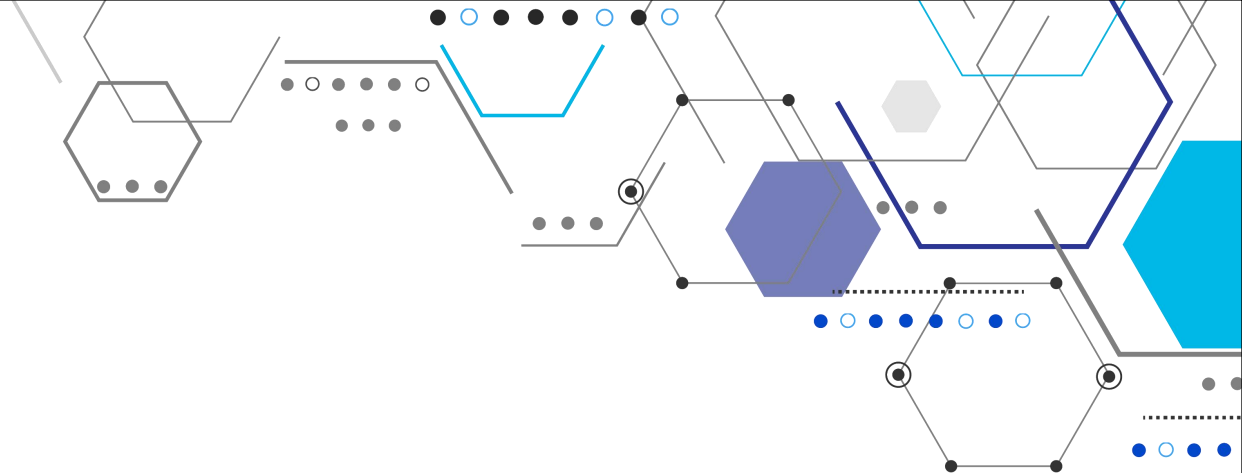
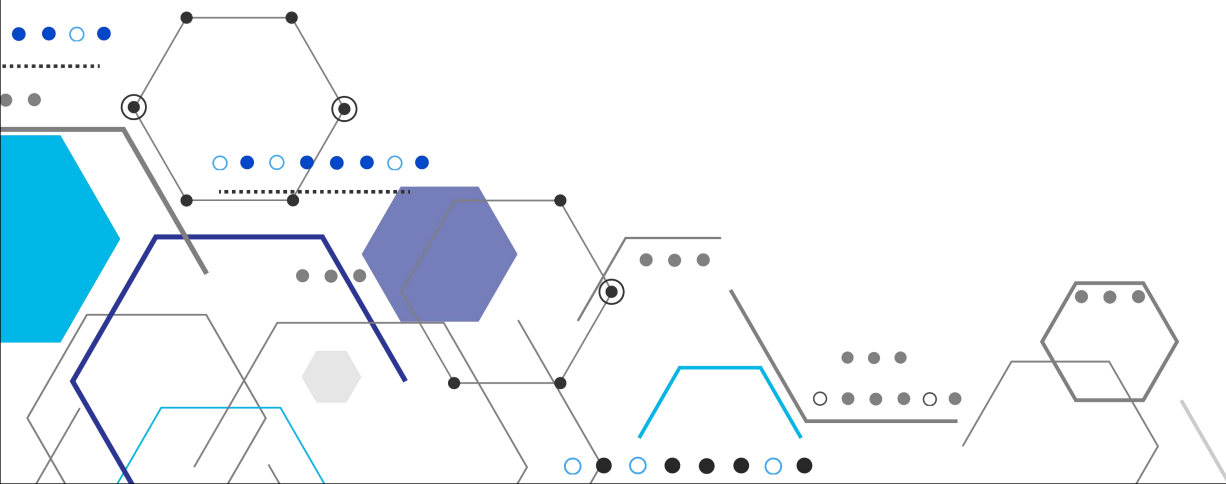




IT Школа





Bombardiro Astronauto

Игра о космическом корабле

Выполнил: Чечин К.

