

GR Interview Task: Space Invaders

Overview

This project comprises of a simple variation of the famous Space Invaders arcade game.

The game has been partially implemented, and a lot of functionality is still missing, your mission is to add as much functionality from the requested task list as is feasible in the time allocated.

Tasks are listed in no particular order, and it is recommended you pick one or two to focus on that you think are achievable. If you find you have time remaining then by all means attempt more.

Don't be afraid to ask questions, communication is important in development and will be encouraged rather than penalised.

The 'hint' sections are there to provide direction, but if you find you have a solution that does not require following these hints, go for it, there are many ways to complete every task.

Tasks

- **Identify areas of common code and create Abstract classes or Interfaces where appropriate.**

Some of the code has a lot of potential for abstraction to save code duplication and enhance the maintainability of the game. Identify these areas and create abstract classes and/or interfaces where necessary.

Hint: The bullet classes may be a good place to start.

- **Increase the complexity of the scoring system to score higher the quicker enemies are defeated.**

Currently there is a flat, 1 enemy, 1 point scoring system. Increase the complexity of this system to award higher point values the sooner the enemy is defeated.

Hint: The controller should handle the amount of points awarded based on its own internal calculations.

- **Display a Game Over message**

Although the game is aware when one of the game over states has been reached, this information is not conveyed to the user. Display an indication to the user that they have either lost or won.

Hint: Listen for the GameOver events on the bus to detect when a game is finished. This would be a nice opportunity for some animation!

- **Add a way to reset the game on death**

On player death, the user is currently forced to reopen the game if they wish to play again. Add a method of interaction (button, stage click or otherwise) whereby the user can 'reset' the game to its original state.

Hint: If you are NOT attempting the task below, also add a way to reset the game on winning, as the user is left in the same state.

- **Add levels of increasing difficulty**

Currently there are a set number of enemies that move and fire at a set time during the game. Add levels of difficulty so that once each level is completed (e.g. all enemies defeated) the next level has more and/or faster enemies. Shields and lives should be reset between levels.

Hint: This is likely to require some additional configuration to be added and parsed from the config, for example, 'NumEnemies' is no longer relevant. Consider adding config values for LevelEnemyNumIncrease, LevelEnemySpeedIncrease and MaxLevels.

- **Improve design by adding animations and/or redrawing game objects**

The game design is quite simple and could benefit from some 'bells and whistles'. Look at the skin file to see where improvements could be made via redesign.

- **Store score on gameover and display top 3 scores on each subsequent game over.**

The user has no sense of progression or improvement between game plays. Add a 'top 3' display to indicate to the user their three best scores since opening the game.

Hint: This requires you to have completed the task 'Add a way to reset the game on death'

- **Add a meter to display what percentage of shield you have left**

The players 'shields' are the green blocks that defend them from enemy attack. As they deteriorate, it would be useful to the user to see a display indicating their overall health, so they can adjust their tactics accordingly.

Hint: The shield 'health' meter could be drawn dynamically, or be a stage clip with a dynamically set height / width (depending on your choice of orientation)

- **Add a 'heat' meter to your gun, when it hits the limit disallow firing until recharged**

Players may find the game too easy when they can fire at will. Add a heat meter that charges up the more bullets you release, on hitting its maximum, firing should be disabled until it has discharged.

Hint: This is similar to a shield health meter, but with the addition of a timer to slowly diminish the meter until firing is enabled again. You may consider adding the time this takes as a config setting.

- **Improve collision detection on shields**

You may notice that shield blocks still remain in place despite appearing to have been hit by a bullet. This is due to collision detection on the shield not being accurate enough. Improve the collision detection to ensure shield blocks are always removed on impact.

Hint: Currently collision detection does not take into account the size of a bullet, or the corresponding shield block it hits.

- **Stop PlayerShip from leaving stage bounds**

PlayerShip movement is controlled via the mouse. At the moment this movement is not limited to the stage bounds and the PlayerShip can 'hide' off stage to the left or right. Correct this by stopping the PlayerShip from leaving the bounds of the stage.