

Санкт-Петербургский политехнический университет
Институт компьютерных наук и технологий
Кафедра «Компьютерные системы и программные технологии»

КУРСОВОЙ ПРОЕКТ
Разработка игры "Treasure Rush"
по дисциплине «Технологии Программирования»

Выполнил студент

гр. 3530901/20002

Иванов Т. В.

Преподаватель

Степанов Д. С.

30 мая 2023 г.

Санкт-Петербургский политехнический университет

ЗАДАНИЕ

НА ВЫПОЛНЕНИЕ КУРСОВОГО ПРОЕКТА

студенту группы 3530901/20002 Иванову Тимофею Вячеславовичу

1. Тема проекта: Создание игры Treasure Rush с графическим интерфейсом на мобильное устройство
2. Срок сдачи законченного проекта: 31 мая
3. Исходные данные к проекту: требования к реализуемому проекту
4. Содержание пояснительной записки: введение с описанием игры, основная часть с описанием особенностей разработки, описание автоматических тестов для бизнес-логики, заключение, список используемых источников.

Дата получения задания: «31» марта 2023 г

Руководитель

Степанов Д. С.

Задание принял к исполнению

Иванов Т. В.

31 марта 2023 г

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
ОПИСАНИЕ ПРЕДЛОЖЕННОГО РЕШЕНИЯ.....	5
ТЕСТИРОВАНИЕ ПРОГРАММЫ.....	6
ЗАКЛЮЧЕНИЕ.....	7
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ.....	8

ВВЕДЕНИЕ

Цель работы: создание игры с графическим интерфейсом на мобильное устройство

Правила игры:

В игре Treasure Rush вы играете за человека, которому нужно пройти лабиринт, обходя противников и умело используя окружающую обстановку.

- В начале игры вы появляетесь в первой комнате - ваша задача добраться до ее выхода. Перемещение осуществляется “свайпами” по экрану в направление, куда вы хотите чтобы персонаж переместился.
- После свайпа герой не остановится, пока не коснется любого препятствия на своём пути. На карте расположены враги, перемещающиеся по локации.
- При столкновении игрока и монстра вы потеряете одну жизнь, когда количество жизней упадет до нуля - игра закончится.
- Порядок расположения комнат и направление выхода из них при каждом прохождении генерируется случайно.
- Во время прохождения лабиринта слева сверху будут отображаться очки игрока, после удачного прохождения очки автоматически запишутся в топ всех очков, который будет обновляться после каждой игры.

ОПИСАНИЕ ПРЕДЛОЖЕННОГО РЕШЕНИЯ

Для создания мобильного приложения я использовал среду разработки Android Studio с функцией тестирования приложения на эмуляторе. Программа была написана с использованием концепции MVC (modelview-controller) для отделения бизнес-логики от визуализации, поэтому весь код можно разделить на 3 части.

Представление - графический интерфейс программы был выполнен в нескольких java классах: LoaderAssets, где происходит загрузка отдельных обрезанных изображений в “спрайты”, которые легче использовать в дальнейшем в коде, GraphicsFw - класс, содержащий в себе набор основных методов, которые могут понадобиться при создании GUI (таких как clearScene, drawLine, drawText, drawTexture, newTexture), благодаря встроенной библиотеке android.graphics*, класс позволяет непосредственно рисовать на экране то, что мы захотим и, наконец, класс AnimationFw, позволяющий создавать анимацию, соответствующего спрайта.

Контроллер - взаимодействие пользователя с программой осуществляется через класс TouchListenerFw, методы которого регистрируют простые жесты со стороны пользователя, такие как нажатие на экран устройства и отжатие, что позволяет использовать их в любых частях программы, где нам необходимо получить обратную связь от игрока.

Модель - классы GameManager, CollisionDetect и др. предоставляют данные и реагирует на команды контроллера, изменяя свое состояние.

