Final Project: Super Tic-Tac-Toe

This game is similar to ordinary Tic-Tac-Toe but in each section, there is mini Tic-Tac-Toe in it. So, there are 9 sections in the table and 9 small cells in each and there are special items in some cells.

1.Gameplay:

1.1. Players can choose any cells in the table in the first round.

1.2. In the second round, let's say player1 chooses the top-left **local** cell in the middle **global** cell. Player2 must play in the top-left **global cell** of the table, choose any local cells in that global cell.

1.3. This repeats until one gets Tic-Tac-Toe in any section and that section belongs to that player.

1.4. Repeat until the table is full or the player who got **Tic-Tac-Toe** in the table wins.

1.5. If the player has to play in the cell that is already **Tic-Tac-Toe**, the player can choose any cell in the table.

1.6. There are special items in some **local** cells.

1.7. Some items can be upgraded, player can upgrade those items via crafting menu by using 2 of those items or use that item combined with **Crafting Catalyst**.

Special item:

item name [amount]

1.Magic Eraser (Upgradable) [5]

Normal: Chose a local cell to erase (erased cell will be white, and can be placed again)

Upgraded: Choose a global cell to erase (all the cell in global cell will be white, and can be placed again)

2. Double Spell(Upgradable) [4]

Normal: Allow player to play two times

Upgraded: Allow player to play three times

3.Ice Blocks(Upgradable) [4]

Normal: Allow player to place 4 ice blocks anywhere on the map, that ice block will last 3 rounds before melting. Players cannot place on ice blocks.

Upgraded: Ice block last for 7 rounds

4.Destroyer Ray [2]

Delete every items in the opponent's inventory.

5.Tie Breaker [3]

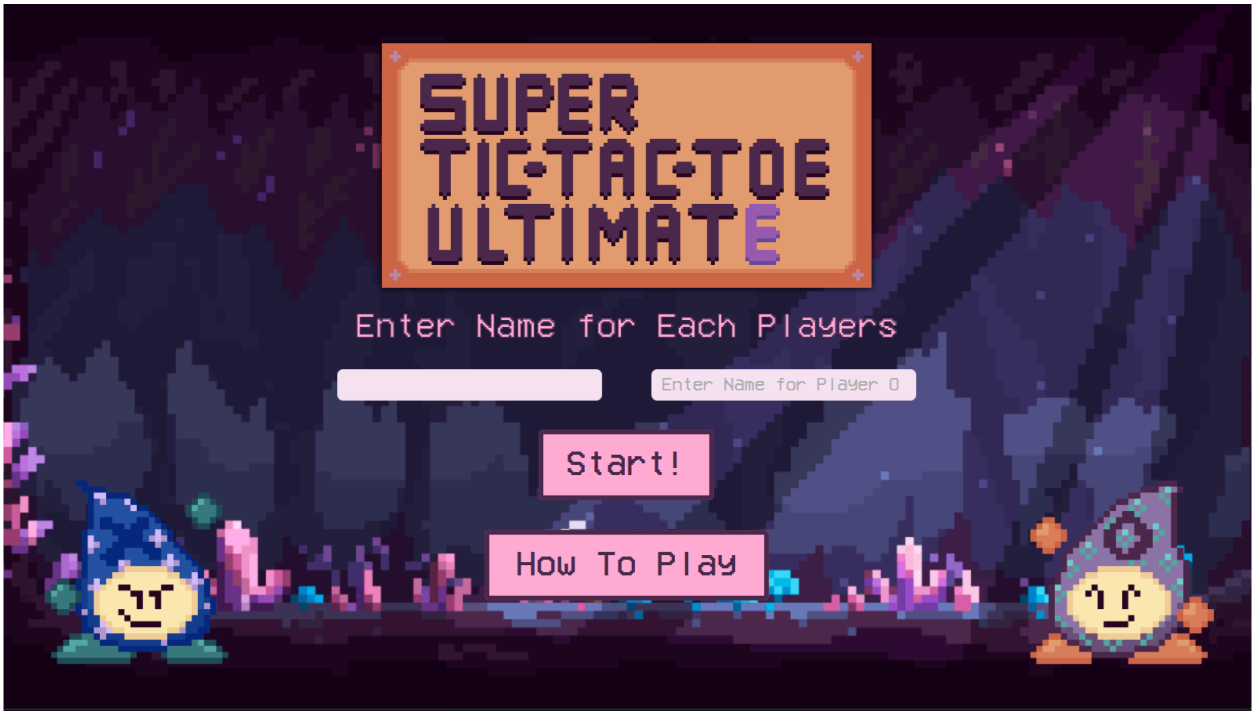
If a global cell has been tied, use this item to break the tie.

