Space Arcade

Android course project

Main idea

- This game is an arcade. You have a spaceship which is going through the space endlessly. Your goal is to hit asteroids with your laser and avoid them.

Key features

- You can play endlessly
- Your score increases with each destroyed asteroid by 50 points and decreases by 1 point if you let asteroid go out of the screen
- You have 3 lives
- You can control your spaceship with virtual joystick or gamepad
- Sound effects and music in background
- We have about 50-60 frames per second in our game

Game parts

- Base structure of the project (Andrey Chernyshov)
- Input controllers (Alexey Katsman)
- Creating and optimizing drawing (Alexey Katsman)
- Collision detection (Alexey Katsman)
- Particle system (Alexey Katsman)
- Sound effects and music (Andrey Chernyshov)
- Styles of the whole game (Andrey Chernyshov)
- Animating dialogs and buttons (Andrey Chernyshov)

Game parts

- Base structure of the project (Andrey Chernyshov)
- Input controllers (Alexey Katsman)
- Creating and optimizing drawing (Alexey Katsman)
- Collision detection (Alexey Katsman)
- Particle system (Alexey Katsman)
- Sound effects and music (Andrey Chernyshov)
- Styles of the whole game (Andrey Chernyshov)
- Animating dialogs and buttons (Andrey Chernyshov)

Base structure of the project

What was done:

- Created base activity and fragments as base classes for the game
- Created some kind of game engine to control update and draw threads
- Proper entering the fullscreen mode was implemented
- Handling the back button depending on the current state

Game parts

- Base structure of the project (Andrey Chernyshov)
- Input controllers (Alexey Katsman)
- Creating and optimizing drawing (Alexey Katsman)
- Collision detection (Alexey Katsman)
- Particle system (Alexey Katsman)
- Sound effects and music (Andrey Chernyshov)
- Styles of the whole game (Andrey Chernyshov)
- Animating dialogs and buttons (Andrey Chernyshov)

Input controllers

What was done:

- Chosen a set of images for spaceship, bullets and so on
- Created input controllers for virtual joystick and gamepad
- Screen was split vertically for virtual joystick: left part is for movements and right one is for firing
- Implemented logic of player movements and bullet firing
- Implemented arrow buttons controller and virtual joystick and the latter one was chosen
- Perfect time between shots was chosen along with speed of spaceship and fired bullets
- Virtual joystick and gamepad support was put together in composite input controller

Game parts

- Base structure of the project (Andrey Chernyshov)
- Input controllers (Alexey Katsman)
- Creating and optimizing drawing (Alexey Katsman)
- Collision detection (Alexey Katsman)
- Particle system (Alexey Katsman)
- Sound effects and music (Andrey Chernyshov)
- Styles of the whole game (Andrey Chernyshov)
- Animating dialogs and buttons (Andrey Chernyshov)

Creating and optimizing drawing

What was done:

- Implemented two game views with and without surface and the first one was chosen as more efficient
- Draw thread was improved by removing timers and adding thread sleeping expecting about 50 fps
- Sprite logic added for all game objects
- FPS counter was added to the left bottom of the screen
- Asteroids were added as new game objects
- Implemented logic for asteroids - random generation of starting x position on top of the screen and angle
- Transformation matrix was used for all translations and rotations
- Parallax background was added
- Layers support was added and used for all game objects including background

Game parts

- Base structure of the project (Andrey Chernyshov)
- Input controllers (Alexey Katsman)
- Creating and optimizing drawing (Alexey Katsman)
- Collision detection (Alexey Katsman)
- Particle system (Alexey Katsman)
- Sound effects and music (Andrey Chernyshov)
- Styles of the whole game (Andrey Chernyshov)
- Animating dialogs and buttons (Andrey Chernyshov)

Collision detection

What was done:

- Tried rectangle and circle collision detection approaches, mix of both was chosen as optimal
- Asteroid and spaceship is treated as circle object while bullets are treated as rectangles
- Implemented logic for detecting collisions of any type
- Implemented quad-tree for detecting collisions so we could only do collision checks for objects which are close to each other
- Problem with duplicated collisions was solved inside quad-tree logic

Game parts

- Base structure of the project (Andrey Chernyshov)
- Input controllers (Alexey Katsman)
- Creating and optimizing drawing (Alexey Katsman)
- Collision detection (Alexey Katsman)
- Particle system (Alexey Katsman)
- Sound effects and music (Andrey Chernyshov)
- Styles of the whole game (Andrey Chernyshov)
- Animating dialogs and buttons (Andrey Chernyshov)

Particle system

What was done:

- Initializers for particles were implemented which give initial speed, direction and rotation for them
- Modifiers for particles were implemented for smooth fading out
- So called one shot particles (such as explodes) and emitting particles (such as spaceship smoke and asteroid trail) support was implemented

Game parts

- Base structure of the project (Andrey Chernyshov)
- Input controllers (Alexey Katsman)
- Creating and optimizing drawing (Alexey Katsman)
- Collision detection (Alexey Katsman)
- Particle system (Alexey Katsman)
- Sound effects and music (Andrey Chernyshov)
- Styles of the whole game (Andrey Chernyshov)
- Animating dialogs and buttons (Andrey Chernyshov)

Sound effects and music

What was done:

- Some basic sound effects were created for asteroid and spaceship explosion and bullet firing
- Created sound manager for working with sounds and music
- Sound effects were properly integrated to sound manager and are used on proper game events
- Music for background was added
- Ability for enabling and disabling sound effects and music using buttons in main menu and pause dialog was implemented

Game parts

- Base structure of the project (Andrey Chernyshov)
- Input controllers (Alexey Katsman)
- Creating and optimizing drawing (Alexey Katsman)
- Collision detection (Alexey Katsman)
- Particle system (Alexey Katsman)
- Sound effects and music (Andrey Chernyshov)
- Styles of the whole game (Andrey Chernyshov)
- Animating dialogs and buttons (Andrey Chernyshov)

Styles of the whole game

What was done:

- Custom font in retro style was added and applied throughout the game
- Background for main menu was added
- Buttons with text were replaced with image buttons such as pause, resume, restart, exit, sound on/off, music on/off and implemented beautiful states for them
- Score on the left side and pause button on the right side were separated from the game area
- The score and lives counter were implemented
- Beautiful dialogs were added (pause, game over and exit)

Game parts

- Base structure of the project (Andrey Chernyshov)
- Input controllers (Alexey Katsman)
- Creating and optimizing drawing (Alexey Katsman)
- Collision detection (Alexey Katsman)
- Particle system (Alexey Katsman)
- Sound effects and music (Andrey Chernyshov)
- Styles of the whole game (Andrey Chernyshov)
- Animating dialogs and buttons (Andrey Chernyshov)

Animating dialogs and buttons

What was done:

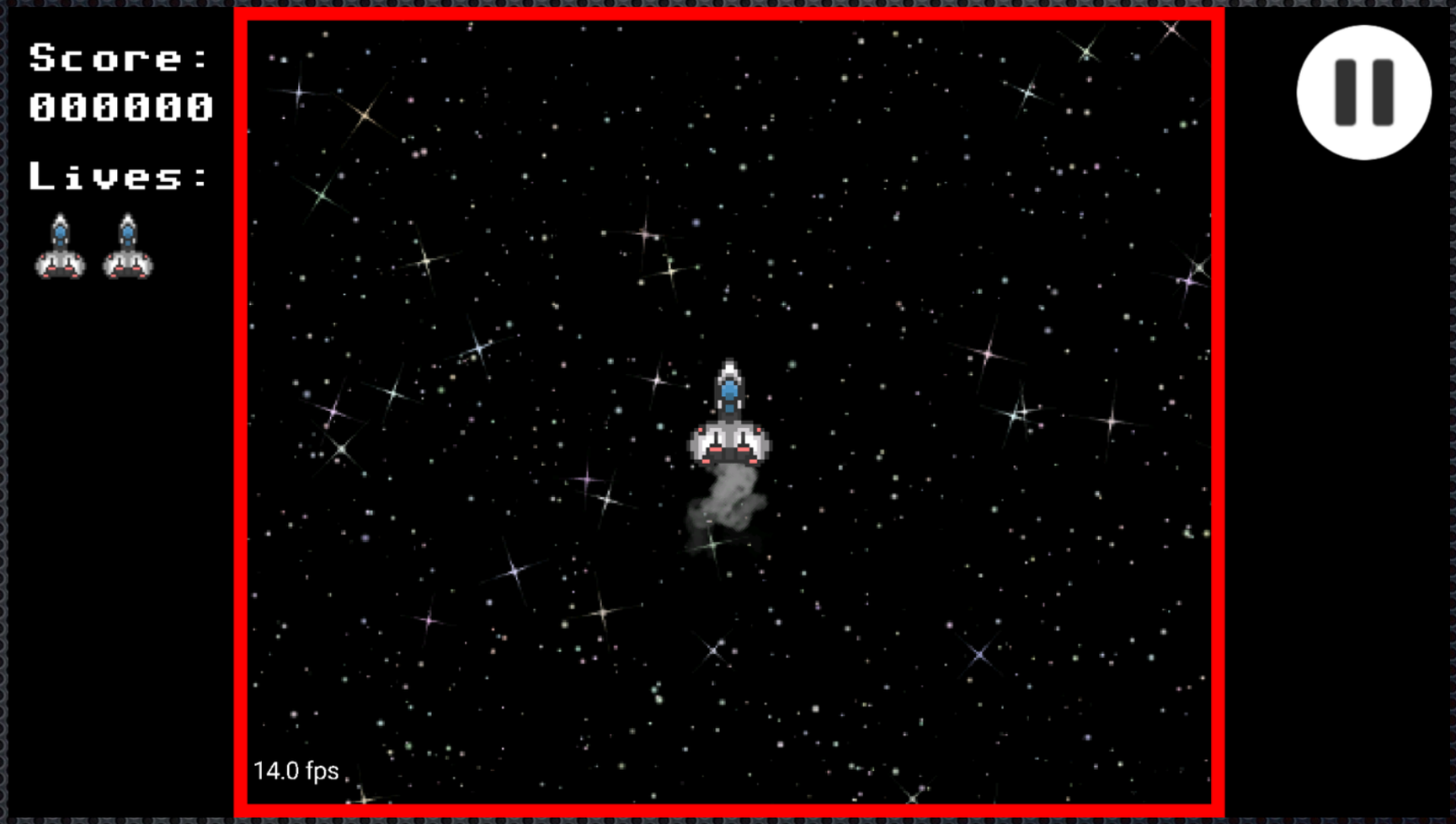
- Spaceship animation using left and right lights was implemented
- Beautiful dialog animation through the top of the screen was added
- Proper buttons handling was implemented in respect that some time is needed for dialog to go away back through the top of the screen
- Added pulsating animation for start button in main menu
- Added animation for game title and subtitle in main menu

Conclusion (some screenshots)

