

GUI VIEW

The GUI is written in Java FX and is composed of different scenes and 3 stages:

- Stage : main application stage
- popUp Stage : stage used only for pop ups
- chat Stage : stage used only for chat

SCENES :

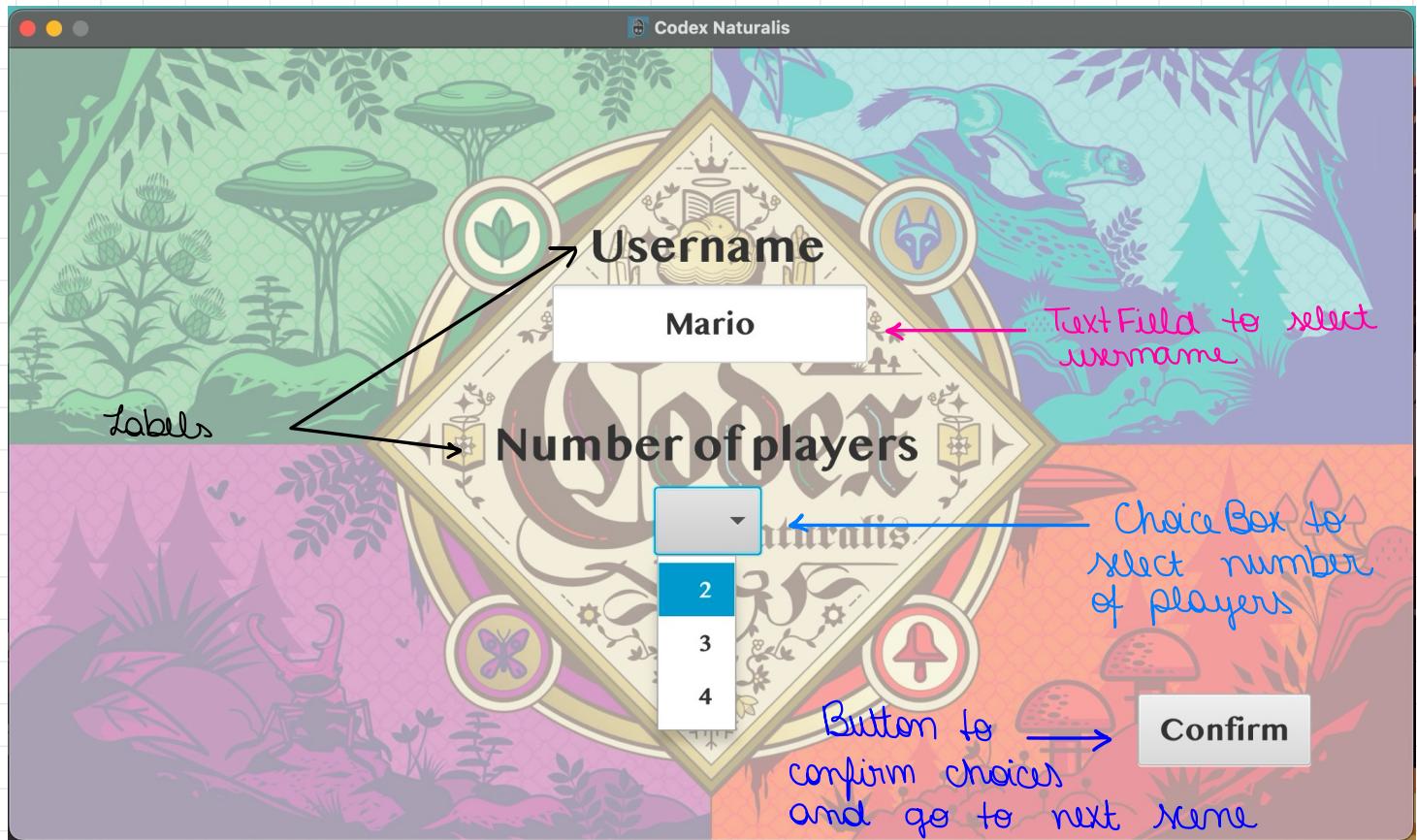
- PlayScene



Clickable imageView
that leads to
next scene

- Start Menu Game

: CREATE MODE

