CAN'T STOP

The video game

Fully Dressed Use Case Description

Design Patterns

Banana inc.

USE OF DESIGN PATTERNS

For the Can't Stop board game, we have used Information Expert pattern in implementing the game. The GameBoard class, all the game statistics, including the players, their current positions, current turn etc. This class follows the information expert pattern.

We have also used the Creator pattern in implementing the Can't Stop board game, In the game class, we have two factory methods which create the GameBoard class objects. One of the methods creates a GameBoard with default state, the other method loads the GameBoard with the data from the save file.

We have aimed to use Low coupling pattern from the Grasp Collection. The Grid Square class is one example where it is responsible for creating the grid of the game board, it doesn't depend on other classes to perform the actions required.

We have also used High cohesion by making different classes performing different functionalities of the game such as Main menu class, setup a game class, take turn class etc.

Pure Fabrication and Indirection pattern can also be observed for the loading and save functionality. The GameBoard class is responsible for saving and loading the game.

Some components of the Can't stop game is designed in a way such that to protect from variations in other components. The GameBoard class contains the game logic such as moving runners, the game state etc. Protected Variations pattern is observed to protect the components of the game.

The Observer pattern could have been used to improve our Can't Stop game. We tried to implement this pattern in certain aspects of the game such as the value of the dice with GameBoard, but due to the time constraints we were unable to implement this observer pattern.