



Space Invaders - C++ Version

03.30.2024

Overview

This project is focused on developing a GUI-based adaptation of the classic 1978 Arcade game Space Invaders using the memory management capabilities and efficiency of C++ , by designing a robust and lightweight terminal that simulates the intricate mechanics of resource management, survival, and decision-making, effectively capturing the essence of the original game.

Features

1. Classic Gameplay
2. Custom Assets
3. Responsiveness
4. Leaderboard (maybe)
5. Thorough Testing

Stack

C++

Option 1(Preferred): OpenGL

Option 2: SFML

Libraries

GLFW + GLEW

Why C++?

C++ is regularly used in graphics contexts due to its efficacy in speeding up the rendering process. I plan to use it to produce a high frame rate GUI with responsive gameplay.

This version, like the original, features no tutorial.

Structure

- **main.cpp**: Runs the game,
- **game**: Manages *game state* (running, end), creates entities, manages game loop, imports ux elements, defaults input to input class.
- **entity.hpp**: Base class for renderable game components. Fields: position, velocity, hp, sprite, projectileRate, weak boolean (dies on 1 collision), etc.
 - ◆ **player.cpp**: Controllable via input, loses health on collision with enemy projectile
 - ◆ **enemy.cpp**: static behavior, periodically creates projectiles, collision (with player projectile) affects health
 - ◆ **projectile.cpp**: weak, non-controllable.
 - ◆ **barrier.cpp**: can be destroyed, static (speed = 0), non-controllable.
- **gui.cpp**: manages gui elements to be printed in menu and game, hp indicator, score, etc.
- **input**: no mouse control, gets key pressed from keyboard and sends to game for status update: start game, quit, control player, shoot.
- **resources**: wrapper for sprites
- **audio.hpp (maybe)**: wrapper for audio, projectile goes pew pew

* when extension not given: hpp+cpp

Work

This is a single developer project for my game dev portfolio. If someone wants to collaborate, sure.

Milestones

I. Basic Demo (04/17)

[Playable Game, no UI only entities, no Menu, Player Controls and Enemy Movement, 1HP enemy, 3HP player]

II. Polishing & Testing (04/24)

[All testing done, final sprites]

III. Full Game Functionality (05/01)

[Menu to Play and Quit, winning loops game, enemy with more hp (like original)]

IV. Stretch

[Scores(I/O) with 3 Letter Input, sound design, physics (maybe)]

Testing

Extensive manual testing will be performed.

Tests will be written for non-GUI classes to make sure they function correctly.