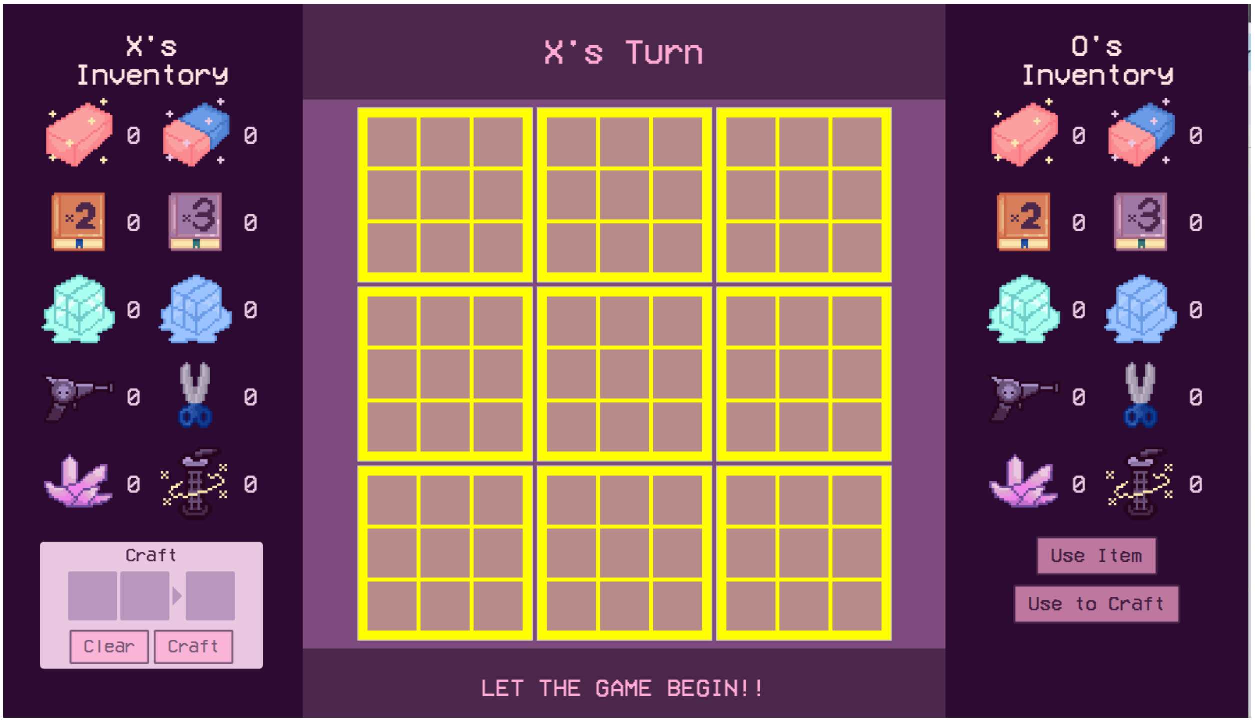
6.Crafting Catalyst [4]