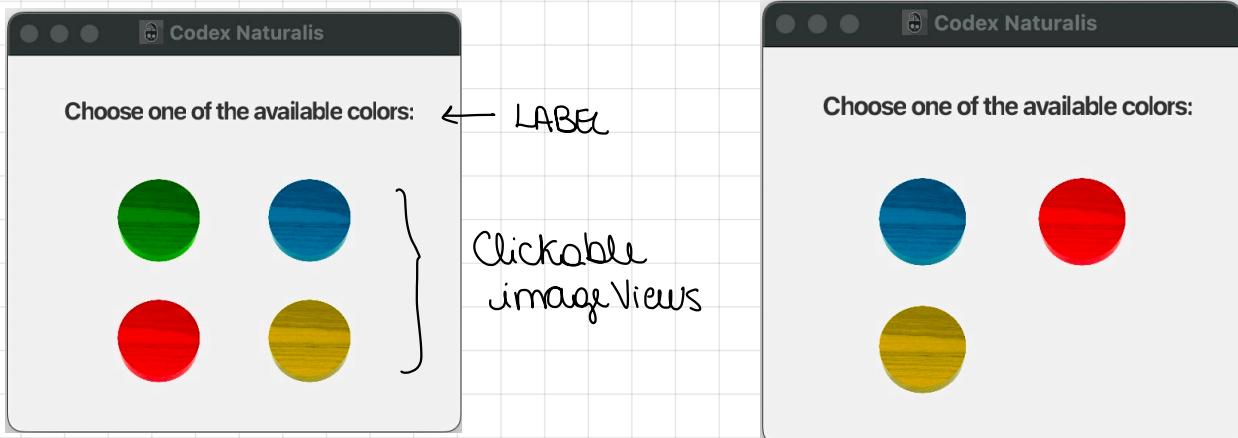


In case of join mode , numberofPlayers label and choiceBox are hidden .

- Loading Scene : contains the loading gif as back - ground

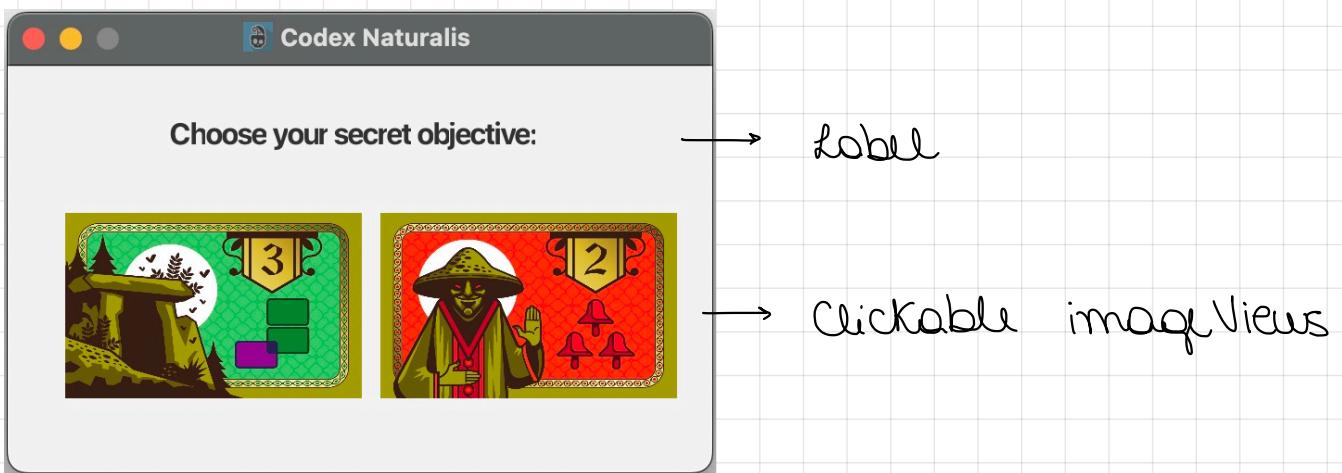


- Token Color Choice Scene :



The scene is displayed in the popUpStage and gets dynamically updated as players choose their colors

