

# Team redpanda: Sprint0

University of Toronto Mississauga, CSC207 Assignment 3

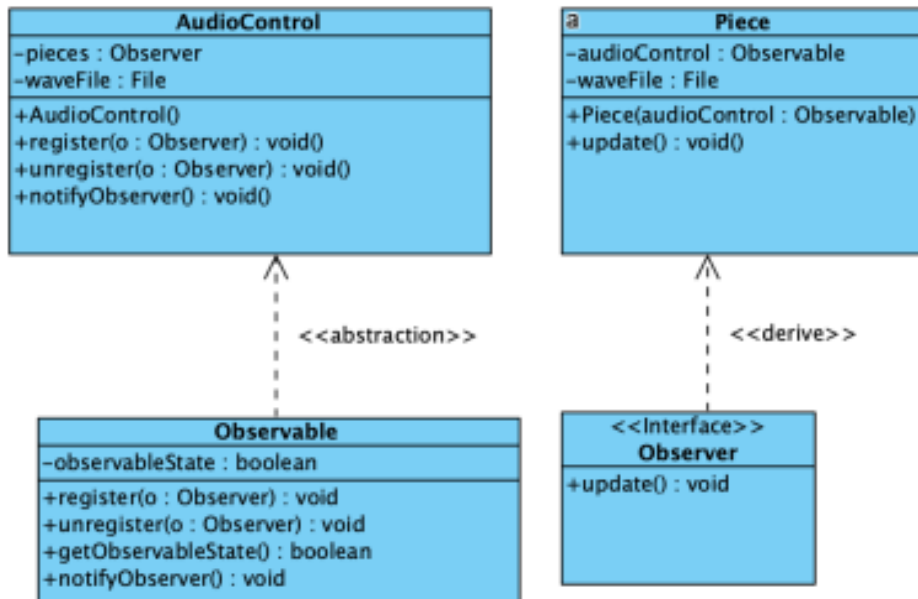
DATE: Sunday, November 21st, 2021

GROUP MEMBERS: Prajwol Poudel, Arushi Madan, Jasmine Chai, Osama Chaudhary

## Table of contents

<b>Observable pattern</b>	<b>2</b>
<b>MVC pattern</b>	<b>3</b>
<b>Command Pattern</b>	<b>4</b>
<b>Iterator Pattern</b>	<b>5</b>
<b>Factory Pattern</b>	<b>6</b>
<b>Prototype Pattern</b>	<b>7</b>

## 1. Observable pattern



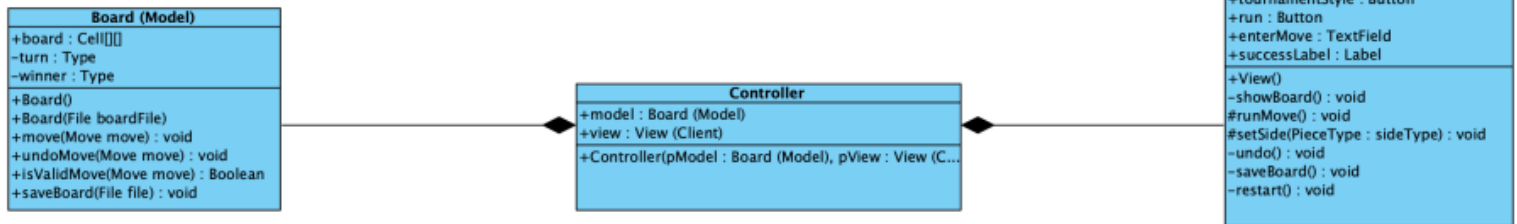
**Description:**  
 How does the Observer Pattern Work?  
 The Observable **AudioControl** updates each **Piece** by updating its `waveFile` attribute with the `File` present in the **Observable** `waveFile` attribute. It does so by calling `notifyObserver()`