Can be used to upgrade item that is upgradable

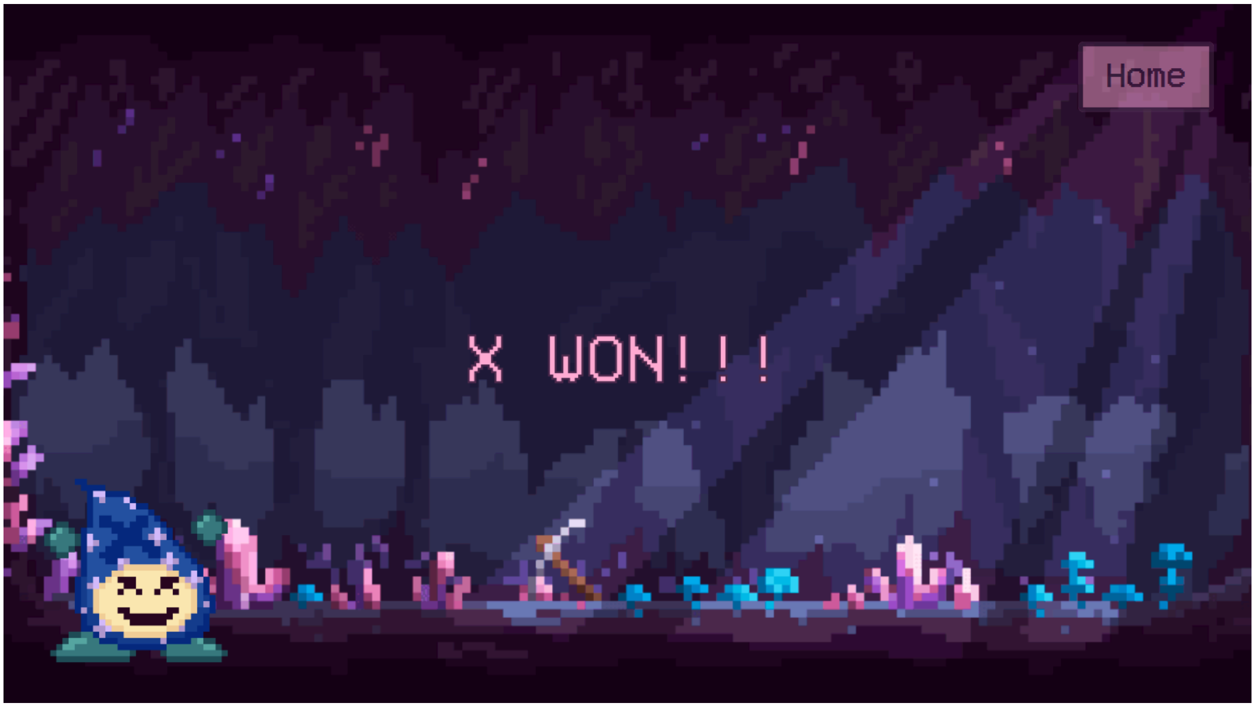
7.Stunt Grenade [4]

Opponent cannot use the item in the next round.

