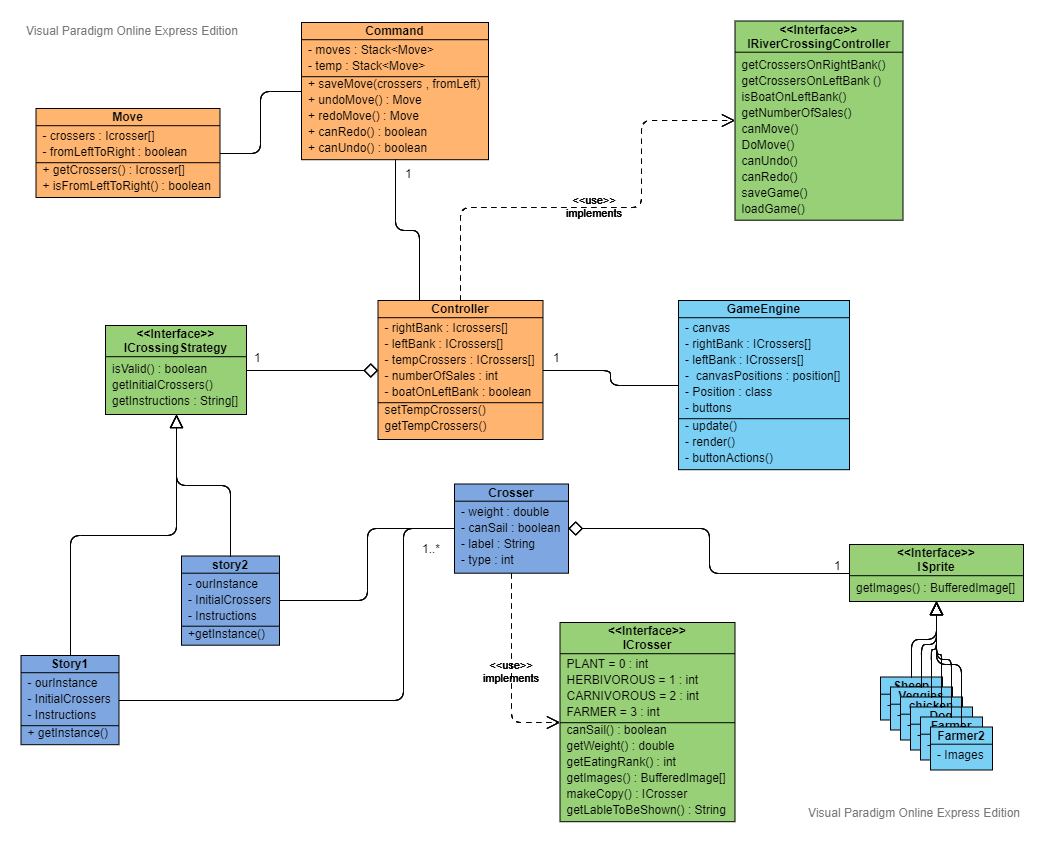
River Crossing Puzzle

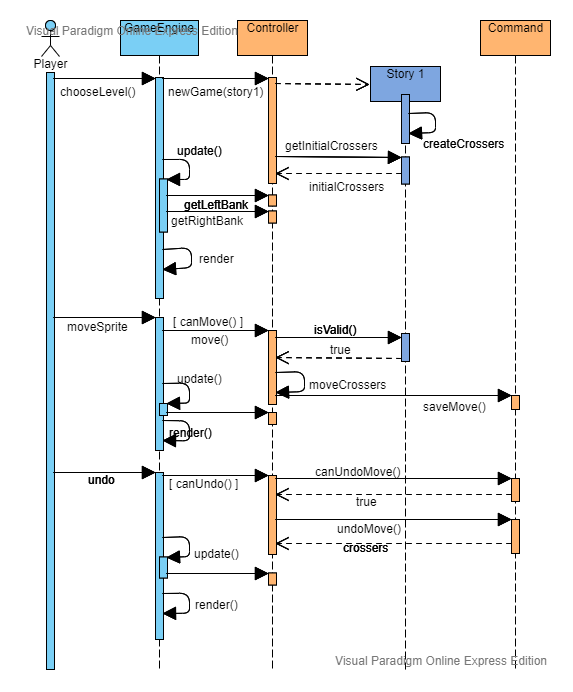
Made with JAVA and JAVAFX

George Amgad Wadie 5002

# Class Diagram:



# Sequence Diagram:



# Main design:

## GUI:

To implement the GUI the project is using fxml files with scene builder to design the user interface and to control the buttons, the file is using the GameEngine class as the controller.

