Space Invaders of the COS 214 kind

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

- **Date issued**: 20 September 2019
- **Date due**: 28 October 2019 at 8:00
- **Submission procedure**: Upload via the CS website

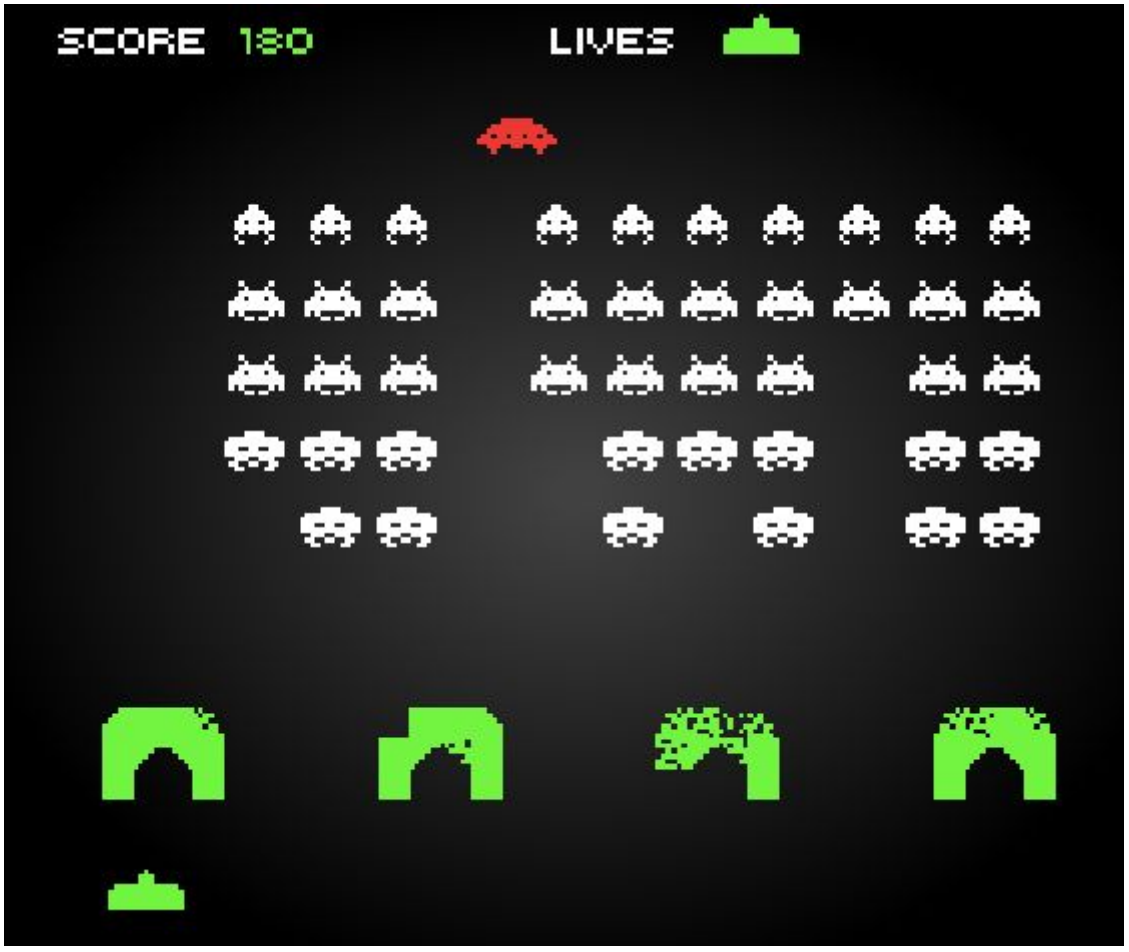

TABLE OF CONTENTS

1. Introduction

During this project you will make use of at least ten different design patterns, some of them already implemented in previous practicals. Some guidelines are provided on which patterns to use and how to implement them for the project. You are however allowed to replace them with similar patterns if you can provide a proper reason for the change. You may also add additional patterns if needed. In addition to implementing the project code, you are also required to submit Doxygen and project documentation. Please read the entire specification before starting the project.

