THE UNIVERSITY OF SYDNEY
Account
Help
Dashboard
Courses
Groups
Calendar
Inbox
History
Studio
Support

2023 Semester 2
Home
Announcements
Modules
Ed Discussion
Recorded Lectures
Zoom
Assignments
Reading List
Marks
Unit outline (COMP9201)
Unit outline (SOFT2201)

Assignment 2 - Report

6 Possible Points

Due: Sun Sep 24, 2023 23:59

Attempt 1 — In Progress
NEXT UP: Submit Assignment

Add Comment

Unlimited Attempts Allowed
Available: Sep 5, 2023 0:00 until Oct 4, 2023 23:59

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 and here.

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 that you can start with for your implementing. A sample configuration reader is also provided here for your consideration.
- gradle file: a sample build.gradle file is provided to you here .
- A codebase 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:

- The game can be created in different sizes, with different sized bunkers at different positions, which must be configurable, defined in and read from the sample JSON configuration file. The colour, initial position (as an x-y coordinate), speed and lives 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** AND You have to submit **your code on Edstem**

- Report. Submit your UML class diagram and your report as a **SINGLE** pdf document on this portal.
  - If your UML diagram is too large, then you need to
    - 1) include the whole UML diagram;
    - **AND**
    - 2) include enlarged versions of the key components when you refer to them.
- Code. Submitted to Edstem by using this link. Your code should be submitted as **only** your src folder, build.gradle, and **readme**.
  - Attention Please: the sample json configuration file must be put into the resources folder of main under the src folder. Mark deduction will be applied if the location is wrong.
  - The readme file has to cover any point you would like your marker to know. In your readme, you must have the following items
    - how to run your code (e.g., any quirks to run your application)
    - which files and classes are involved in each design pattern implemented
    - anything else that you would like your marker to know
  - We will execute your code by running 'gradle clean build run' in the terminal with the environment configuration below:
    - Gradle 7.4.2
    - JDK 17
    - Unix-based System
  - If your code fails to run using the instructions provided above, all coding parts of your submission will receive a **ZERO** mark.

Academic integrity

While the University is aware that the vast majority of students and staff act ethically and honestly, it is opposed to and will not tolerate academic integrity breaches and will treat all allegations seriously.

Further information on academic integrity, and the resources available to all students can be found on the academic integrity pages on the current students website: https://sydney.edu.au/students/academic-integrity.html.

We use Turnitin, which includes AI detection, to detect potential instances of plagiarism or other forms of academic integrity breach. If such matches indicate evidence of plagiarism or other forms of academic integrity breaches, your teacher is required to report your work for further investigation.

You may only use artificial intelligence and writing assistance tools in assessment tasks if you are permitted to by your unit coordinator, and if you do use them, you must also acknowledge this in your work, either in a footnote or an acknowledgement section.

Further information for on research integrity and ethics for postgraduate research students and students undertaking research-focussed coursework such as Honours and capstone research projects can be also be found on the current students website: https://sydney.edu.au/students/research-integrity-ethics.html.

Compliance statement

In submitting this work, I acknowledge I have understood the following:

- I have read and understood the University of Sydney's Academic Integrity Policy 2022.
- The work is substantially my own and where any parts of this work are not my own I have indicated this by acknowledging the source of those parts of the work and enclosed any quoted text in quotation marks.
- I have acknowledged any assistance provided in preparing the work including the use of copy-editing, proof-reading, and automated writing and drawing tools (including artificial intelligence (AI), reference generators, translation software, grammar checkers, but not spell checkers).
- The work has not previously been submitted in part or in full for assessment in another unit unless I have been given permission by my unit of study coordinator to do so.
- The work will be submitted to similarity detection software (Turnitin) and a copy of the work will be retained in Turnitin's paper repository for future similarity checking. Note: work submitted by postgraduate research students for research purposes is not added to Turnitin's paper repository.
- Engaging in plagiarism or academic dishonesty in coursework will, if detected, lead to the University commencing proceedings under the Academic Integrity Policy 2022. and the Academic Integrity Procedures 2022.
- Engaging in plagiarism or academic dishonesty in research-focussed work will lead to the University commencing proceedings under the Research Code of Conduct 2013 and the Academic Integrity Procedures 2022.
- Engaging another person to complete part or all of the submitted work will, if detected, lead to the University commencing proceedings against me for potential student misconduct under the University of Sydney (Student Discipline) Rule 2016.

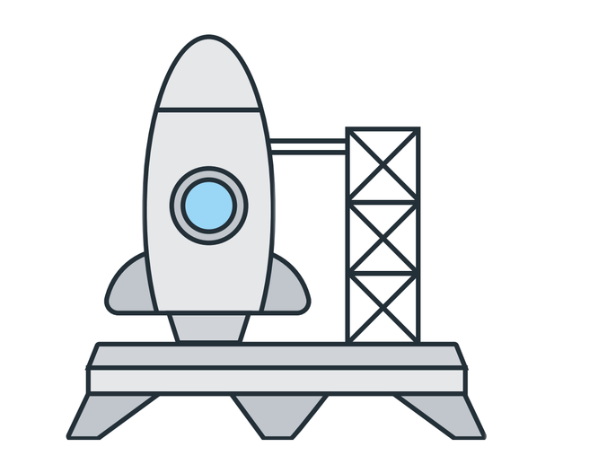
View Rubric

Assignment 2 Report

Criteria	Ratings			Pts
Discussion on Design Changes view longer description	**1 pts** **Full Marks** Covered the design changes you have made from the assignment 1 design (i.e., class diagram).	**0.5 pts** **Half Marks** Weak discussion on the design changes you have made from your assignment 1 design	**0 pts** **No Marks** no discussion on the design changes you have made from your assignment 1 design	/ 1 pts
UML Diagram -- Design Pattern view longer description	**1 pts** **Full Marks** Have correctly displayed the design pattern(s) used in the code implementation	**0.5 pts** **Half Marks** Have displayed the design pattern(s) used in the code implementation with flaws	**0 pts** **No Marks** Have displayed the design pattern(s) used in the code implementation with major flaws or not matching the code implementation	/ 1 pts
UML Diagram -- Correctness view longer description	**1 pts** **Full Marks** Have drawn all required elements correctly	**0.5 pts** **Half Marks** Have drawn some of the required elements correctly	**0 pts** **No Marks** Have drawn the UML elements with major flaws or haven't drawn the UML diagram	/ 1 pts
Identifying Design Patterns view longer description	**1 pts** **Full Marks** Clearly identify where the design patterns are used in the code implementation -- specific to your code	**0 pts** **No Marks** Incorrectly identifying the design pattern(s) used in the code implementation or no design pattern(s) presented		/ 1 pts
Discussion on Design Patterns view longer description	**2 pts** **Full Marks** Comment on benefits and drawbacks of design patterns you used according to design principles and OO theory -- specific to your code	**1 pts** **Half Marks** Comment on benefits and drawbacks on using these design patterns according to design principles and OO theory -- specific to your code -- with major flaws or omission	**0 pts** **No Marks** No discussion on benefits and drawbacks on using these design patterns according to design principles and OO theory or not specific to your code	/ 2 pts
				Total Points: 0

Choose a submission type.

Upload | Studio | More



Drag a file here, or
Choose a file to upload
File permitted: PDF

or

Canvas Files

I agree to the tool's End-User License Agreement
This assignment submission is my own, original work

Submit Assignment