

EECS 3311 - Software Design

Project 2

Group: Pritesh Ramani (217174020),

Smit Shiroya (217614728),

Isha Gajera (217629288)

Course: EECS 3311

Section: A

TA: Kazi Mridul

Email ID: prits209@my.yorku.ca

Introduction

Explain what the software project about and what are its goal:

The software project is all about completing the implementation of the 2d soccer game where we are a striker and we need to try to hit goal in front of AI which moves randomly in front of the goal post. The goals are to complete the implementation of the code such as write the missing classes, methods etc and also test cases to cover the maximum coverage of the code.

Explain the challenges associated to the software project:

The main challenge associated with the software project was to cover the maximum coverage area for test cases and finding missing classes with implementation.

Explain the concepts (e.g., OOD, OOD principles, design patterns) you will use to carry out the software project:

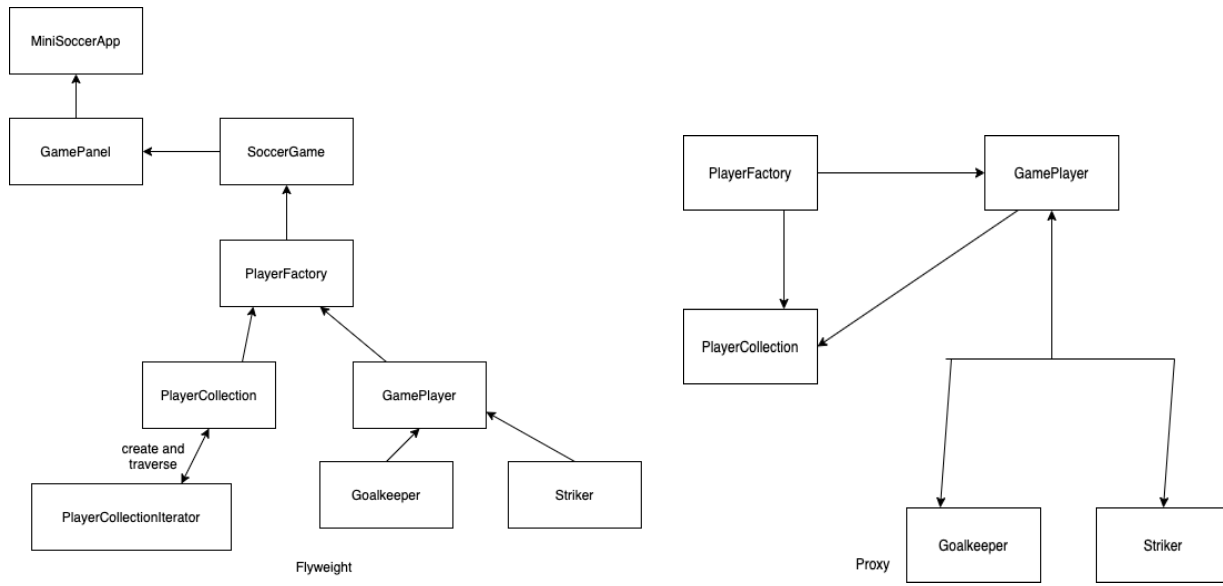
Design patterns such as Factory and Bridge were used. Bridge pattern was used for iterator and collection class. Also inheritance and super was used for calling constructor from striker's and goalkeeper's object.

Explain how you are going to structure you report accordingly:

I'll use all the possible terms mentioned in the presentation pdf included with the project question on eclass to make sure that we satisfy all the things and rubrics.