**What does it solve?**  
 It simply adds entertainment value to the game, as most games have sounds.

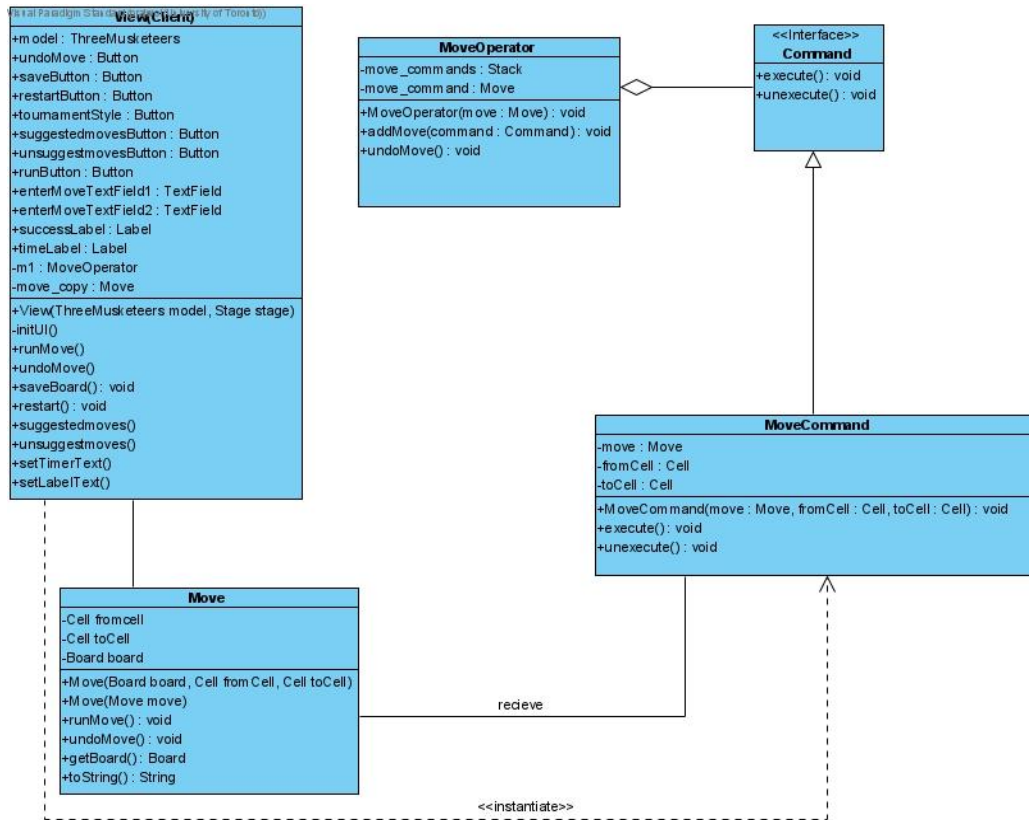
## 2. MVC pattern

**Description:**  
 How does the MVC Pattern Work?  
 successLabel is added to the View of our GUI which is a Label that indicates whether the move was successful or not. When a user clicks on a cell. The handle function of the boardPanel that we create for our GUI will set an attribute of our boardPanel class with "Success" or "Fail". In the updateCells() function of our boardPanel GUI, we will set the view successLabel attribute to the new boardPanel variable we just created.

**What does it solve?**  
 This allows the user to know whether their move failed or succeeded. They will not be clicking aimlessly on the screen



### 3. Command Pattern



4.

#### What does it do :

The Command Pattern is used here to run and undo move commands in the board.

