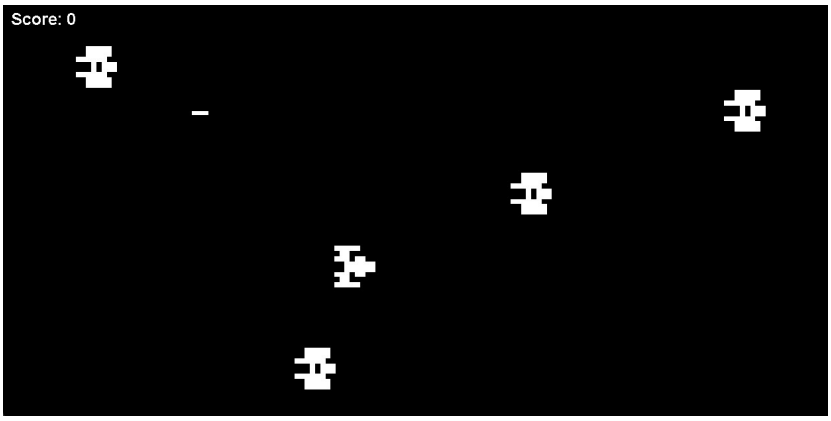
Space Encounter

The goal of this project was to make a fun, yet very simple arcade shooter style game.

The final game is a **real-time 2D arcade shooter** where the player controls a spaceship that can:

- Move using the W/A/S/D keys.
- Boost speed by holding the Shift key (becoming invincible).
- Shoot dual bullets with the Space key (unless boosting).
- Avoid or destroy enemy bullets to survive.
- Earn points by destroying enemies.
- Restart the game with **R** after a Game Over.
- Quit using **Q**.

Screenshot:



System Requirements:

- Java version: Java 17 or higher
- Development Tools: IntelliJ IDEA / Eclipse / any Java IDE
- Assets: PNG sprites for ship, enemies, etc. stored in /src/Sprites/

Packages:

Game

- **Game**: Main loop and game panel, handles input, rendering, and updates.
- **GameContext**: Singleton that manages the current game state.
- **GameModel**: Stores shared game data (player position, bullets, enemies, score, etc.)

State

- **GameState (interface)**: Abstraction for different game states.
- **PlayingState**: Main game logic and input handling.
- **PausedState**: pause feature.

Entity

- **Bullet**: Represents bullets, tracks position and movement.
- **Enemy**: Represents enemies and their bullets.
- **EnemyFactory**: Randomly spawns new enemy instances.

Communication

- GameModel acts as shared data across game states.
- GameContext controls the current state and calls its methods (update(), render(), handleInput()).
- Input events are passed from Game to the current GameState.

Manual Testing Steps:

1. **Movement Test**: Use W/A/S/D to move around the screen.
2. **Boost Mechanic Test**: Hold Shift — the ship should move faster and become invincible (fades visually).
3. **Bullet Test**: Press Space — two bullets should fire unless boosting.
4. **Enemy Spawning**: Let the game run — enemies should appear and shoot bullets.
5. **Collision Detection**:
    - Try to collide with enemy bullets while **not boosting** — should trigger Game Over.
    - Try to collide while boosting — no Game Over should occur.
6. **Restart/Quit Test**: After Game Over, press **R** to restart or **Q** to quit.

These tests verify movement, shooting, enemy behavior, game state transitions, and input responsiveness.

Controls:

Key	Action
WSAD	Movement
Shift	Boost (faster, invincible, semi-transparent)
Space	Shoot bullets (only when not boosting)
R	Restart the game (after Game Over)
Q	Quit the game (after Game Over)

Challanges:

- Input Handling: Initially, the Shift mechanic broke the Restart and Quit keys due to the structure of input delegation. This was fixed by ensuring input was passed to the current game state.

Conslusion:

This project reminded me that making your own games is much more achievable than it seems to be.