|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство образования и науки Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ИУ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ИУ-7\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ОТЧЕТ ПО УЧЕБНОМУ ПРАКТИКУМУ**

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*подпись, дата фамилия, и.о.*

Ментор команды **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*подпись, дата фамилия, и.о.*

Руководитель практики **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*подпись, дата фамилия, и.о.*

*2019 г.*

Оглавление

[Введение 3](#_Toc9361003)

[Аналитический раздел 4](#_Toc9361004)

[Конструкторский раздел 6](#_Toc9361005)

[Технологический раздел 11](#_Toc9361006)

[Личный вклад 13](#_Toc9361007)

[Заключение 15](#_Toc9361008)

[Литература 16](#_Toc9361009)

Введение

На время проведения практики наша команда поставила себе задачу разработать игровое ПО с использованием двумерной графики. Командой было принято решение разработать игру “Bauman Student “.

В рамках практики мы самостоятельно разработали всё необходимое для выполнения задачи программное обеспечение, используя при этом в качестве языка программирования **Python**, и библиотеку **Cocos2D**. А также во время выполнения проекта научились работать в команде, получать практические навыки и умения разработки игрового ПО. Для закрепления, изучаемого на учебных занятиях материала при групповой разработке использовали удаленный репозиторий в Gitlab.

**Актуальность**

Разработка игр является очень перспективной и хорошо оплачиваемой работой. Игры популярны для всех возрастов. На данный момент язык **Python** успешно внедряется во многие сферы разработки ПО и в том числе для разработки игр.

**Цель**

Целью игры является успешное прохождение всех мини-игр, представленных в главном геймплейном блоке.

**Роли членов команды SpalMalo:**

Турсунов Жасурбек - тимлид, разработчик

Казакова Элиза – разработчик

Чаушев Александр – разработчик

Ивченко Артем – тестировщик

Унтилова Арина - гейм-дизайнер 

Аналитический раздел

**Постановка задачи**

Цель проекта - создать игру Bauman Student life на платформе Windows, в которой содержится 4 ключевых геймплейных блока: карта университета, буфет (buffet), спортзал (sport hall), кабинет математики (math world). Студент находится внутри университета и может по нему передвигаться. Когда главный герой заходит в один из выбранных кабинетов, запускается мини игра.

Проект должен быть написан с использованием библиотеки cocos2d. В игре реализованы:

* Интуитивный и понятный интерфейс
* Меню, содержащее кнопки:

1. New Game
2. Quit

* Карта университета, по которой передвигается персонаж.
* Анимация персонажа при движении в разные стороны.
* Секундомер в игре Buffet, отчитывающей время с начала игры
* Недопустимость выхода персонажа за пределы карты (добавление коллайдеров при помощи Tiled Map Editor)
* Появление сообщения об успешном окончании игры

**Обзор аналогов**

Разрабатываемый программный продукт не является уникальной разработкой на данный момент времени. Существует множество аналогов. Но каждый программный продукт создается непосредственно для решения определенного количества задач. Очень сложно найти универсальный программный продукт, который решал бы все необходимые задачи.

### Одним из известных аналогов игры «Bauman Student Life» является “The Sims 3: University Life”, в которой студенты учится, принимают участие в спортивных соревнованиях, сдают экзамены, участвуют в протестах и пр.

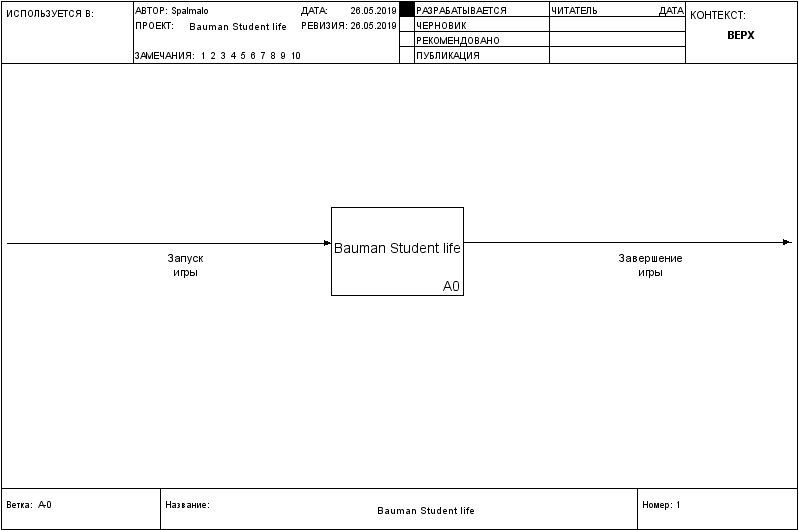
Самый распространенный аналог игры SportHall - Subway Surfers, которое по состоянию на февраль 2016 года загрузили более миллиарда раз.