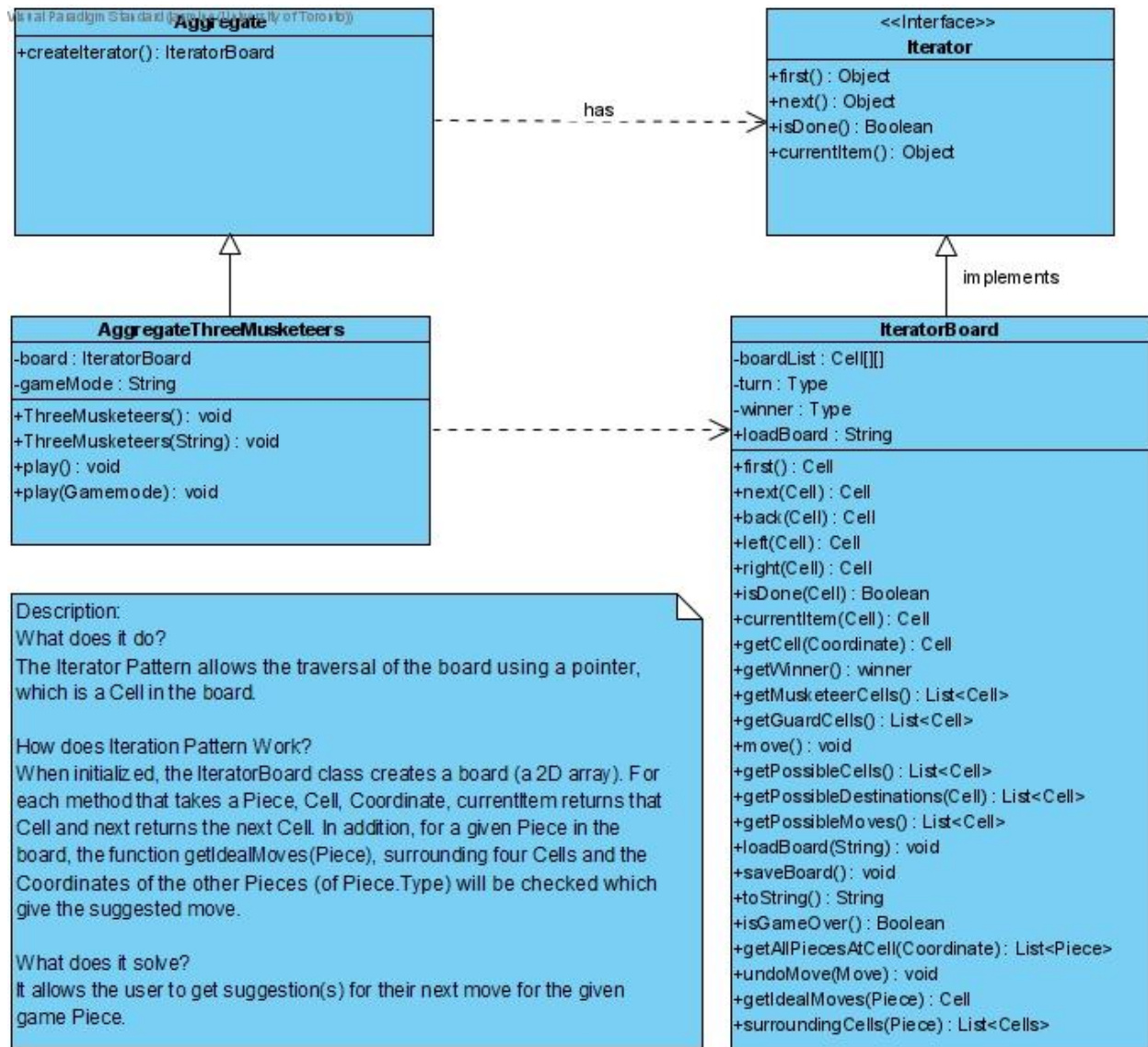
#### How does it work:

Here, the view(client) asks for a command to be executed when the user presses the buttons, run or undoMove. Then, a MoveCommand is created from the user inputs. The MoveOperator(Invoker) takes the new MoveCommand, encapsulates it, adds it to the move\_commands list and calls the execute method and then the Move class performs the operations. When the undo operations are called the invoker removes the last performed move.