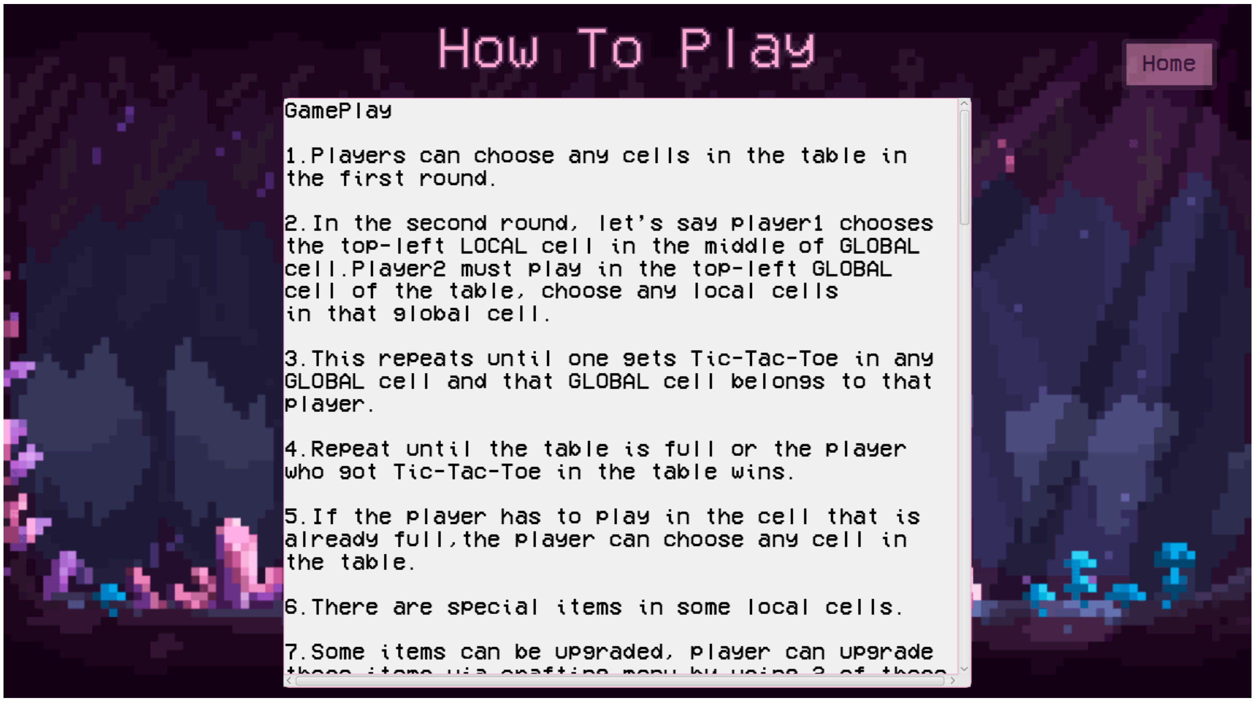




LET THE GAME BEGIN!!



X WON!!!



GamePlay

1.Players can choose any cells in the table in the first round.

2.In the second round, let's say player1 chooses the top-left LOCAL cell in the middle of GLOBAL cell.Player2 must play in the top-left GLOBAL cell of the table, choose any local cells in that global cell.

3.This repeats until one gets Tic-Tac-Toe in any GLOBAL cell and that GLOBAL cell belongs to that player.

4.Repeat until the table is full or the player who got Tic-Tac-Toe in the table wins.

5.If the player has to play in the cell that is already full,the player can choose any cell in the table.

6.There are special items in some local cells.

7.Some items can be upgraded, player can upgrade those items via crafting menu by using 3 of those

UML:
https://github.com/2110215-ProgMeth/project-cp-2023-2-hairynballs653/blob/741f079e9a1684d09c4c26151a88d86000eec069/Final-Project/UML+GUI.png

2.Implement Detail

+ (public)

# (protected)

- (private)

underlined (static)

ALL_CAPS (final variable)

Italic (abstract)

2.1 Package Base

2.1.1 Abstract Class BaseItem

This class generalizes all items in the game.

Fields

Name	**Description**
- String name	Name of item

Methods

Name	**Description**
+ BaseItem(String name)	Constructor.
+ void setName(String name)	Setter of name.If name is null,set name to be “Unnamed item”
+ *String toString()*	Abstract method that return the name of item
+ *void use()*	Abstract method for using item
+ Object clone() throws CloneNotSupportedException	Clone object
+ void *updateInventoryPaneUI(InventoryPane inventoryPane, int count)*	Update player’s inventory UI

2.2. Package gui

2.2.1 class ControlPane extends HBox

This class is the main pane of the game

Fields

Name	Description
- Text gameText	Text that tells whether it is O’s turn or X’s turn
- TicTacToeGlobal ticTacToeGlobal	Global tictactoe pane

Methods

Name	Description
+ ControlPane(TicTacToeGlobal ticTacToeGlobal)	constructor and pane setup
- void initializeGameText()	Initialize game text
+ updateGameText(String text)	Update text for new turn
- initializeNewGame()	Initialize new game
+ TicTacToeGlobal getTicTacToeGlobal()	Getter for ticTacToeGlobal