- Private Objective Choice Scene



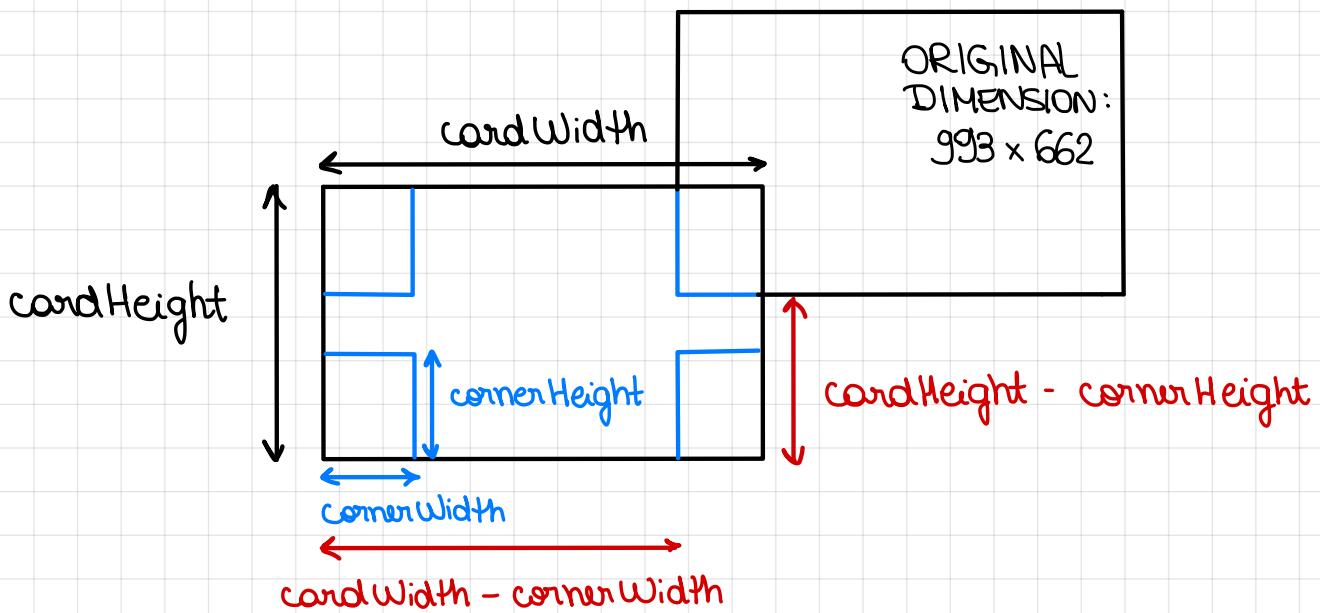
The scene is displayed in the popUpStage and is dynamically constructed with 2 random objectives for each player

- Main Game : Some where game will develop, composed of different classes that act as nodes in the same graph.



All elements in the main scene are dynamically updated and will be briefly explained:

- The board is a ScrollPane that contains an AnchorPane that will hold the cards. Its dimension was calculated to accommodate the maximum number of cards that is possible to place in the board during a game, i.e., 41 cards in a diagonal pattern, as follows:



Worst case scenario : 41 cards diagonally

Board's dimension :

$$\text{board Width} : \text{cardWidth} + (41-1) \cdot (\text{cardWidth} - \text{cornerWidth})$$

$$\text{board Height} : \text{cardHeight} + (41-1) (\text{cardHeight} - \text{cornerHeight})$$

The positioning of the cards in the board to make them overlap was calculated as follows:

- Initial card in the middle of the board .

$$\begin{aligned} \text{Layout X : } & \frac{\text{boardWidth} - \text{cardwidth}}{2} \\ \text{of Initial Card} & \end{aligned}$$

$$\begin{aligned} \text{Layout Y : } & \frac{\text{board Height} - \text{card Height}}{2} \\ \text{of Initial Card} & \end{aligned}$$

- Layout X and Y of following cards :