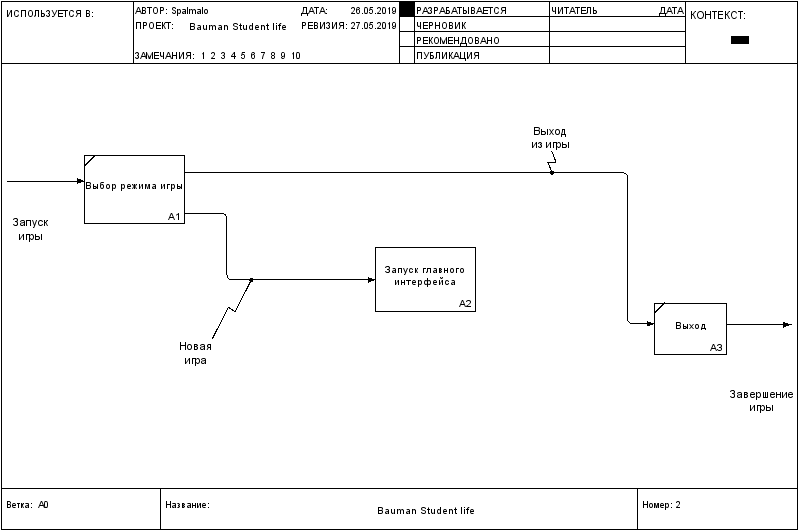
Аналогами игры “Buffet” являются игры-запоминалки (игры-пары).