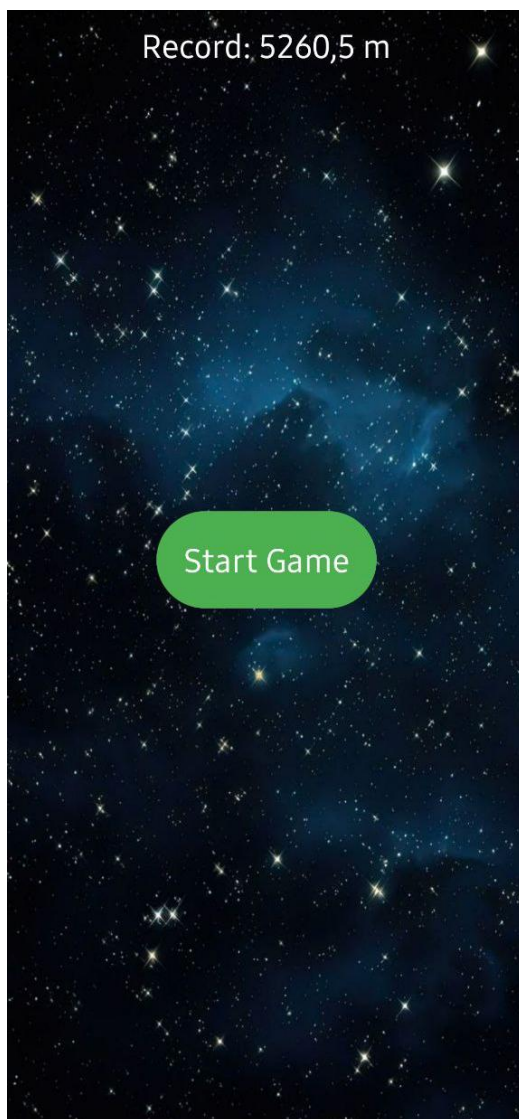
Научный руководитель: Нужных Алексей Васильевич

- Заккрытие спроса на 2D игры подобного формата
- Развитие скорости реакции и бдительности игрока
- Доставка визуального наслаждения пользователю

Преимущества приложения

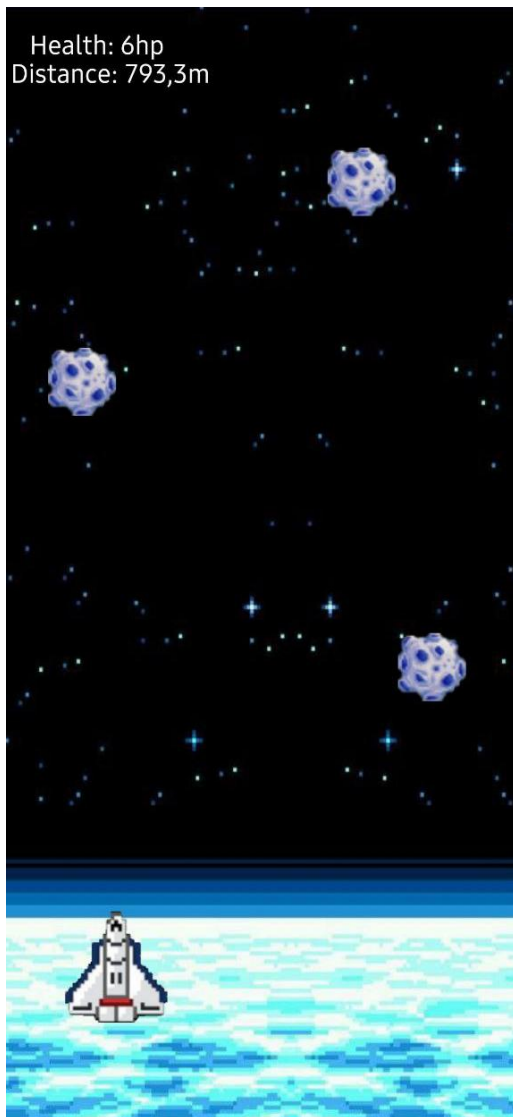
- Интуитивно понятный интерфейс
- Захватывающий геймплей
- Красивый 2D дизайн