2. Constraints

- This project must be completed in groups comprising of 3 to 5 members. You will be given an opportunity to register your group.
- During the project demonstration all members of the group must be able to fully explain all parts of the project.
- The project has to compile and run on the Linux system installed on the lab computers.

3. Submission Instructions

Submission of the projects comprises of two parts. The first are milestones which are spread throughout October 2019. The second is the final submission of the project.

3.1. Milestones

For each milestone, an upload will be created and you are required to upload your most recent working version of your group's project before the given deadline.
During the practical session, directly after the submission of a milestone, the teaching assistants will be available to discuss your progress and answer questions you may have.

The milestones and the corresponding functionality to be completed are as follows:

- 1 October 2019 - Building Spaceships
- 8 October 2019 - The Space Station and Manning the Spaceships
- 15 October 2019 - Day-to-Day Running of the Ship and Finding Habitable Planets
- 22 October 2019 - Interaction with Critters

3.2. Final Submission

You are required to upload all your source files and documentation as a single archive (tar.gz or zip) to the CS website before the final deadline. The archive should contain two PDFs, one for the Doxygen documentation and one for the project documentation with your project description and UML diagrams. No other formats besides PDF are allowed. You do not have to bring a hard copy of the documentation to the project demo. No late submissions will be accepted. Do not show up during the marking session with your code on a flash drive, since your code from the uploaded archive will be used during the demonstrations.

4. The Story

Earth has been flattened by a good-hearted soul! Rather than building his super highway (apologies to Douglas Adams) and removing the green and blue planet without warning, Mellenia contacted the powers on earth and gave them a hundred year warning about his plans.

Skip ahead to 2119. Well, the earthlings rallied and were able to leave their planet just in time. This is thanks to your team, when they in 2019 heard about the “imminent” disaster that was about to take place, decided to build a simulation for man to invade space in a way never seen before. *Space Invaders of the COS 214 kind* (with apologies to Tomohiro Nishikado the creator of the original Space Invader game and Steven Spielberg the creator of Close Encounters) had just dawned.

5. Pattern Overview

The following list of patterns links the suggested pattern to be used for the described functionality. This is not prescriptive. You are required as part of the documentation to provide the rationale for using a particular pattern to model specific functionality.

Pattern used	**Section in which it could be used**
Abstract factory	Building Spaceships[i], Manning the Spaceships[ii]
Decorator	Building Spaceships
Prototype	Building Spaceships
Observer	Day-to-Day Running of the Ship, Finding Habitable Planets
Mediator	Day-to-Day Running of the Ship, Finding Habitable Planets
Composite	Building Spaceships
State	The Space Station, Day-to-Day Running of the Ship
Strategy	Manning the Spaceships, Day-to-Day Running of the Ship
Visitor	Should be used where appropriate.
Iterator	Should be used where appropriate.
Façade	Running the Simulation
Builder	Building Spaceships
Factory Method	Building Spaceships[i], Manning the Spaceships[ii]
Template Method	Should be used where appropriate.
Flyweight	No specific reason to use.
Singleton	The Space Station
Bridge	Could be used where appropriate.
Command	Day-to-Day Running of the Ship, Finding Habitable Planets
Memento	Building Spaceships, Day-to-Day Running of the Ship
Adapter	No specific reason to use.
Proxy	No specific reason to use.
Chain of Responsibility	Day-to-Day Running of the Ship
Interpreter	Interaction with Critters

[i],[ii] Either of these patterns can be used

6. Functional Requirements

6.1. Building Spaceships

Spaceships are built to provide specific functions resulting in different types of spaceships. Examples of spaceship types include:

- Fighter transporters - transport the fighters to and from the Battleships and other planets.
- Spaceship transporters - transport other spaceships
- Frigates - carry the top brass
- Exploration vessels - sent out to planets to scout the area
- Battleships - protects other spaceships

Spaceships are made of components. These components can be built in different ways. Each type of spaceship has a minimum blueprint to which it must adhere. Spaceships are created from the blueprints by different factories.