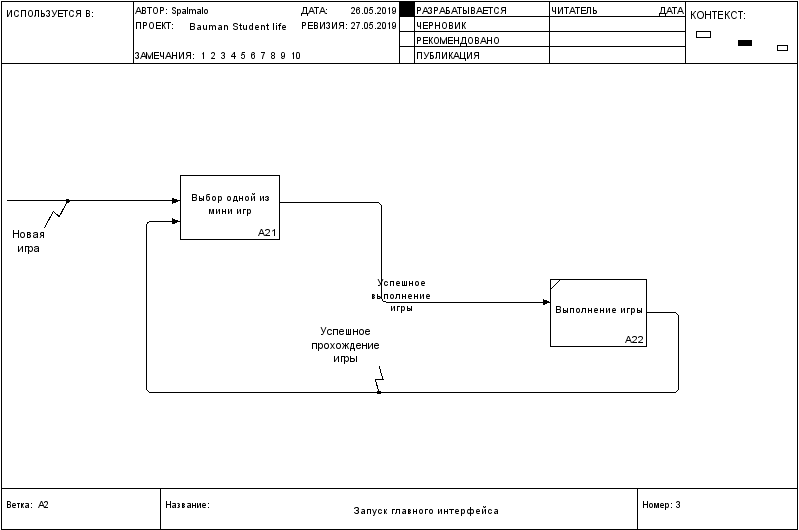
Конструкторский раздел

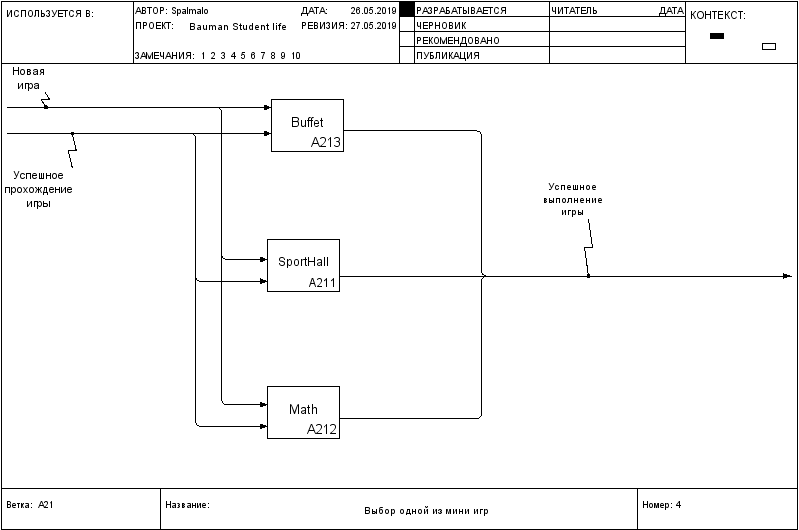
**Декомпозиция**

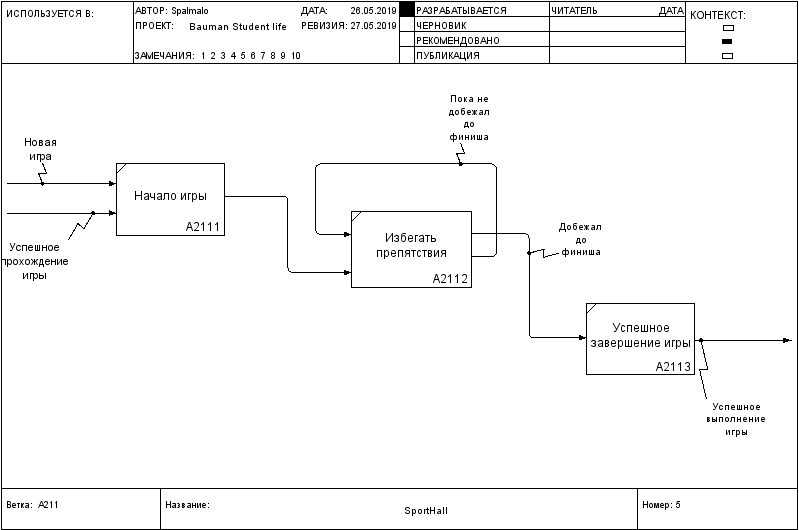
Декомпозиция представляет собой блок под названием «Bauman student life», которая в свою очередь делится на 6 подзадач, а именно на мини-игры.

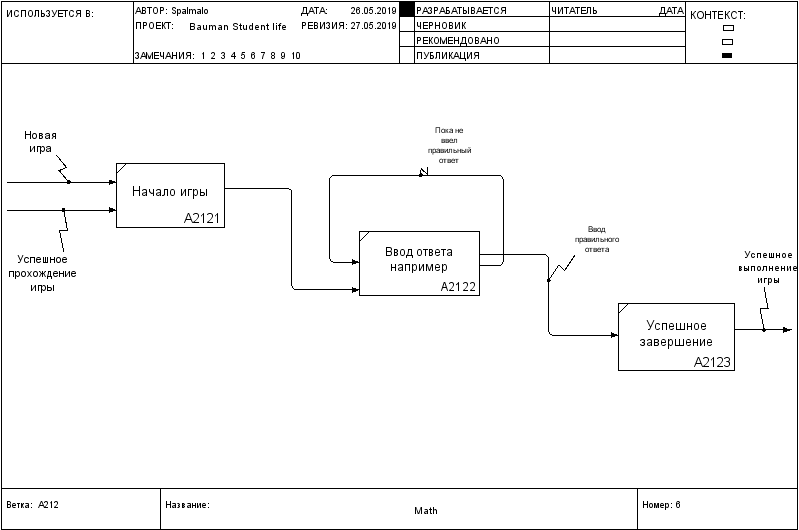


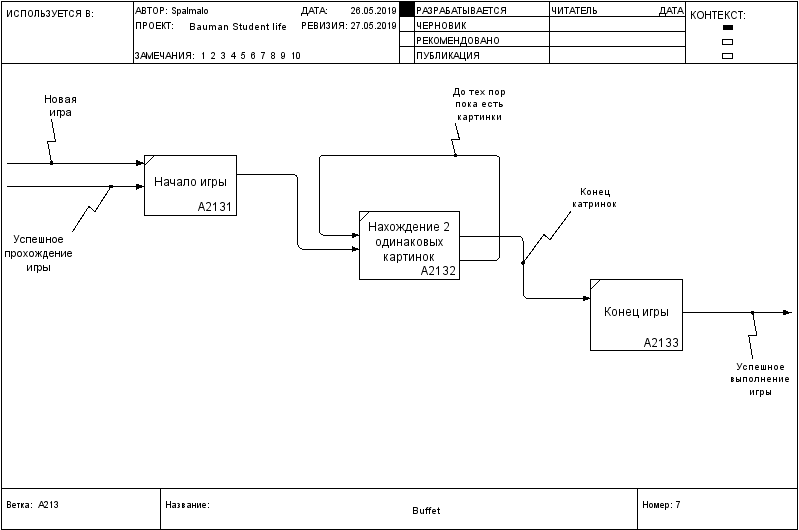












3. Задачи, поставленные перед каждым участником:

**Чаушев Александр** – игра Math, анимация главного персонажа, создать меню, подключить мини игры и коллайдеры.

**Унтилова Арина** – создать карту для всех сцен игр, игра SportHall.

**Казакова Элиза** – игра Buffet, добавить финишную сцену для игры SportHall.

**Турсунов Жасурбек** - исправить баги в SportHall, создать коллайдеры, добавить ссылки в игре.

**Ивчёнко Артем** - создание тестов для каждой из игр(функциональное тестирование и тестирование вручную).

**Разработка структуры создаваемого программного продукта.**

Основными идеями разработки структуры программного продукта являются использование рациональной структуры, которая устраняет проблему сложности разработки; делает программу понятной людям, повышает надежность работы программы при сокращении срока её тестирования, а также сроках её выполнения

**Описание назначения, требований к выделенным компонентам и их интерфейса**

Основными назначениями разработанного ПО являются:

* Умение одновременно следить за несколькими объектами.
* Быстрая и правильная реакция на происходящие события.
* Способность интенсивно работать в течение всего сеанса игры.