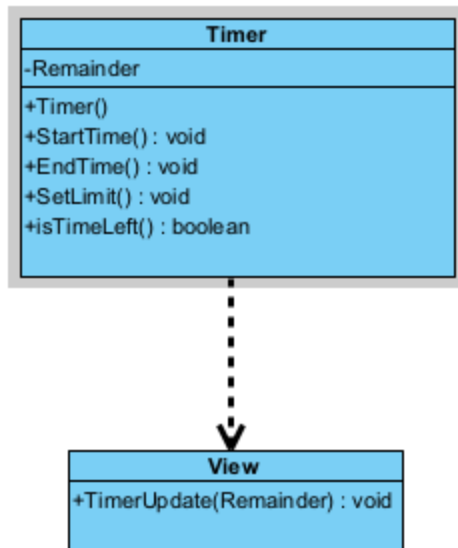
#### What does it solve:

It allows for encapsulating the commands and storing them in a list, or undoing the last command.

## 5. Iterator Pattern



## 6. Singleton Pattern



### Timer Implementation

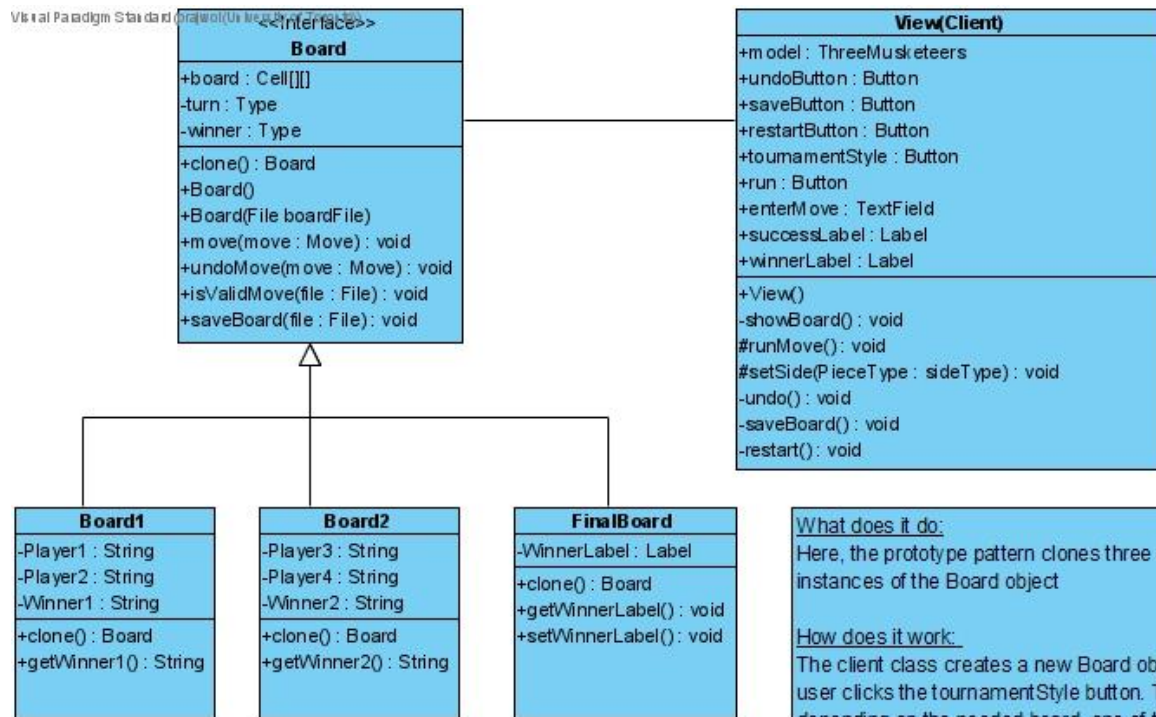
#### What does it do?

The timer counts down from a set time, and the player's whose turn it is, and provides a time limit to make the move, if a move is not made within the time frame, the game will move on and it will be the next player's turn

#### How does the singleton pattern work?

it simply starts with the creation of the timer object in the timer class, which would have many functions that will be implemented. The view class would contain the GUI elements that would use

## 7. Prototype Pattern



### What does it do:

Here, the prototype pattern clones three custom instances of the Board object

### How does it work:

The client class creates a new Board object when the user clicks the tournamentStyle button. Then, depending on the needed board, one of the boards will handle the cloning and makes a new instance of itself.

### What does it solve:

It allows for more people to participate in the game and also using the prototype pattern uses less computer resources.