2.2.2 class CraftingPane extends VBox

Fields

Name	Description
- ItemBox item1Box	First box for crafting item
- ItemBox item2Box	Second box for crafting item
- ItemBox resultBox	Result of crafting
- Button craftBtn	Button for crafting
- Button clearBtn	Button for clear the crafting box
- Arraylist<BaseItem> ingredients	Initialize new Arraylist of basitem

Method

Name	Description

+ CraftingPane()	constructor and pane setup
+ void addIngredients(BaseItem item)	Add item to crafting slot
+ onClearClicked()	Clear the crafting slot
+ void updateResultUI()	Update the UI for crafting result
+ void onCraftHandler()	Item crafting handler
+ void clearItemBox()	Clear item in the boxes
+ void initiateCrafting(BaseItem item, String PlayerName)	Initiate the information when player crafts item
+ ArrayList<BaseItem> getIngredients	Getter for ingredients

2.2.3 class InfoPane extends HBox

This class is the pane that tell if player found item or not.

Fields

Name	**Description**
- Text infoText	Information text

Method

Name	**Description**
+ InfoPane()	constructor and pane setup
- void initializeInfoText()	Initialize text
+ setter for infoText	-

2.2.14 class HowToPlayPane

Fields

Name	Description
- Button menuButton	Back to main screen button
- ScrollPane scrollPane	Scroll pane
- Text infoText	How to play the game text
- Text for every item	-
- ImageView imageview for every item	-

Methods

Name	Description
+ HowToPlayPane()	constructor and pane setup
+ getter for menuButton	-

2.2.5 class InventoryPane extends GridPane

This class is the inventory pane for each player

Fields

Name	Description
- int count for every item(included upgraded item)	-

- Text text for count	-
- ImageView imageview for every item(included upgraded item)	-

Methods

Name	Description
+ InventoryPane()	constructor and pane setup
+ getter/setter for every fields	-

2.2.6 class ItemBox extends StackPane

Fields

Name	Description
- ImageView imageview for every item(included upgraded item)	-

Methods

Name	Description
+ ItemBox()	constructor and pane setup
+ void appear(BaseItem item)	Item appear in the box
+ void disappear(BaseItem item)	Remove item from the box
+ getter/setter for every imageview fields	-

2.2.7 class MenuPane extends StackPane

Fields

Name	Description
- Button startBtn	Start button
- Button howToPlayBtn	Instruction button
- String playerX	String for image of X
- String playerO	String for image of O
- String titleURL	String for image of title
- String bgURL	String for image of background
- VBox centerPane	Base Pane for other pane
- HBox inputPane	Pane for input the name of player
- TextField xInput	Input for X’s name
- TextField oInput	Input for O’s name
- String xName	Name that player input for X
- String oName	Name that player input for O

Methods

Name	Description
+ MenuPane()	constructor and pane setup
+ EvenHandler<KeyEvent> maxLength(final Integer i)	Method for handle input’s max length
+ getter/setter for every field	-

2.2.8 class OPlayerPane extends VBox

Fields

Name	Description
- Text titleText	O’s title text
- InventoryPane oInventory	O’s inventory
- Button useBtn	Item using button in the game
- Button addCraftBtn	Item crafting button in the game
- String oName	“O”

Methods

Name	Description
+ OPlayerPane()	constructor and pane setup
+ getter/setter for all fields	-

2.2.9 class TicTacToeCell extends Pane

Fields

Name	Description
- boolean isDrawn	Check the cell that is already drawn or not
- Color baseColor	Base color
- int xPosition	Local’s position in x-axis
- int yPosition	Local’s position in y-axis
- int xBoard	Global’s position in x-axis
- int yBoard	Global’s position in y-axis

- String oURL	String for O’s url
- String xURL	String for X’s url

Methods

Name	Description
+ TicTacToeCell(int BoardX, int BoardY, int cellX, int cellY)	constructor and pane setup(include handler for click)
- void onClickHandler()	Handle clicking,when click,the image is drawn on the cell
- void draw(Image image, Color backgroundColor)	Draw image
+ void initializeCellColor()	Initialize cell’s color
+ getter/setter for every fields	-

