|  |  |
| --- | --- |
| 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 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Информатика и системы управления (ИУ)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_Программное обеспечение ЭВМ и информационные технологии (ИУ7)\_\_\_

**ОТЧЕТ ПО УЧЕБНОМУ ПРАКТИКУМУ**

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Садулаева Т.Р.\_\_**

*подпись, дата фамилия, и.о.*

Ментор команды **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Мельников Д.А.\_**

*подпись, дата фамилия, и.о.*

Руководитель практики **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Оленев А.А.\_\_\_\_\_**

*подпись, дата фамилия, и.о.*

*2019 г.*

Оглавление

[Введение 3](#_Toc9361003)

[Аналитический раздел 4](#_Toc9361004)

[Конструкторский раздел 5](#_Toc9361005)

[Технологический раздел 6](#_Toc9361006)

[Личный вклад 7](#_Toc9361007)

[Заключение 8](#_Toc9361008)

[Литература 9](#_Toc9361009)

Введение

**Разработка point&click квеста на Python.** Главной задачей игры является побег из бункера в Измайловском общежитии. Игроку необходимо искать подсказки, решать головоломки и собирать предметы, чтобы открывать двери от уровня к уровню.

**Распределений ролей команды:**

* Садулаева Теона **–** teamlead, разработчик интерфейса
* Мишина Елена – разработка интерфейса, тестирование
* Богаченко Артем – Разработка программного кода
* Тевс Вадим – разработка программного кода, тестирование
* Таламбуца Артем – разработка программного кода, презентация продукта