В соответствии с выбранным шаблоном разработки, пользователь взаимодействует с view, все команды от пользователя обрабатывает controller, который в свою очередь обращается к model и, если это необходимо, перерисовывает view

ТЕСТИРОВАНИЕ ПРОГРАММЫ

Для тестирования бизнес-логики было написано 5 автоматических тестов, использующих возможности библиотеки JUnit

Первый тест проверяет правильность работы коллизии - если 2 объекта не имеют никаких пересечений, то коллизии происходить не будет, второй тест симулирует столкновение, путем пересечения “хитбоксов” между двумя объектами, проверяя как на это отреагирует класс. Третий тест проверяет правильность считывания нажатия на экран, а четвертый корректность считывания отжатия пальца от экрана. Пятый тест позволяет проверить правильность считывания свайпа программой.

ЗАКЛЮЧЕНИЕ

Было создано приложение с графическим интерфейсом на мобильное устройство. Были разработаны автоматические тесты, фиксирующие корректное и некорректное считывание данных программой и их обработка. В ходе выполнения этого задания мною был изучен шаблон MVC и особенности создания приложения на мобильные устройства.

Исходные файлы приложения находятся в репозитории на GitHub:

<https://github.com/TimoshaPonchik/polyTask3>

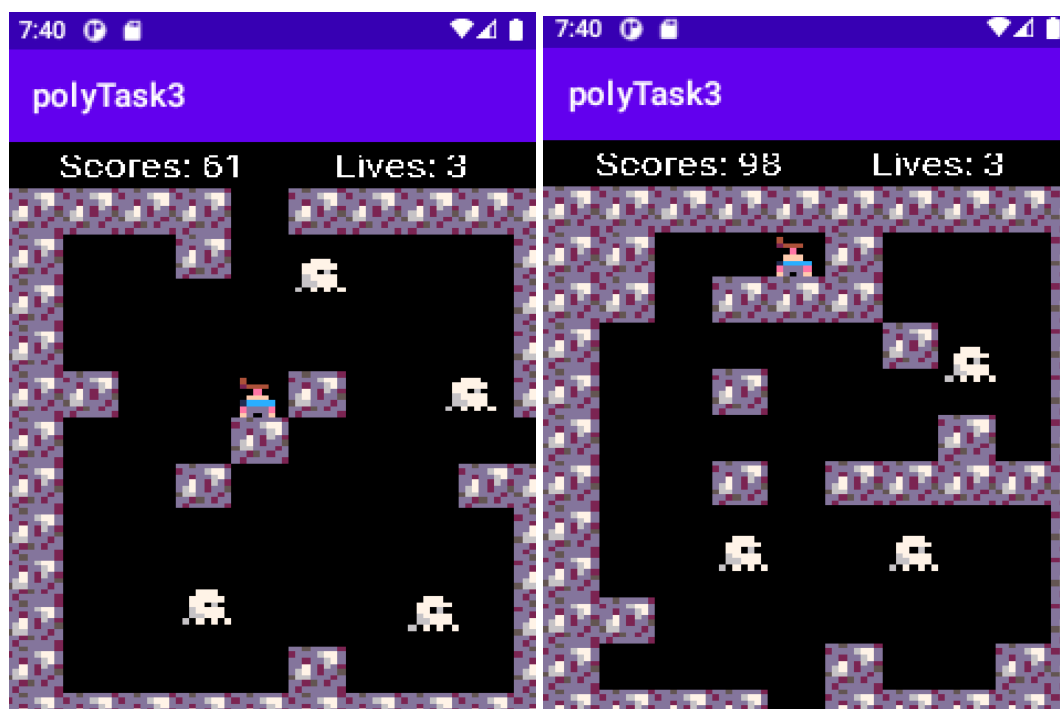


Рис 1. Скриншоты приложения

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. <https://developer.android.com/reference/android/graphics/package-summary> - описание библиотеки графики
2. <https://site.mockito.org/> - использовался для теста
3. <https://opengameart.org/textures/all> - текстуры для игры