2.2.10 class TicTacToeGlobal extends GridPane

Fields

Name	Description
- ArrayList<TicTacToePane> allPanes	All 9 panes of global cell

Methods

Name	Description
+ TicTacToeGlobal()	constructor and pane setup
+ getter for allPanes	-

2.2.11 class TicTacToePane extends StackPane

Fields

Name	**Description**
- ArrayList<TicTacToeCell> allCells	All 9 cells in each global cell
- Rectangle xRect	X’s cell
- Pane xPane	X’s pane
- Rectangle oRect	O’s cell
- Rectangle tieRect	Tie cell
- GridPane pane	Base pane

Method

Name	**Description**
+ TicTacToePane(int x, int y)	constructor and pane setup
+ void initializeRect(Rectangle rect, Color color)	Initialize the rectangles in the pane
+ void resetLocalBoard(int x, int y)	Reset the global cell

2.2.12 class WinPane extends StackPane

Fields

Name	**Description**
- Text winText	Win text

- Text playerWinnerText	Winner text
- ImageView playerXAnims	X’s GIF
- ImageView playerOAnims	O’s GIF
- ImageView tieAnims	Tie GIF
- Button menuButton	Menu button

Methods

Name	**Description**
+ WinPane()	Constructor and pane setup
+ void toggleXWin, toggleOWin, toggleTie (boolean bool)	Toggle the animation when game end
+ getter/setter for every fields	-

2.2.13 class XPlayerPane extends VBox

Fields

Name	**Description**
- Text titleText	X’s title text
- InventoryPane xInventoryUI	X’s inventory
- CraftingPane craftingPane	Crafting pane
- String xName	“X”

Methods

Name	**Description**
+ XPlayerPane()	constructor and pane setup

+ getter for every fields	-

2.3 Package Interface

2.3.1 interface Upgradeable

Methods

Name	Description
void upgrade()	Upgrade item
void setUpgraded(boolean upgradedState)	Setter for upgradeable item
boolean isUpgraded()	Getter for upgradeable item

2.4 Package item

2.4.1 class CraftingCatalyst extends BaseItem

Methods

Name	Description
+ CraftingCatalyst()	constructor
+ String toString()	Return name of item
+ void use()	useitem
+ void updateInventoryUI(InventoryPane inventoryPane, int count)	Update inventory UI

2.4.2 class DestroyerRay extends BaseItem

Methods

Name	Description
+ DestroyerRay()	constructor
+ String toString()	Return name of item
+ void use()	Use item
+ void updateInventoryUI(InventoryPane inventoryPane, int count)	Update inventory UI

2.4.3 class DoubleSpell extends BaseItem implements Upgradeable

Fields

Name	Description
- boolean Upgrade	-

Methods

Name	Description
+ DoubleSpell()	constructor
+ void upgrade()	Upgrade item
+ void setUpgradeable(boolean upgrade)	Setter for upgrade
+ boolean isUpgraded()	Getter for upgrade
+ String toString()	Return name of item
+ void use()	Use item
+ void updateInventoryUI(InventoryPane inventoryPane, int count)	Update inventory UI

2.4.4 class IceBlocks extends BaseItem implements Upgradeable

Fields

Name	Description

- boolean Upgrade	-

Methods

Name	Description
+ IceBlocks()	constructor
+ void upgrade()	Upgrade item
+ void setUpgraded(boolean upgradeState)	Setter for upgrade
+ boolean isUpgraded()	Getter for upgrade
+ String toString()	Return name of item
+ void use()	Use item
+ void updateInventoryUI(InventoryPane inventoryPane, int count)	Update inventory UI

2.4.5 class MagicEraser extends BaseItem implements Upgradeable

Fields

Name	Description
- boolean Upgrade	-

Methods

Name	Description
+ MagicEraser()	constructor

+ void upgrade()	Upgrade item
+ void setUpgraded(boolean upgradedState)	Setter for upgrade
+ boolean isUpgraded()	Getter for upgrade
+ String toString()	Return name of item
+ void use()	Use item
+ void updateInventoryUI(InventoryPane inventoryPane, int count)	Update inventory UI

2.4.6 class PackedIceBlocks extends BaseItem

Same fields and methods as IceBlocks.