Each spaceship has displacement, power, thrust, max speed, stall speed, crew and passenger capacities. All actions a spaceship makes will result in the consumption and depletion of resources. Smaller ships might have to rely on bigger ships or a space station for refuelling or restoring of energy.

All ships have a bridge, sick bay and sleeping quarters. On some ships they are more fancy than others. All the different components are manned by different crew members (refer to the section, Manning the Spaceships) and may be transporting passengers.

Spaceship transports will also have the functionality to save the specific spaceships that are carried within them at a certain time, but this will be added in the next section.

6.2. The Space Station

The sole space station forms the center of the space invasion effort. It is a special type of spaceship with all the characteristics of a spaceshift. The space station moves slowly and is home-base to the spaceships. Spaceships are brought to the space station to change crew, be repaired etc. It is where the Commander of the Fleet is stationed if not on one of the spaceships.

A space station may be at different threat levels. This changes the behaviour of the station.

6.3. Manning the Spaceships

People are placed on the ships from the space station. Each person on a ship is either a member of the crew or a passenger. Members of the crew have role designations. Some crew members may have dual roles. When not on duty a crew member reverts to having the role of a passenger.

Crew member designations:

- Captain - is the highest office on a spaceship if the Commander of the Fleet is not present.
- Navigator - each spaceship has a navigator. Navigators communicate with each other. Periodically, navigators report their current location to their comms representative on the spaceship who in turn forwards this to surrounding spaceships and to the space station.
- Comms - manages all communications within the spaceship and to other ships in the fleet.
- Doctor - is in charge of the physical and mental well-being of the crew and passengers on a spaceship or on the space station.
- Chief Engineer - manages the repairs of the spaceship
- Engineers - report to the Chief Engineer and are given repair tasks to complete.
- Fighter - fights off the enemies. Are transported on fighter transporters between battleships, the space station and possibly other spaceships and planets.
- Commander of the Fleet - represents the highest office. All Captains of spaceships in the fleet report to the Commander.

One ship carries the Commander of the Fleet when the Commander is not at the space station. And this ship should be well protected from damage and critters (refer to the section, Interaction with Critters).

Spaceship transports, added in the first section, must also be adapted with the functionality to save the specific ships that are stored in them at a specific time. These save states are necessary for logs in case a spaceship's records need to be examined.

6.4. Day-to-Day Running of the Ship

The day-to-day running of the spaceship not only influences the current state of the spaceship, but may influence other spaceships in the fleet as well as the space station and habitable planets. With spaceships influencing others, it is conceivable that a spaceship may be influenced by external factors.

Some factors which influence the day-to-day running of the spaceship include, but are not limited to:

- Every day announcements are made about the progress of the fleet. All crew members should deliver their respective daily reports. If any problems are raised during the reports, the crew responsible need to fix them. For example, engineers will have to repair the ship, and doctors should tend to the injured. Different crew members might also report on different types of inventory within each spaceship. In cases where crew members report problems that higher ranked crew members need to attend to, these reports are passed up the chain of command until they are received by a crew member that is able to attend to them. That is, if some system in the doctor's quarters fail, their report will be passed onto the engineer. If the engineer is unable to fix it, they pass on the report to the chief engineer, until a crew member is found with high enough rank to fix the problem. If a problem cannot be solved by any crew members on a certain ship, the space station must be notified of this.
- General announcements which are aimed at crew and/or passengers can be made. The may include invitations to social events or just general information which take the well-being of the crew and passengers into account.
- To steer a spaceship through the unknown, strict protocols are followed, especially in the reporting structures on the bridge.
- To survive in space, mechanisms to procure - either by purchasing or sourcing of the raw materials and then manufacturing - resources is a necessity.
- Every spaceship keeps a log - usually referred to as the Captain's Log, in which the events are documented and stored for posterity.

