

# Project Title

Cosmic Conquistadors

This is our attempt at replicating the Space Invaders game using the Standard Library as provided in the CSE214 course, and the guidelines set out in the project documentation.

## Getting started

Step 1 – Download the project folder zip from SunLearn onto the H drive and extract the main project folder onto the root of your H: drive.

Step 2 – Open the .bat file as provided in the DrJava folder on the H: drive to open the command line

Step 3 – Type in the following to go to the directory of the project: *cd CosmicConquistadors*

Step 4 – Compile the program by typing in the following: *javac Invaders.java*

Step 5 – Run the program by typing in the following: *java Invaders*

## Details on Inheritance and PolyMorphism

An example of interface inheritance is how DefaultCriticter extends the Critter interface to enable the DefaultCriticter to move around the canvas by specifying getters and setters for its radius, coordinates and velocities.

An example of class inheritance is how Shooter extends the DefaultCriticter class to create a new object that has functionality for a critter with a moveable barrel that that is able to fire missiles and collect powerups.

An example of polymorphism is how the BossEnemy is both a child of DefaultCriticter, Critter and Enemy at the same time. We utilize this aspect of polymorphism to put all the different types of enemies in one ArrayList and then to retrieve each enemy by using the InstanceOf keyword in java.

## Summary of Additional Work Done

- Interesting Game Elements:
  - The final boss moves around randomly on the screen and randomly fires missiles in the direction of either player one or player two to make the final boss fight even more challenging.
  - An ArrayList of small circles of random size and position between -800 and 800 were implemented to give the game that look of dispersion between the stars, these small objects were moved down the screen at a constant velocity. This same system was used to generate the powerUps appearing on the screen.
  - Missiles bounce once off side walls.

- A high score screen was implemented that shows the player that the game is over and also lists the previous high scores in descending order on the screen. Previous high scores are all stored inside a text file, and the program does all necessary error checking for this text file, and creating it if necessary.
- A png image was used, in unison with changing the size and not clearing the screen gave the image an impression of moving. This method was used for the game over and win screen, this illusion was also implemented in the main menu screen to flicker the different colours for the border.
- The game features different types of Enemies
  - A light enemy that shoots regularly compared to a heavy enemy and has relatively few hitpoints.
  - A heavy enemy that shoots less than a light enemy, but has more hitpoints
  - A boss enemy that moves around randomly on the screen, has more hitpoints than all other enemies, and that can shoot three missiles at once in the direction of the shooter.
  - Bunkers that are distributed evenly across a horizontal point, bunkers are advantageous to the enemies as they destroy powerups, enemy missiles can pass through them, and the shooter has to shoot them a certain amount of times before they get removed from the screen. If an enemy touches a bunker the bunkers gets destroyed and no damage is dealt to the enemy.
- The game features multiple types of power ups that are randomly generated and that move vertically down the screen, if the shooter collides with a power up he obtains its effect until the power up has expired:
  - Extra lives were implemented
  - A missile that has a faster velocity than the default missile, the user only has a certain amount of these.
  - A missile that is larger, and that does more damage than a default missile, the user only gets one of these per powerup.
  - The ability to shoot three missiles at once, each at a different angle, the player only has a certain amount of these.
  - The ability to shoot a certain amount of missiles without any cooldown period between the shooting, the player only has a certain amount of these.
- The game also features the ability for collaborative multiplayer when a player presses P, the second player features full functionality, from having extra lives to being able to get destroyed by enemy missiles.
- The enemies, bunkers and shooters all have hit points corresponding to the type of entity that the object is, different types of missiles remove different amount of hitpoints of each type of entity.
- The graphics of the game were improved using various png's created and designed by hand by a group member in photoshop, we also used a third party font to make our main screen look better.
- Different sounds for firing missiles, destroying objects and missile hits were manually created using the online tool beepbox.co, these self created sounds were then implemented into the game.

none

none

see ClassDiagram.png added into this folder for a more detailed view