Стартовый экран

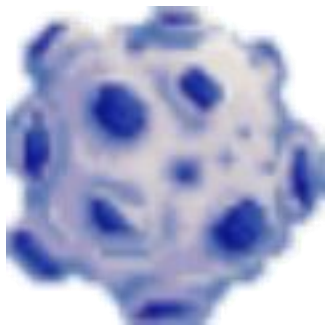
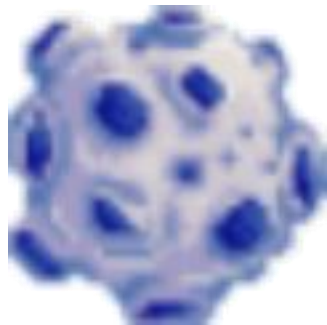
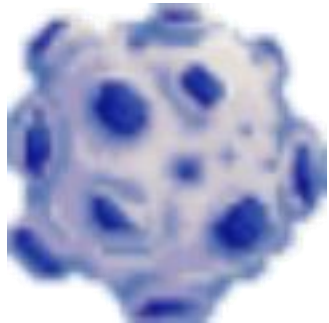


- Лаконичен и интуитивно понятен
- Присутствует отображение результата лучшего полета
- Обработка нажатия кнопки осуществляется с помощью метода `setOnClickListener()`

Немного о геймплее



- Игрок управляет космическим кораблем
- Цель - пройти как можно большее расстояние
- Задача - уворачиваться от камней
- Передвижение объектов происходит с помощью вызова метода `update()` несколько раз в секунду



- При попадании камня в игрока у него забирается 1 единица здоровья (1 hp)
- Механика их появления весьма интересна: случайная координата вдоль оси X и отрицательное значение Y



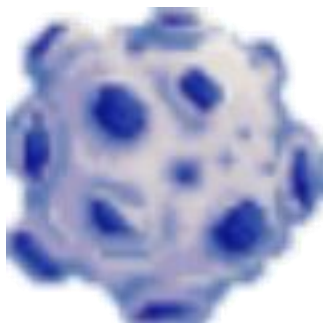
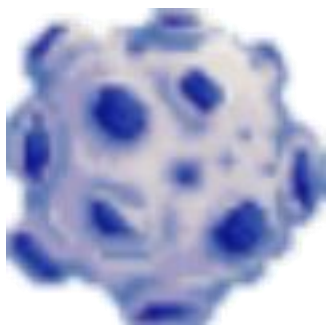
- Корабль игрока является космическим шаттлом, бороздящим просторы вселенной
- Управление происходит путем считывания нажатий правее или левее корабля с помощью MotionEvent

Distance: 793,3m

- Отображает в метрах расстояние, пройденное игроком
- Присутствует слева вверху во время игры
- Вычисляется в зависимости от текущего времени полета игрока

Health: 6hp

- Отображает количество очков здоровья, оставшихся у игрока
- Начальное количество hp - 10
- Также, как и счетчик расстояния, находится в верхнем левом углу



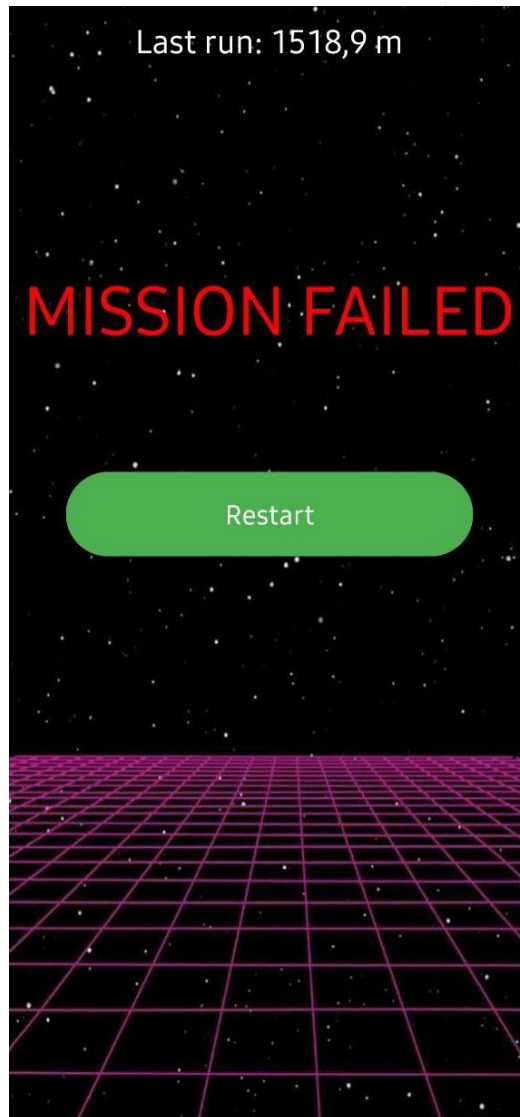
- Столкновения происходят по типу «квадрат с квадратом»
- При столкновении метеорита с кораблем вызывается метод `reset()`, возвращающий метеорит в исходное положение

