

FRUIT NINJA



Design Patterns

Singleton Design Pattern:

The intent of using this design pattern in our game with the player is that our goal was to make sure that throughout the whole game, only one instance of player is created. This also ensures that the attributes in PlayerSingleton are the same throughout the whole game.

Command Design Pattern:

The intent of using this design pattern in our game is that it holds a set of commonly used commands during the game like, save, load, start a new game,...etc. Command design pattern facilitates calling these commands to be executed from any of the game classes using an invoker.

Factory Design Pattern:

The intent of using this design pattern in our game is that our game objects, fruits and bombs, are created through it. They all implement the same interface as they all share the same methods such as slice, getVelocity, getLocation,...etc. The type of fruit or bomb is specified and then created accordingly through polymorphism.

State Design Pattern:

The intent of using this design pattern in our game is that our game objects should appear on screen but by different states, these states are the different levels of difficulty that change according to the player's current score.

Memento Design Pattern:

The intent of using this design pattern in our game is that it helped us to capture a specific moment in the game and save it to be restored later on when needed. The game is saved when the player pauses the game or reached a checkpoint to catch up from where he left his/her unfinished game when he opens the game again by clicking on "continue".