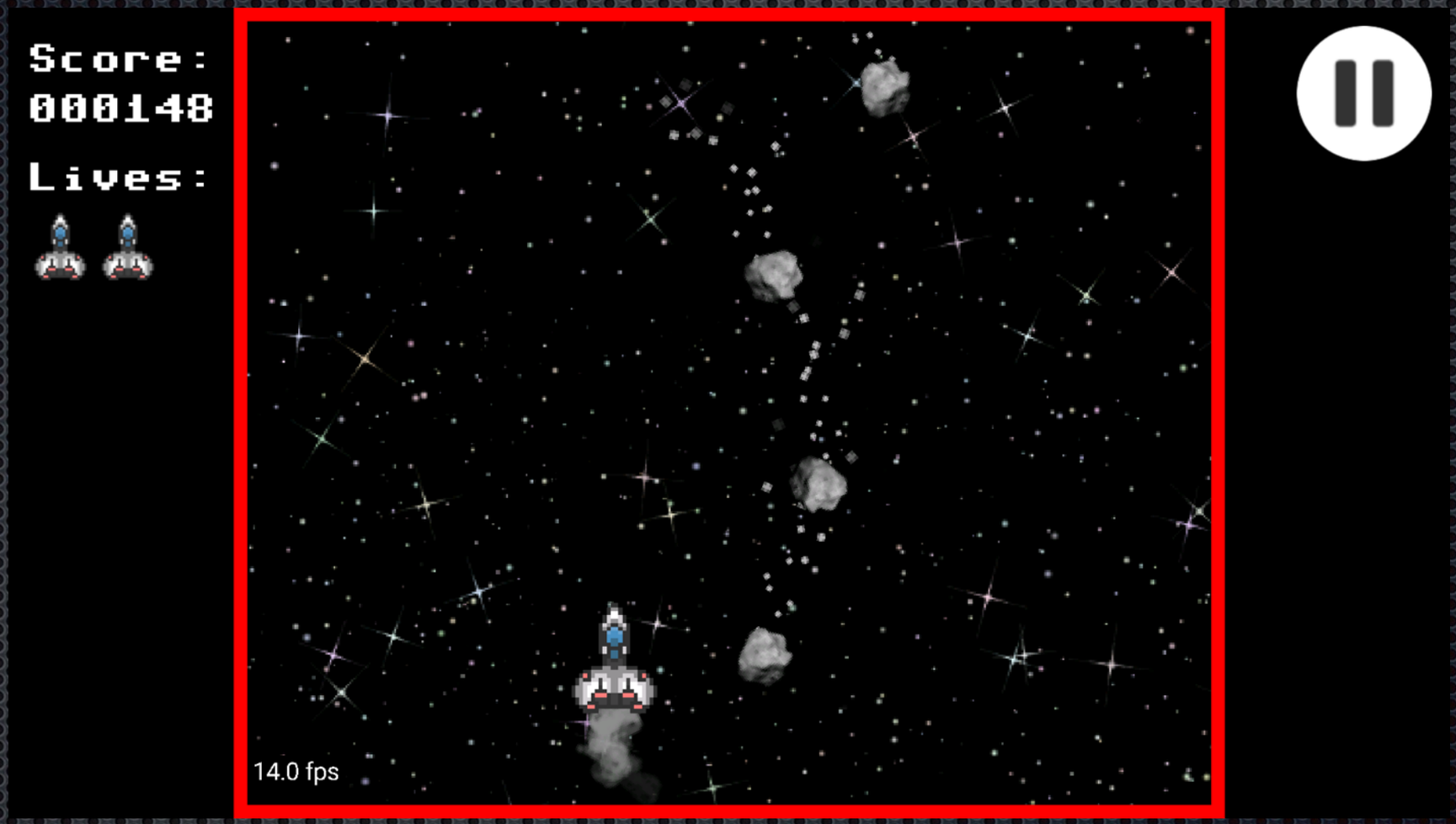
Conclusion (some screenshots)

Conclusion (some screenshots)

