

Assignment 2 - Report

9/24/2023

MISSING 6 Possible Points

Attempt 1



In Progress

NEXT UP: Submit Assignment

Add Comment

Unlimited Attempts Allowed

9/5/2023 to 10/4/2023

▼ Details

Problem Overview

You are required to design an application model for a Space Invaders game. Space Invaders is a classic arcade game played on a rectangular screen with different types of aliens on it. The player controls a spaceship at the bottom of the screen and can shoot projectiles at the aliens. The aliens move horizontally and descend towards the player's spaceship. If the aliens reach the bottom of the screen, shoot the player, or collide with the player's spaceship, the game is lost. The player can move their spaceship horizontally to avoid the aliens' projectiles and shoot them down. The game is won when all the aliens are destroyed. The score is calculated based on the number of aliens destroyed. The duration of the game is clocked until all the aliens are destroyed.

Q: What is Space Invaders?

A: Some real-world examples could be found [here](https://www.youtube.com/watch?v=uGjgxwiemms)  (<https://www.youtube.com/watch?v=uGjgxwiemms>) and [here](https://en.wikipedia.org/wiki/Space_Invaders)  (https://en.wikipedia.org/wiki/Space_Invaders).

Assignment 2 Requirement

In assignment 2, you are going to implement your space invaders game and refactor your UML class diagram according to your code. You will need to ensure that your application is configurable with a JSON text file. You **must use the GoF design patterns** in your implementation as requested below:





- Projectiles must be created using the Factory method.
- The state/colour change of each bunker must be controlled using the State pattern.
- The enemies and bunkers must be created using the Builder pattern.
- The behaviour of enemy projectiles will be controlled using the Strategy pattern.

Please find the detailed tasks below:

Implementation Task

You will use the Java Programming Language to implement the UML class diagram you designed in assignment 1. In this assignment, you are now responsible for the implementation of the entire application code.

What we provide to you

- JSON file: an example JSON file format is provided to you [here](https://canvas.sydney.edu.au/courses/53385/files/33141454?wrap=1) (<https://canvas.sydney.edu.au/courses/53385/files/33141454?wrap=1>)  (https://canvas.sydney.edu.au/courses/53385/files/33141454/download?download_frd=1) that you can start with for your implementing. A sample configuration reader is also provided [here](https://canvas.sydney.edu.au/courses/53385/files/33141546?wrap=1) (<https://canvas.sydney.edu.au/courses/53385/files/33141546?wrap=1>)  (https://canvas.sydney.edu.au/courses/53385/files/33141546/download?download_frd=1) for your consideration.
- gradle file: a sample build.gradle file is provided to you [here](https://canvas.sydney.edu.au/courses/53385/files/32871967?wrap=1) (<https://canvas.sydney.edu.au/courses/53385/files/32871967?wrap=1>)  (https://canvas.sydney.edu.au/courses/53385/files/32871967/download?download_frd=1) .
- A [codebase](https://edstem.org/au/courses/12969/lessons/41891/slides/289079)  (<https://edstem.org/au/courses/12969/lessons/41891/slides/289079>) is provided to you here to help get you started. You are free to modify this as much or as little as you like.

What we expect from you

Your space invaders game is now expected to support the following features in your code:

[Submit Assignment](#)

of the player spaceship must be configurable, specified in and read from the sample JSON configuration file. Furthermore, the Enemy positions and projectile strategies are defined within the JSON configuration file.

- The Bunkers
 - If a bunker is hit by a projectile (either from the player or an enemy), the bunker receives damage (i.e. the colour changes) and the projectile is destroyed.
 - If a bunker is hit 3 times, it should be removed from the game.
 - The bunker will change colour according to the number of hits it has received, this represents its current state. Green = 0 hits, Yellow = 1 hit, Red = 2 hits.
- The Player Spaceship
 - You must be able to control the spaceship with the arrow keys to move the spaceship left and right across the screen.
 - Pressing the spacebar should shoot a projectile (Slow Straight Projectile) from the players position, moving in an upwards direction. This projectile will either hit an Alien or Bunker, or disappear at the top of the screen. The projectile can be shot at anytime (even while moving), however, only 1 player projectile can be on screen at a given time.
 - If a player projectile collides with an enemy projectile, both projectiles are destroyed.
 - If the player spaceship receives a hit from an enemy projectile, 1 life will be deducted. If there are no lives left, the game ends.
- The Enemies
 - All enemies must be able to move back and forth horizontally across the screen in an organised group, while descending down when they reach the edges of the screen.
 - Enemies must be removed from the game when they are hit by a player projectile. When this occurs, the speed of all enemies is slightly increased.
 - If any enemy reaches the bottom of the screen or reaches the player, the game ends.
 - If an enemy touches a bunker, the bunker is immediately removed from the game (i.e. the bunkers should not affect enemy movement).
 - Enemies shoot projectiles at the player at random intervals. There can be no more than 3 enemy projectiles on screen at a time.
 - Each enemy will shoot a projectile with a particular behaviour. The two behaviours are "Fast Straight Projectiles" and "Slow Straight Projectiles". The fast straight projectiles should travel twice as fast as the slow straight projectiles. This should be implemented using the strategy pattern,

Attention Please: You are only allowed to implement the above features in your current assignment, and you **should not include any other features**. Mark deduction will be applied otherwise.

Report Task

You are allowed a maximum of 1000 words report in this assignment which must **clearly and concisely** cover the followings:

1. A discussion on how your design (i.e., class diagram) for assignment 1 helped or hindered your design made in this assignment
 - Rationalise changes you have made to your assignment 1 design
2. A discussion on each design pattern you have used including
 - Where you used it (**be explicit** as to what classes are involved and in what roles)
 - What this pattern does for your code in terms of SOLID/GRASP principles
 - What overall benefits this pattern provides (**be specific** to your code, not the pattern in general)
 - What drawbacks this pattern causes (**be specific** to your code, not the pattern in general)
3. The UML class diagram describing your whole system, including design patterns.
4. Any acknowledgement/reference required.

Submission Details

You are required to submit all assessment items by the due date to different portals.

You have to submit your report on Canvas

(<https://canvas.sydney.edu.au/courses/53385/assignments/481509>)

Submit Assignment