Space Invaders

Authors:

_____Yaseen Alam__________K23009500______BSc Artificial Intelligence

_____Aditya Ranjan_________K23149795______BSc Artificial Intelligence

_____Kasim Morsel_________K24060083______BSc Artificial Intelligence

_____Yusuf Rahman_________K22040245______BSc Artificial Intelligence

Introduction:

Purpose:

This project aims to recreate the classic arcade game "Space Invaders" using elements of JavaFX. The objective was to design a functional, interactive and visually styled 2D game where the player defends the Earth by shooting down waves of alien invaders.

Context:

Space Invaders is a foundational title in the history of video games, emphasising strategic shooting, player reaction time and wave progression. Our version mimics the core experience, using modern JavaFX elements for rendering, layout and input.

As we all had a personal connection with shooter games, this was a choice that we felt passionate about working on, making us always find creative ideas to extend the game.

Objectives:

- Develop a professional version of Space Invaders with fundamental roots alongside some touches of personalisation to achieve a version we were proud of.
- Create the felling of rush as the user wanting to resist from dying through different methods, this is to keep the user hooked onto the game and make them want to reach a high score.
- Demonstrate and use OOP fundamentals for exploration and experience in professional and high-quality code through the usage of GitHub and in a team environment where responsibilities lie on each member.

Game Description:

Rules and Mechanics:

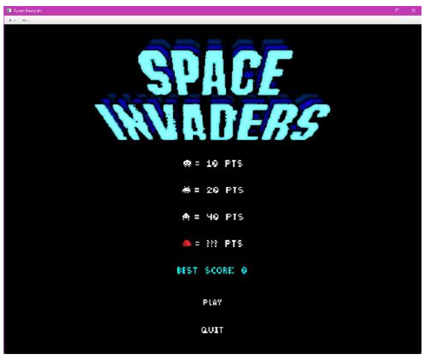
- The player controls a spaceship that moves left and right across the screen using arrow keys.
- Pressing the spacebar shoots a laser upward.
- Aliens move in formations and descend as they bounce off screen edges.
- Aliens shoot lasers downward at the player.
- Barriers can absorb hits from alien or player lasers.
- Scoreboard to keep track of points gained by the player.
- The player loses when all lives are lost or aliens reach the bottom.

Scoring System:

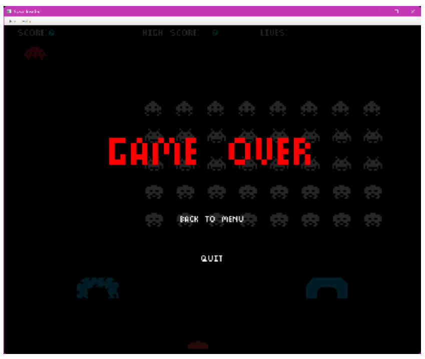
- Different alien types give different number of points:
  - Small Alien: 10pts
  - Medium Alien: 20pts
  - Large Alien: 40pts
  - Super Alien: random amount from 10-500
- 500 points for completing every round, that increase linearly as more rounds are completed.
  - A round is defined as eliminating a full wave of aliens.

User Interface:

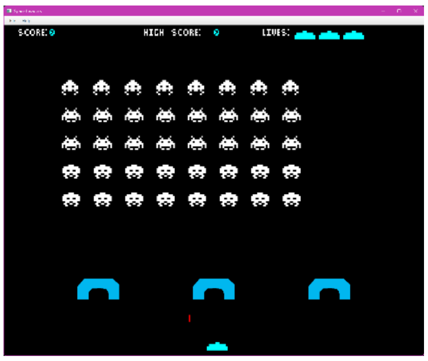
Menu:



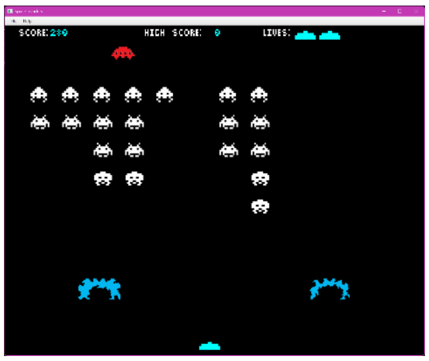
Game Over:



Game (start):

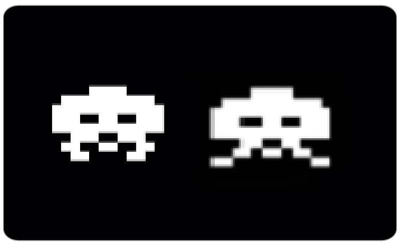


Game (in progress):

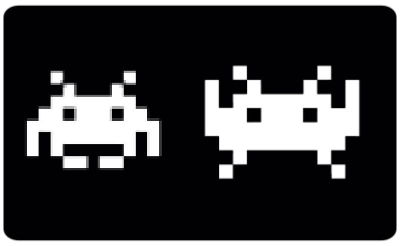


Animations:

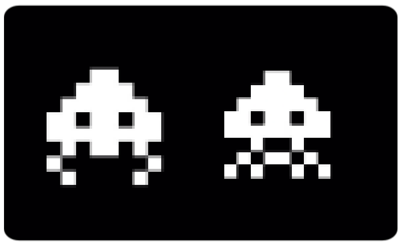
Small Alien:



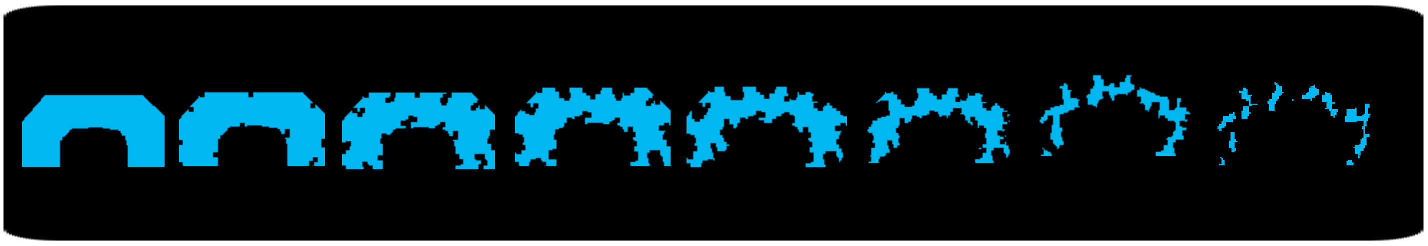
Medium Alien:



Large Alien:



Barrier:



Laser:

- Red pellet-like rectangle that moves upwards or downwards.

Conclusion:

Future Improvements:

- More levels:
  - Adding more levels to increase difficulty, this includes from similar swarms with higher health to having bosses with creative functions that require more creative solutions to beat with custom barriers (maps) that serve different function in fights.
- Power-ups:
  - Maybe killing some special aliens give like a power boost, like:
    - Rapid shooting
    - Auto shoot (so no need of tapping)
    - More than one gun, so more lasers releasing from player
    - Moving faster to dodge lasers (more useful for bosses)
    - Immunity/shielding around player so they get extra lives for a short amount of time
    - Faster lasers, rockets and etc. for more explosive attacks
  - Picking up power ups by moving to it on time to get any help or lives
- Add different spaceship designs players can acquire to customise experience, as well as laser colours
  - Maybe build your own spaceship function

- More optimisation to get minimal lag as possible as at the moment it is not the most optimised but in a good playable state
- Dynamic High Score update and using file input and output to store highest scores and display them.
- Adding a Leaderboard to increase competition between players.

Summary:

This project reinforced the value of leveraging collaborative tools such as GitHub for coherent teamwork. Using GitHub allowed us to manage version control effectively, coordinate our changes, and merge our work seamlessly within our pair. We learned that following proper coding etiquette such as organising code into clearly defined methods and classes, using descriptive variable names, and employing abstract classes to define shared behaviour leads to cleaner, more maintainable code.