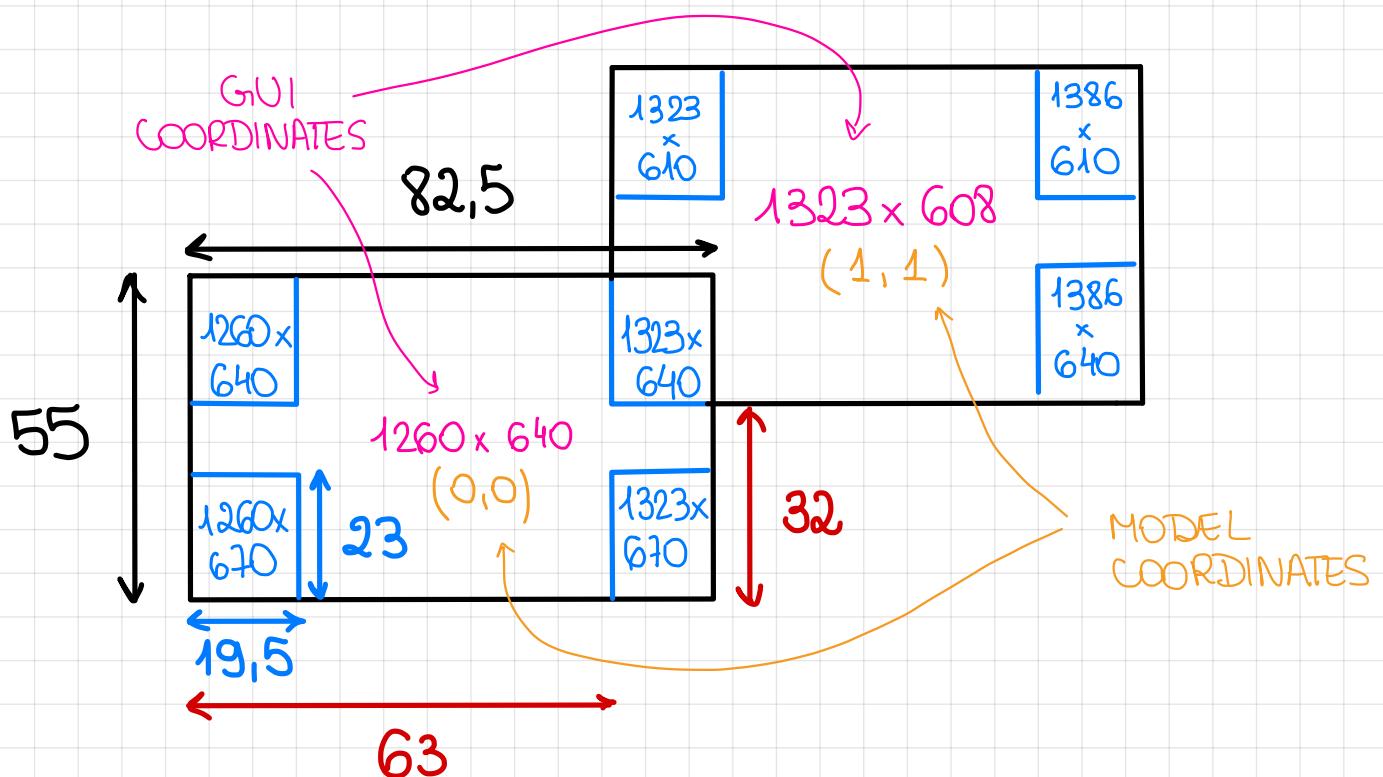
$$X : (\text{cardWidth} - \text{cornerWidth})x + \text{Layout X of Initial Card}$$

$$Y : \text{Layout Y of Initial Card} - (\text{cardHeight} - \text{cornerHeight}) \cdot y$$

} EQUIVALENT TO CARD (0,0)

Where x and y are the model coordinates of the card to be placed on the board

So, for example, if $\text{cardWidth} = 82,5$, $\text{cardHeight} = 55$, $\text{cornerWidth} = 19,5$ and $\text{cornerHeight} = 23$, this would be the configuration of the initial card plus a second card on the board.



The corners when clicked indicates where to place. When to the board, according to

of the cards are regions that set the model coordinates that player wants the next card to be placed. a card is then created to be added to the board, its layout is automatically set according to the model coordinates.

The relation between the card's coordinates and the next card's coordinates (based on what corner was selected) is calculated on the Corner Coordinates Calculator class, used also for the model.

- The **hand** is an HBox that contains 3 cards that can be selected to be added to the board. The hand of a player can be flipped at all times during the game.

The board and hand are unique elements for each player. Both can be seen by other players by clicking the "see other player's game" button, that display the board and all hand's cards facing down to the player that requested

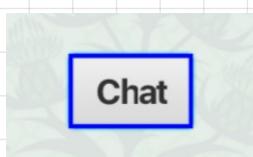
it. The elements can be seen but cannot be interacted with when this functionality is enabled.