There is only one user interface which is the game itself other interfaces are dialogue boxes created and destroyed by the game engine itself.

Each button has an onAction method that is implemented in the Game Engine and called when the user clicks on the button.

Inside the UI there is a canvas that is responsible for viewing all the movements of the sprites inside it and return the position of clicking actions made by the user and the game engine process the position and move sprites accordingly.

Also, the game engine is responsible for all the communications between the UI and the main game controller.

## Resources:

All the resources ( icons, images and fxml ) of the game are saved in a file named resources to make the project files neater.

## Game strategies and levels design:

The game levels are implemented with strategy design pattern to make it easier to change them when the user ask for new game also each level is a singleton because it’s never change and the crossers type inside it never change it only send a copy of what it holds whenever asked for.

## Saving and loading:

The project use xml files with DOM parser and external XStream library to save and load the current state of the controller.

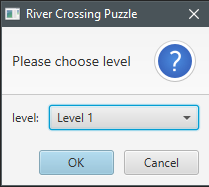
## Undo and redo:

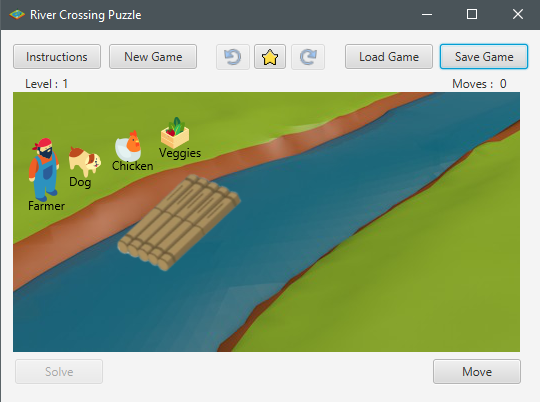
the project uses a design to implement undo and redo similar to the command design pattern which is separating the caller from the receiver but here there is only one caller and one receiver so the pattern is modified a bit with a class called command which holds the stack of the moved and pop from it whenever undo is called and push moves in it whenever a move is initiated.

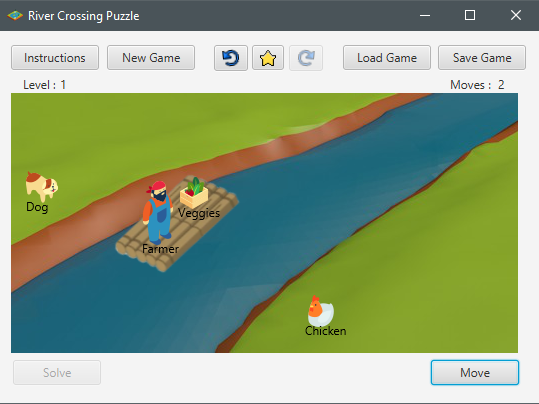
## Crossers and visuals:

The crossers is implemented using the strategy design pattern to allow different types of crossers for different types of strategies for example each crosser can have a different weight, type, image, ability to sail and a label to be shown and each strategy can create the crossers it need also the game engine and controller are designed to handle and view any type of crosser.

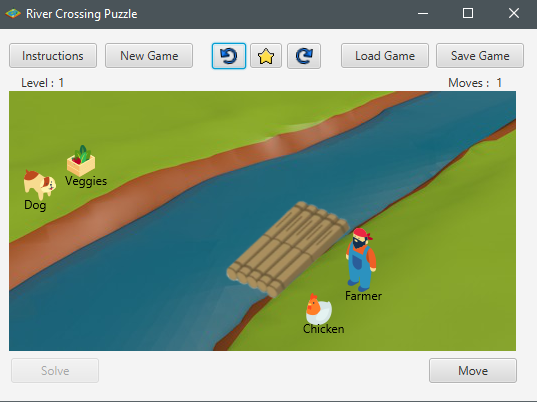
# Screen shots:



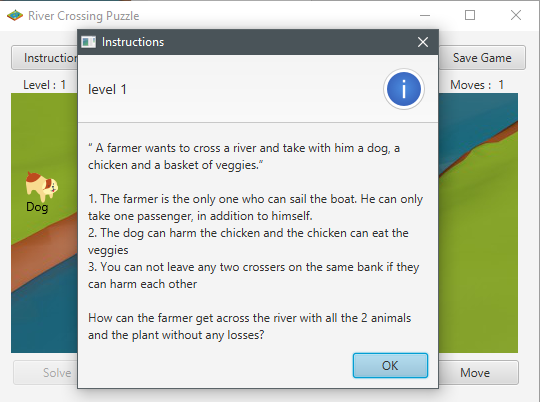




## Undo move :



## Instructions:



## New Game:

