Bitcode Assignment 3

1 Design Patterns

Using design patterns in software project is a good practice. It helps to make your software understandable, sustainable and expendable. We have chosen two design patterns and implemented them in our existing code, the state pattern and the strategy pattern.

1.1 The State Design Pattern

why? tbd

1.1.1 State Class Diagram

tbd

1.1.2 State Sequence Diagram

tbd

1.2 The Strategy Design Pattern

why? tbd

1.2.1 Strategy Class Diagram

tbd

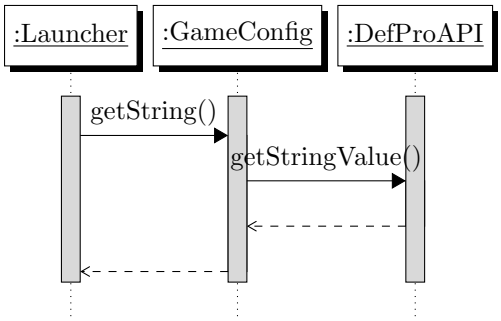
1.2.2 Strategy Sequence Diagram

tbd

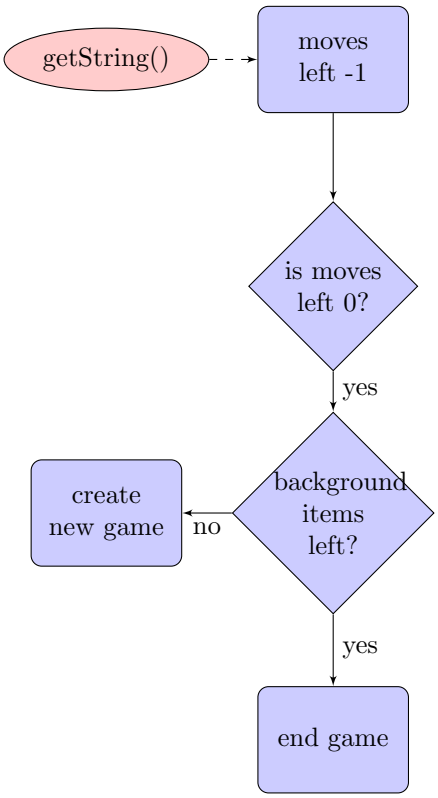
2 Defensive Programming

In the previous assignments it was required to use a game configuration file for initializing variables using a provided library. However, the provided library introduced bugs in the game.

To solve the bugs we have created a wrapper class (CameConfig) around the library's API that catches all the bugs and checks if the variables are within limits. This means that for every different API call we created a method in the GameConfig class. In the sequence diagram below is shown how the wrapper class works.



In every method in the CameConfig class that implements the API checks are build in to verify the data that is returned by the API. Also if the API throws an exception it is catched in the method. If an exception occurs or the data returned by the API is not within the defined boundaries the method will return the defined default value. The flowchart below shows how the method getString() is implemented.


getString()
moves left -1
is moves left 0?
yes
background items left?
no
create new game
yes
end game

3 Selectable Difficulty

story

3.1 Requirements

tbd

3.2 Software Design

tbd