6.5. Finding Habitable Planets

Multiple exploration vessels (occupied with fighters) are sent out to possible habitable planets. Their mission is to determine if the planet is safe or not. They should study if the planet is worth it to explore (that is, does it have the necessary resources to sustain the fleet and serve as a replenishing stop for the fleet) or not. A planet is not habitable if the danger of leaving people on the planet outweighs that gain in terms of resources or having a replenishing stop. The spaceships should then report back to the Commander of the fleet who on a decision may send command for additional fighters to be transported to the planet to ready the planet for habitation and defend the planet against other critters. Once the planet is suitable for habitation, spaceships transports them as passengers to the planet.

6.6. Interaction with Critters

Critters on other planets or in their own spaceships may be friendly or not. Friendly critters have a level of tolerance. When this level is depleted - for example if you use them too much rather than having a two way relationship - the critters become irritable and may turn into enemies. Critters that are regarded as enemies need to be defeated. Once defeated they are assigned to a spaceship in your fleet and either work or are imprisoned.

Communication with critters can be difficult. They are not necessarily proficient in English.

6.7. Running the Simulation

Simulations can be to defend the fleet from an enemy when under siege, penetrate uncharted planets to increase resources that may need to be replenished. Additionally you should provide enough information to show how you have implemented everything and how it all works together.

Be creative. The user should be able to simulate all functionality using a single interface.

7. Bonus

7.1. Additional Patterns

Bonus marks will be given for any additional patterns you add to the project up to a maximum of 3 patterns. Note, when adding a pattern you must be able to state why it is a good choice and how you implemented the design pattern. Additionally, a more detailed user interface will more likely receive better marks i.e. user interaction with the program.

7.2. GUI

The system till now was in all likelihood a text-based interface. For a few additional marks, design and implement a Graphical User Interface (GUI) for your system. Tools such as Qt can be used to implement the GUI.

8. Tools

Working as a team can be difficult without the necessary tools. Therefore you are advised to take a look at the following tools and recall the lectures on them.

8.1. Git

There are great version control applications for Git. Take a look at GitHub and set up a team environment for your group. Note that everyone in your team should contribute to the project through git. ***Note: link your student email to make your GitHub profile a Pro profile. GitHub Student Kit.***

8.2. Visual Paradigm

Everyone should have access to the enterprise edition for Visual Paradigm. You should use this with your team to design your system and export the necessary diagrams.

8.3. Doxygen

Code comments allows you to understand what the code does. Doxygen takes all your comments and produce a well formatted document to easily see what your code does. Make sure everyone in your team adds their comments - it's not fun to write them when you forget what your code does.

9. Documentation

In addition to the automated Doxygen documentation, you will have to submit an additional PDF containing your project documentation. This should include a general overview of the project, descriptions of the individual design patterns and how/why they were used. If you decide to use different or additional patterns to the ones suggested, you should also document this.

Add UML diagrams that were drawn using Visual Paradigm. You need a class diagram for the entire project and at least one of each of the other UML diagrams covered in the course. For instance, the Observer and Mediator can be represented with a sequence diagram. A state diagram can be used to illustrate the current state of an object, for example a spaceship, and the state changes it can experience.

You don’t have to write many pages of documentation. Taking the diagrams into account, five to ten pages should suffice (excluding doxygen pages). Professional and correct documentation will earn more marks.

Hint: *You could use the output to Latex of the Doxygen comments included in your code as a starting point for your documentation.*

10. Demonstration

You have to demonstrate your project during the booked sessions on 28 and 29 October 2019. All members of the group have to be present and should be able to fully explain all aspects of the project. If you do not pitch for the demo session, you will forfeit all marks for the project, even if you uploaded a functional program. You will be required to compile and run your program from the uploaded archive and explain various parts of the project by referring to the submitted code and documentation. The marker will ask you questions about the implemented design patterns and you will have to show and explain the necessary code. Please arrive at least ten minutes prior to your booked time to set up.