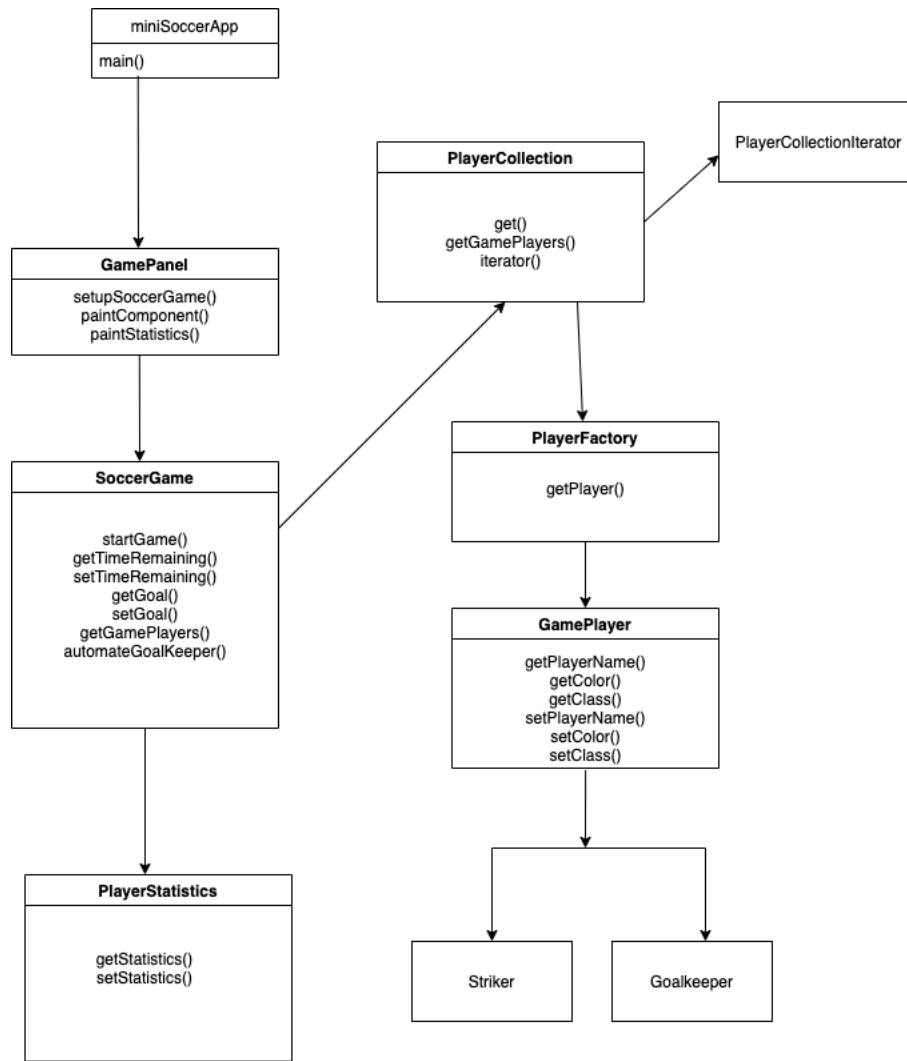
Part 2

Design of the Solution



I've used bridge pattern for **PlayerCollection** and **PlayerCollectionIterator**. Also there's inheritance in proxy as **GamePlayer** is parent class and **Goalkeeper** and **Striker** are child classes.

As both classes share same characteristics such as player info we can save space by writing common methods and later identifying that which objects is belonging to the class.



Part 3

We've started with clearing out all the errors and then when classes are made, we started looking for the flow of the execution of project. Later when we found it out we struggle a bit in iterator part but in the end we found out the solution. And then we keep solving new errors and it led to a successful run as more than half of the project was already made.

We've tried to cover as many as classes used in about class diagram to know about its method and flow of execution. Striker and Goalkeeper are player classes and it has a parent class name GamePlayer. PlayerFactory stands for creating a object or a player. PlayerCollection does the task of storing all the players used by soccerGame. Also there's iterator class to traverse from the PlayerCollection class. PlayerStatistics is used to get and set score of the players. While soccerGame and GamePanel is used to control and draw the game respectively.

We've also tried to cover as much as code possible by test classes. Jacoco helped a lot to identify the code coverage. It also showed us how much coverage is possible for one single test class of particular model.

We've used Eclipse, Junit 5, JDK 13 and Jacoco.

Part 4

As we solved all the errors and gave the correct flow of execution. All the things were completed and project code prewritten was working very well.

Finding out the iterator class was a bit difficult and how to run it successfully was bit tricky.

I've learned about AI moving object, key controlling player and a proper followed MVC pattern.

The advantages are you always feel motivated because there're people working with you. Also you feel supported enough to carry out thing by yourself even if it's about to go wrong. The drawbacks are you need to narrow down to all the solutions and decide one of the best which can led to a debate.

The top three recommendations are first is providing hint for iterator class, And a bit coding implementation in striker and goalkeeper class for more exposure.

Pritesh was given work for carrying out coding and helping a bit in report writing. While Smit has done all the writing and findings. Isha has completed all the diagrams and corrected all the wrong ones out.