**Тестирование**

Важно, чтобы программный продукт работал правильно, в противном случае уровень его защиты, удобства использования, скорость работы не играют никакой роли.

В ходе написания проекта тестированием занимался каждый участник, стараясь подстраивать свой код под уже работающую структуру. Более детальное тестирование проекта на каждом этапе его написания проводилось тестировщиком.

Технологический раздел

**Выбор ЯП и ПС**

В рамках практики требовалось разработать всё программное обеспечение, используя при этом в качестве языка программирования **Python**, а также библиотеку **Cocos2D**, мы использовали еще библиотеку **pyglet**, для создания карт сцен игры использовали приложение **Tiled Map Editor.**

**Реализация тестирования**:

При написании программного кода мы пользовались функциональным тестированием, целью которого являются:

· Проверка правильности работы приложения при различных условиях использования.

· Проверка соответствия продукта требованиям.

Этапами функционального тестирования игры были:

1)Планирование. На этом этапе определялись функциональные требования к продукту, составляется план анализа, определяются методики проверки.

2)Анализ данных. Изучение технических требований, согласование тестов, определение сроков тестирования.

3)Проведение тестирования. На основе полученных данных реализуются тестовые сценарии, в ходе которых выявляются и фиксируются ошибки в работе игры.

Для получения полной картины о работе продукта мы используем 2 метода тестирования:

· Ручное тестирование: подготовка и проведение тестов в ручном режиме.

· Исследовательское: комплексное изучение продукта и выявление дефектов за пределами требований.

Личный вклад

Во время написания проекта передо мной были поставлены следующие задачи:

Создание игры Math

При создании игры Math передо мной была поставлена задача написать классы, отвечающие за генерацию нового примера, отображение написанного к нему ответа при нажатии на цифровые клавиши (встроенная функция на Cocos 2d: on\_key\_press, которая самостоятельно вызывается при нажатии на любую клавишу клавиауры) и появление надписи Wrong или Correct рядом с данным ответом в зависимости от его правильности.

Создание анимации главного персонажа при передвижении

При создании анимации картинка с изображением главного персонажа в разных ракурсах делилась на части (модуль pyglet, функция ImageGrid делает из данного изображения массив маленьких картинок, которые вызываются при повороте персонажа в одну из сторон)

Основная трудность заключалась в подключении созданной анимации к сценам игр.

Создание переходов между сценами игры

Новые сцены в игре добавляются двумя методами: «сверху» и «снизу»

При добавлении сцен "сверху" происходит их накладывание на уже отображаемую сцену на экране

При добавлении сцен "сверху" важно было помнить о том, что для нее необходимо иметь достаточно свободной памяти; поэтому у нас все добавляемые «сверху» легкие.

Создание меню

(функция MenuItem) Меню содержит кнопки «New Game», после нажатия которой происходит переход на главную сцену университета и кнопки «Quit», после нажатия на которую игра закрывается.

Подключение карт с коллайдерами, созданных в Tiled Map Editor к сценам

(Карта с коллайдерами подключается при помощи функции cocos 2d - mapcolliders)

Соединение всех мини-игр и остальных частей программы в один рабочий код.

В Cocos 2d доступно много интерсеных переходов между сценами, я старался не делать затянутых переходов и выбирал тот переход от сцены к сцене, который выглядел наиболее логичным.

Заключение

Нашей командой была создана интересная игра. Изучение чего-то нового всегда интересно и захватывающе, а создать игру в начале обучения вообще казалось невозможным.

В ходе написания проекта была сформулирована и утверждена идея создаваемой игры, разработан алгоритм ее создания, составлена и отлажена программа. Проведенные тестирования показали, что программа работает успешно.

Преимуществом работы в команде заключается в том, что каждый ее участник выполняет определенную часть работы, которая для него наиболее интересна и в которой он имеет наибольший опыт, также каждый разработчик старается написать такой код, чтобы он был понятен всем членам команды. Также полученный ценный опыт командной работы и работа с удалённым репозиторием будет полезна в дальнейшей разработке такого рода проектов.

Литература

* www.python.org
* https://www.youtube.com/watch?v=wicgBgZIUQA&list=LLe21P\_YMCOUFeBnFNReg3wg&index=11&t=1s
* www.cocos2d-x.org
* www.habr.com/post/120438
* www.python.cocos2d.org/doc/index.htm
* http://python.cocos2d.org/doc/programming\_guide/cocosnodes.html#menu
* https://www.raywenderlich.com/2778-how-to-make-a-tile-based-game-with-cocos2d-2-x
* https://python-scripts.com/python-class