment.tutsplus.com/ru/articles/how-to-learn-gamemaker-studio--gamedev-10911)

# **Приложения.**

**Приложение 1.**

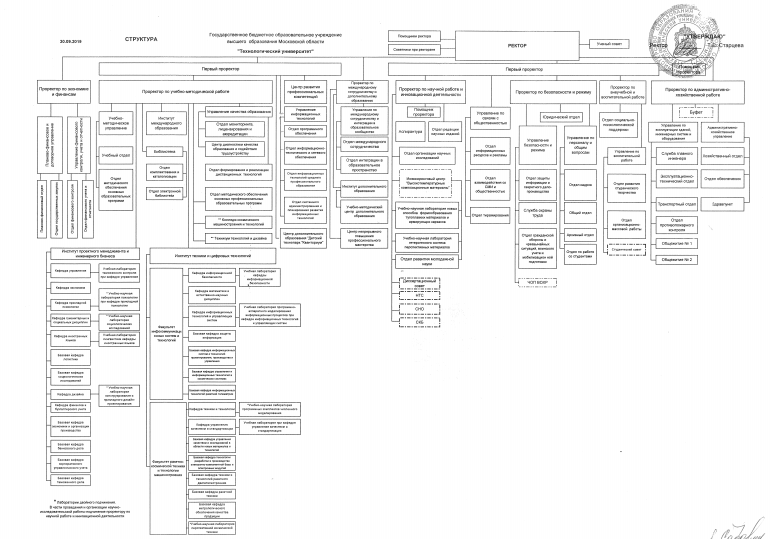


Рис. 6.1. Организационная структура колледжа