- The **objective's section** is an HBox that contains the private chosen objective (visible only to the player that choose it) and the public objectives (visible to all players in game)
- The **drawable area** is composed of 2 VBoxes to contain the resource and gold cards that can be draw throughout the game.
- The **scoreboard** is a GridPane constructed with all the players usernames (in the order of their turns) and with the players scores that are updated as the game moves on. When one of the players hits 25 or more points in the score board, an "End Game" label is set to indicate to all players that the game is close to finish.

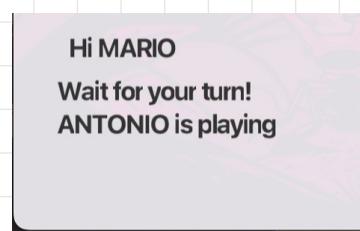
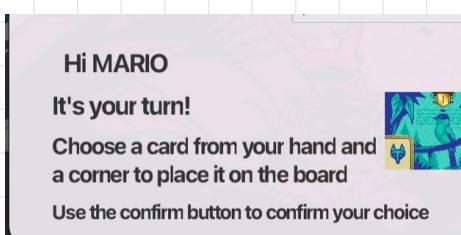
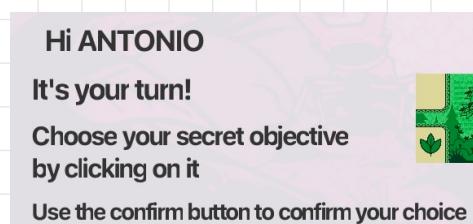
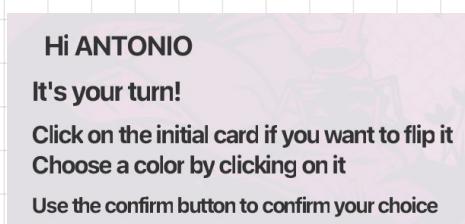
When the game ends, the score board is updated with game points + objectives points, that can be seen if user clicks the "see game" button present in the End Game Scene. In this way, the user is able to explicitly see how many points he or she obtained with the objectives.

End Game!	
Players	Score
MARIO	22+8
ANTONIO	28+13

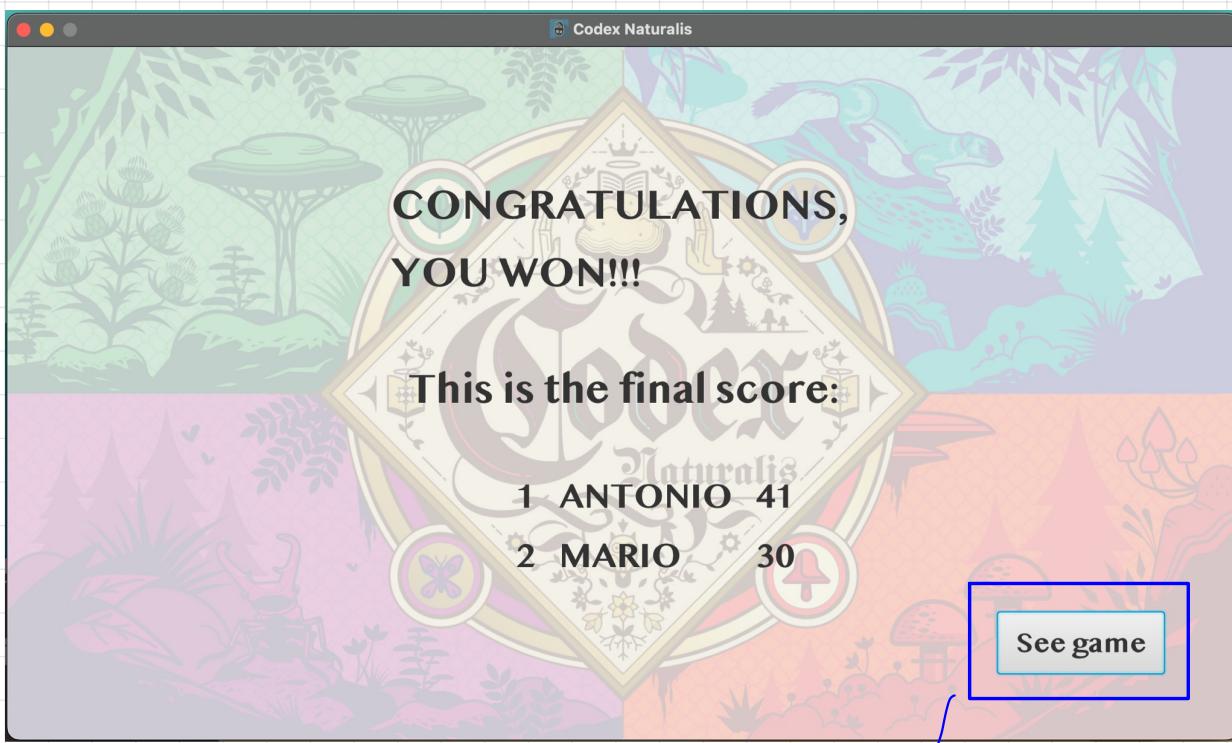
- The **Chat** button opens the chat scene to be able to interact with other players. When a message arrives to a player, if he or she does not have the chat open, the Chat button is updated with a blue border to indicate to the player that there is a new message in chat:



- The Confirm button is activated only when the player has completed their turn and is responsible for sending the choices to the server
- The labels in the left bottom of the scene are updated to help the player to correctly finish their turn or to tell the player to wait and notify him about who is currently playing.



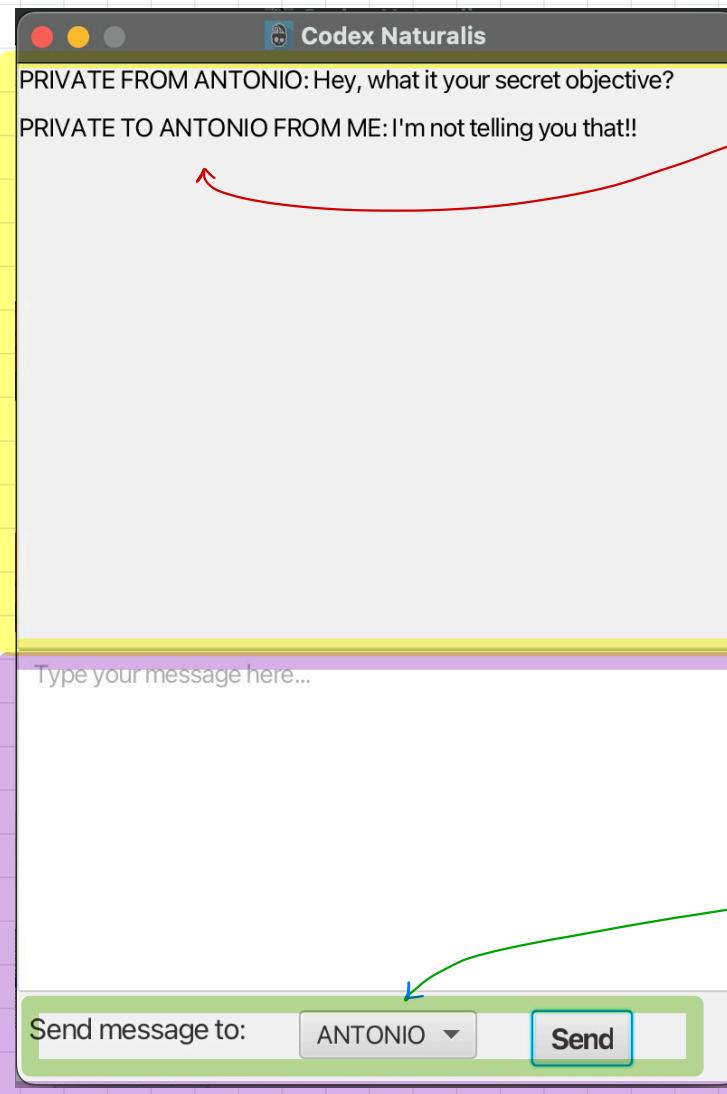
- End Game Scene : WINNER



Allows player to go back and see their game and their adversary's game and the updated score points counted.

If player loses, the name appears with a "Sorry, you lost :(" label.

- Chat GUI Scene : contains a scroll pane with a VBox to act as the **messages container**, a VBox to act as the **input container** and hold the TextArea for the messages to be sent and the **Send Options Container**, that is an HBox composed of the "Send message to :" label, the receiver Choice Box and the "Send" button.



Informs if the message is private or public (in case of games with more than 2 players), the sender and the receiver of the message

Allows players to select who they want to send the message to