Мельников Дмитрий – ментор

Аналитический раздел

**Обзор существующих решений.**

The Room

(+) Качественная работа с оформлением

(+) Большое количество головоломок

(-) Высокая цена

(-) Отсутствие сюжета

Масяня

(+) Наличие сюжета

(+) Продолжительная игра

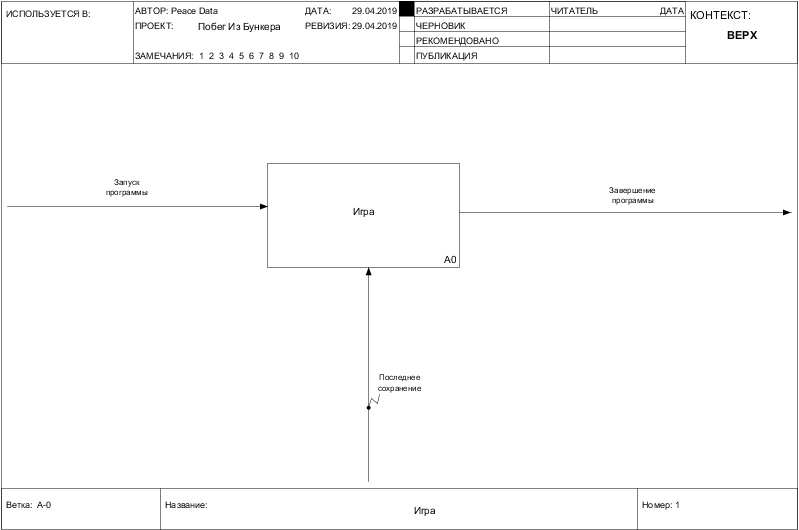
(-) Отсутствие полной версии игры

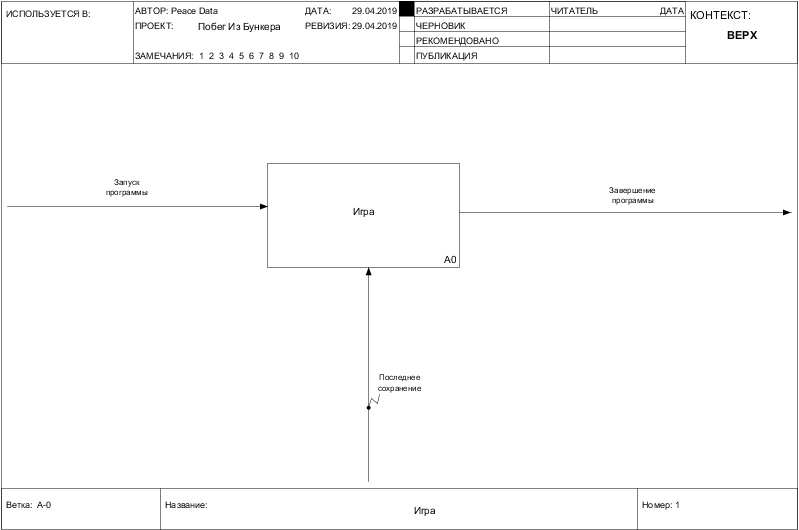
(-) Устаревшие технологии

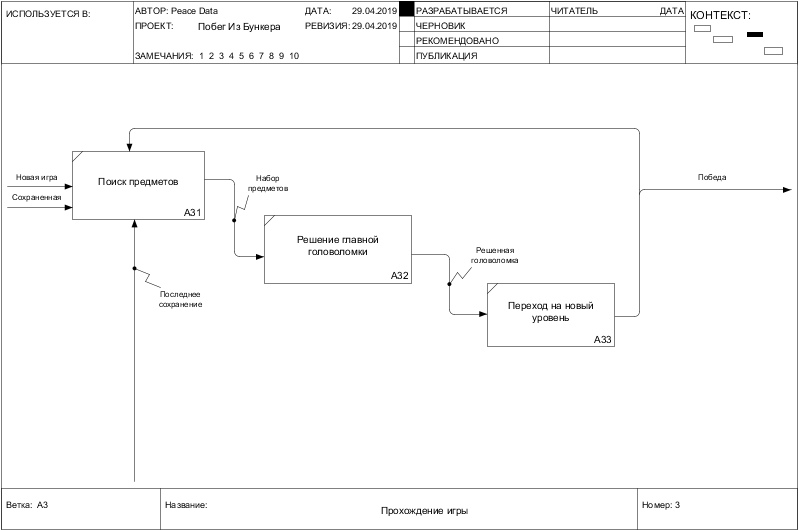
С учетом анализа рынка задача нашей команды – это создание остросюжетного, небольшого и доступного поинт&клик квеста.

Конструкторский раздел

**Декомпозиция задачи.**







**Задачи, поставленные перед каждым участником.**

Садулаева Теона

* Управление кнопками главного меню
* Создание и обработка ресурсов
* Обработка и оформление поясняющих сообщений

Мишина Елена

* Создание настроек, реализация их открытия и закрытия
* Реализация закрытия и открытия инвентаря
* Реализация открытия главного меню во время игры

Богаченко Артем

* Создание основного функционала сцен
* Реализация переключения между уровнями
* Создание основного функционала меню

Тевс Вадим

* Реализация работы с инвентарем
* Управление внутриигровыми интеракциями
* Тестирование программы

Таламбуца Артем

* Презентация проекта
* Создание и управление поясняющими сообщениями
* Тестирование программы

**Разработка структуры создаваемого программного продукта.**

Структура игры разделена на четыре части (сцены): главное меню, уровни игры, инвентарь и настройки.

При первом запуске пользователь знакомится с главным меню и только через него осуществляет передвижение к любой другой сцене. Выход реализован как из меню, так и из настроек.

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

Главное(начальное) меню – компонент, отвечающий за переход к остальным сценам и выход из программы. Появляется в самом начале игры и служит навигационным пунктом новому пользователю. Так как дополнительная информация об игре не предусмотрена, то меню необходимо быть наиболее понятной частью программы.

Основная игра – компонент, реализующий игровой процесс. Приступить к новой игре можно из главного меню, а продолжить – из настроек. В игре поддерживается динамика переходов от одного уровня к другому – из комнаты в комнату. Игра считается законченной, если игрок выходит из программы или проходит все уровни – в этом случае его встречает финальный экран.

Инвентарь – компонент внутри основной игры, осуществляющий сохранение предметов, необходимых для прохождения уровня. В процессе игры необходимо собирать предметы, которые на последующих уровнях можно использовать. Инвентарь выполняет функцию хранения вещей, приобретенных на различных уровнях.

Настройки – компонент, реализующий управление программой в пределах пользовательских прав. Позволяет включать и выключать звук, показатель FPS, переключаться между сценами. Осуществляет выход из игры. Важно считать, что при выходе из игры или в главное меню весь прогресс требуется очистить. То есть при каждом выходе в главное меню программа «впервые» приветствует пользователя. При нажатии команды «назад» в настройках этого не происходит и весь прогресс игрока сохраняется.

Технологический раздел

**Технические средства**

ЯП – Python 3.7. Данный язык был выбран нами т.к. все участники проекта разбираются в нём и умеют с ним работать.

Игровой движок – Cocos2d. Этот движок полностью удовлетворяет нашим требованиям, а также написан на Python, что существенно упрощает работу с ним.

Библиотеки – pyglet. Данная библиотека была выбрана под игровой движок. Она привносит дополнительный функционал, который прекрасно сочетается с игровым движком.

**Выбор и обоснование модели разработки.**

Для разработки нашего относительно небольшого проекта была выбрана каскадная модель. Эта модель проста и понятна, а также мы сразу обговорили и утвердили все требования, что является краеугольным камнем для данной модели (ввиду последовательной разработки).

**Организация выпуска сборок, использование CI.**

Сборки выпускаются по мере создания/улучшения/обновления отдельных частей программы (функционала).

От использования CI было принято решение отказаться, чтобы минимизировать потери времени.

**Реализация программного продукта**

Данный этап разработки программного обеспечения организован в соответствии с моделями эволюционного типа жизненного цикла ПО. При разработке применяются экспериментирование и анализ, строятся прототипы, как целой системы, так и ее частей. Прототипы дают возможность глубже вникнуть в проблему и принять все необходимые проектные решения еще на ранних этапах проектирования. Такие решения могут затрагивать разные части системы: внутреннюю организацию, пользовательский интерфейс, разграничение доступа и т.д. В результате этапа реализации появляется рабочая версия продукта.

**Реализация тестирования**:

В связи с небольшими временными рамками мы решили остановиться на функциональном тестировании. Каждая функциональная часть продукта тестируется на соответствие требованиям. На близком к завершению этапе было рассмотрено сорок два тестовых случая, которые покрывают сто процентов функционального кода.

**Развертывание разработанного программного продукта, инструкция для системного администратора (установка) и пользователя (использование).**

Перед нами не стоял серийный выпуск продукта. Установка и использование происходит в соответствии с техническими средствами.

**Выводы:** Структурированный подход, чёткое видение поставленной задачи.

Личный вклад

**Создание и обработка ресурсов**

В нашей игре нет ни одного заимствования текстур. Все предметы, сцены и кнопки нарисованы мной с помощью программы Procreate.

Но создание ресурсов – не такая кропотливая и тяжелая работа, как ее обработка. Библиотека cocos2d открывает специалисту множество новых возможностей, хотя легкое и удобное редактирование спрайтов – не одно из них. Главной проблемой в процессе тестирования текстур оказалась проблемой разрешения. Статический фон, например, помещать на сцену было легко. Но спрайты, с которыми можно взаимодействовать, очень сложно усмирить.

**Управление кнопками главного меню**

Для успешной работы программы кнопки было необходимо соединить с картинками, фон – с картинками, а функции – с кнопками. Одной из задач и было скомпоновать все ресурсы на одной сцене.

В компоновке одного с другим больших проблем не возникало. Для передачи фона на сцену в библиотеке есть отдельный параметр background\_layer, который изменяет фоновое изображение. Для соединения кнопок с картинками была использована функцию menu.ImageMenuItem, которая задавала необходимое изображение и сразу присваивала ей функцию, как объекту родительского класса MenuItem.

**Обработка поясняющих сообщений**

Проблема поясняющих сообщений звучит сложно, но решается легко. Основным камнем преткновения оказалось не развитие алгоритма, а его выбор. Для каждого предмета следовало завести свое поясняющее сообщение, а значит и новую, уникальную текстовую информацию. Где держать такую информацию?

Самым оптимальным решением казалось завести словарь, ключи которого – предметы, но на деле функционал библиотеки осуществляет вставку текста кодом и работой объемнее, чем вставку изображения. Поэтому было решено преобразовать весь текст в картинки без фона, и держать в словаре не текст, а ссылки на ресурсы репозитория.

Заключение

Поставленные задачи и требования успешно функционируют, хотя в процессе реализации возникало немало трудностей с управлением различными сценами. Одним из самых тяжелых аспектов оказалось создание инвентаря. Проблемы возникли с отображением собранных предметов, их удалением и добавлением на новые сцены.

В процессе работы над задачей мы не раз отказывались от обусловленных алгоритмов, оптимизировали их, или наоборот, усложняли, столкнувшись с границами функционала предоставленных технологических средств. Как итог – игра успешно реализована.

Литература

<https://habr.com/ru/post/339564/>

<https://habr.com/ru/post/120438/>

<http://python.cocos2d.org/doc/index.html>

<https://riptutorial.com/ru/cocos2d-x>

<https://www.youtube.com/watch?v=wicgBgZIUQA>