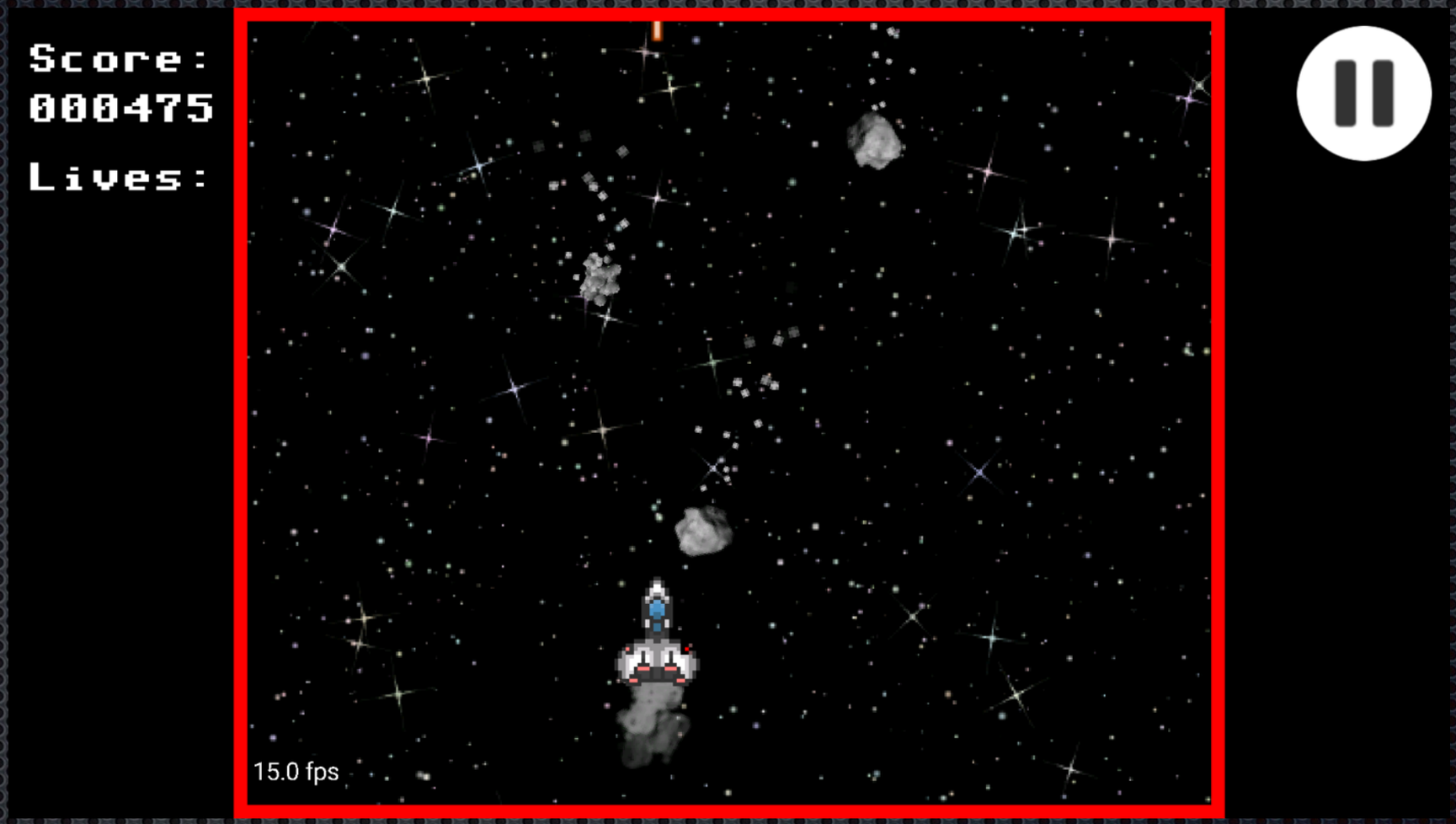
Conclusion (some screenshots)



