SCC110: Software Development Term 2. Java Programming.

Problem Set 6: Evaluating and improving OO programs

Aims

This week's lab aims to:

- Provide you with time to improve and reflect on your group work, through feedback from your peers.
- Help you practice evaluating Java code written by others, making sure it follows object orientated programming principles.

Task 1: Evaluating each other's games

For the past few weeks, you have been developing your own Object Orientated games using the GameArena classes. Now it's time to showcase these games to your peers in your lab and gather feedback on the game itself, as well as your code!

Find another team in the lab to evaluate each other's code. If you prefer, please ask the teaching staff in the labs to set you up with another team.

Consider the following:

- **Does the game work as expected?** Are there any bugs or issues with the game? Are users properly informed of the status of the game e.g., through outputs of the score? Record any reported bugs/suggestions for improvement as a **github issue** on your repository.
- **Is the programming code object orientated?** Thinking of the game they have developed, what objects would you expect and what classes would be needed? Does this map how the team have designed the game?
- **Is the code properly encapsulated?** Have the team protected their code using private instance variables, and have they provided public accessor and mutator methods where required? Are they enforcing their classes to be initialised in a consistent way using constructors?
- **Does the code make good use of type composition?** Are there opportunities to reduce code complexity through type composition?
- **Does the code have good code style?** Have they ensured consistent use of brackets? Are Java naming conventions followed e.g., capitalised class names and camel case methods and variables?
- **Is the code well documented?** Have Java doc comments been used for each of the classes and methods, providing a well-defined API? Are in-line comments provided where code is particularly complex? Are the comments useful, i.e. could you use the Java doc comments to use and build on the game yourself?

Task 2: Updating your game

In your teams, discuss the feedback you received from your peers on your game. **Assign the issues you create** in GitHub to each member of the group to work on this feedback and improve your game.

Additional Task

I've created an open source repository containing my GameArena based Space Invaders game. Work **in your teams** to review this code, considering the questions from task 1. Remember, any other student from SCC110 may be doing this… so:

- Inspect the repository: https://github.com/finneyj/Invaders
- Discuss feature requests using **github issues** to gain a common view on what to change.
- **Assign issues** to people to ensure work is not duplicated.
- Research how to make **pull requests** to push changes to a repository you do not own.