Програмная реализация столкновений

```
// Проверка столкновения с игроком
if (((balls[i].getX() > player.getX() - balls[i].getWidth()) &&
    (balls[i].getX() < player.getX() + player.getFrameWidth())) &&
    (balls[i].getY() + balls[i].getHeight() >= player.getY())) {
    balls[i].reset(viewWidth, balls, i);
    health--;
}
```

- Условие проверяется постоянно для каждого члена массива метеоритов balls[]
- Условие в if() - проверка, заходит ли область метеорита на область корабля

Экран проигрыша



- Появляется когда здоровье игрока падает до 0
- Отражает длину только что проигранного залета
- Кнопка «Restart» возвращает на стартовый экран

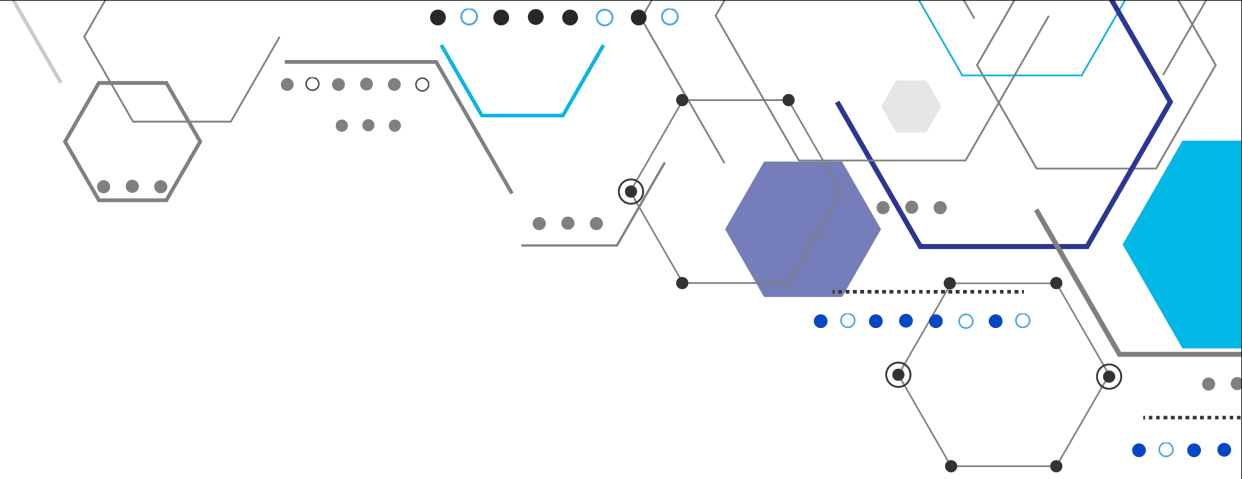
A black rectangular banner with a starry space background. The text "Record: 5260,5 m" is written in a white, sans-serif font. A bright star with a four-pointed diffraction pattern is visible on the right side of the banner.

Record: 5260,5 m

- Осуществляется с помощью приватной переменной `recordDistance` класса `GameView`
- Остается неизменной при перезапуске приложения

- Добавление глобального рекорда среди всех игроков игры
- Появление различных карт(дизайнов)
- Добавление различных уровней сложности

В этой игре была применена лишь немногая часть знаний, полученных мною на курсе Android разработки. К ним относится представление о устройстве приложения, его разметке, работе с графикой и такими классами, как `Timer`, `View`, `Bitmap` и др. Также я хочу выразить благодарность моему преподавателю Нужных Алексею Васильевичу за огромный вклад в мое образование и помощь в разработке этого проекта. Спасибо за внимание



SAMSUNG