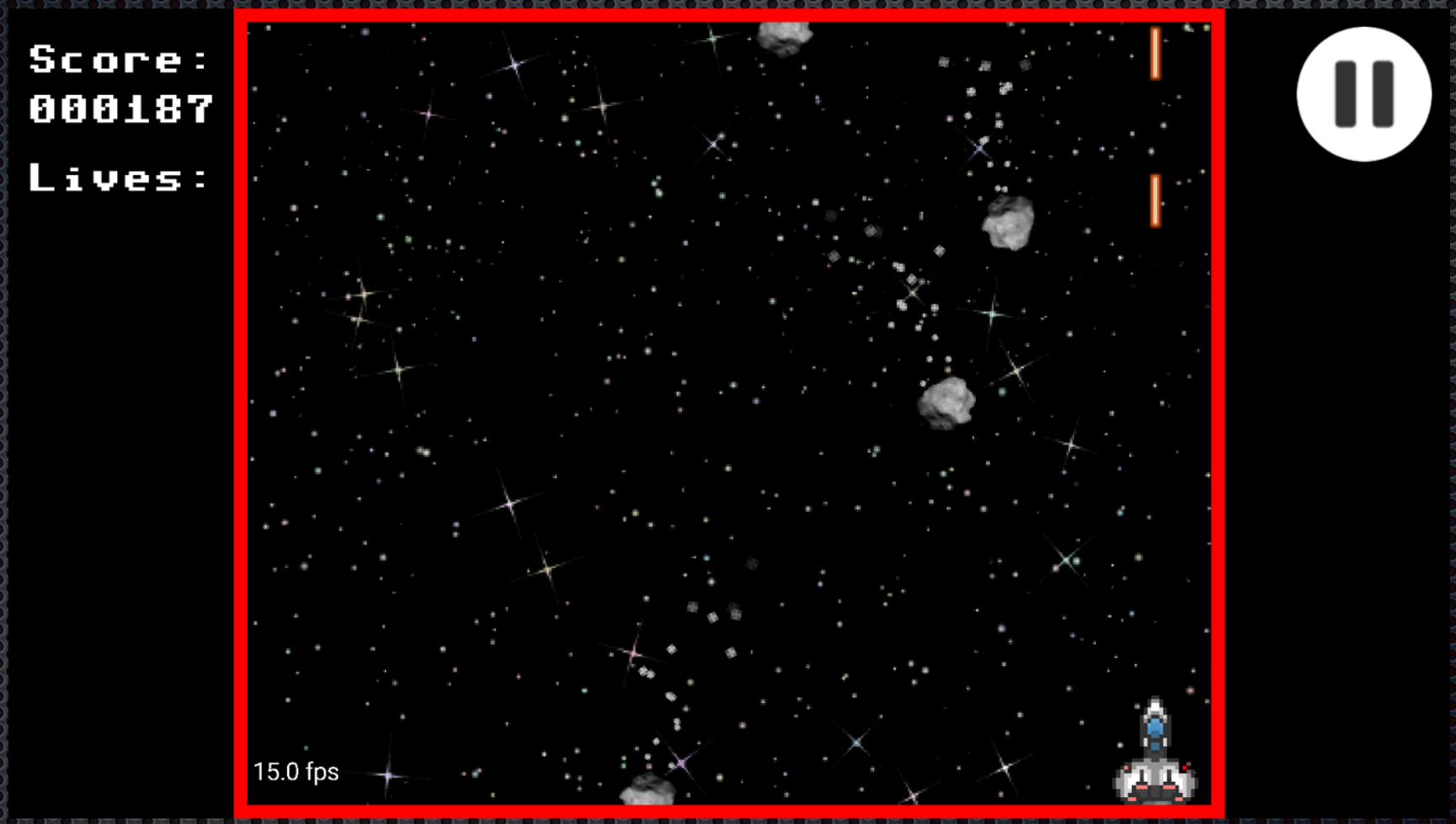
Conclusion (some screenshots)



Conclusion (some screenshots)



Conclusion (some screenshots)



Conclusion (some screenshots)

Conclusion (some screenshots)

Thank you for your attention

Source code:

https://github.com/alexkats/AndroidCourseProject

Any questions?

Developers: Alexey Katsman and Andrey Chernyshov