2.4.7 class StuntGrenede extends BaseItem

Methods

Name	**Description**
+ StuntGenede()	constructor
+ String toString()	Return name of item
+ void use()	Use item
+ void updateInventoryUI(InventoryPane inventoryPane, int count)	Update inventory UI

2.4.8 class TieBreaker extends BaseItem

Methods

Name	Description
+ TieBreaker()	constructor
+ String toString()	Return name of item
+ use()	Use item
+ void updateInventoryUI(InventoryPane inventoryPane, int count)	Update inventory UI

2.4.9 class TripleSpell extends BaseItem

Same fields and methods as class DoubleSpell.

2.4.10 class UpgradedMagicEraser extends BaseItem

Same fields and methods as class MagicEraser.

2.5 Package logic

2.5.1 class GameLogic

Fields

Name	Description
- GameLogic instance	Instance of GameLogic
- ControlPane controlPane	-
- InfoPane infoPane	-
- XPlayerPane xPlayerPane	-

- OPlayerPane oPlayerPane	-
- boolean canDraw	Check if the cell can be drawn
- int[][][][] board	4D array store data of whole board
- int[][][][] lootTable	4D array store item of whole board
- int[][] globalBoard	2D array store data of global board
- char currentPlayer	Current player
- int LastRoundR	Location row of last round
- int LastRoundC	Location column of last round
- boolean gameEnd	Check if the game end
- boolean isFilled	Check if the cell is already filled
- boolean isFound	Check if player found item
- int finalGameState	-
- HashMap<String, Integer> OplayerItem, XplayerItem	Hashmap store item for each player
- ArrayList<BaseItem> oInventory, xInventory	Arraylist store item for each player
- Player playerO, playerX	Each player
- BaseItem selectedXItem, selectedOItem	Item that player select
- boolean roundItemUsed	Check if player use item
- int stuntMove	Player get stunted or not
- int moveNumber	Number of round
- int multipleTurns	Player get to play multiple rounds or not
- boolean tieBreaker	Check if player use tiebreaker or not
- boolean erasePhase	Check erasing phase
- upgradedErasePhase	Check upgraded erasing phase
- int frozenRoundLeft	Number of rounds left for double or triple spell

- int iceBlocksLeft	Number of rounds left for iceblocks

Methods

Name	**Description**
GameLogic()	Initialize new game method
+ void newGame()	Initialize the game
+ void drawNumber(int boardX, int boardY, int cellX, int cellY)	Store number data in board
+ void fillLoot()	Fill item in board
+ void clearLootTable()	Clear item in board
+ boolean getItem(int boardX, int boardY, int cellX, int cellY)	Check if player found item
+ void updateHighlight()	Highlight the global cell that player can play
+ void removeHighlight()	Remove highlight
+ void updateLocalUI(int boardX, int boardY, int num)	Update UI in local cell
+ boolean checkLastRound(int boardX, int boardY)	Check if chosen cell is the same as LastRound cell or not
+ boolean checkValidMove(int boardX, int boardY, int cellX, int cellY)	Check if the move is valid or not
+ boolean checkFull(int boardX, int boardY)	Check if the global cell full or not
+ void initiatingWin()	Check who win.If finalGameState is 1,X wins.If finalGameState is 2,O win.If finalGameState is -1, tie.
+ boolean checkGlobalFull()	Check if the board fill or not
+ boolean checkLocalWinner()	Check the winner of global cell
+ boolean checkGlobalWin()	Check the winner of the game
+ getter/setter for every fields	-

2.5.2 class GameUtils extends Thread

Methods

Name	Description
+ int paneIndexConverter(int boardX, int boardY)	Convert pane position to integer
+ int[] reversePaneIndexConverter(int i)	Reverse paneIndexConverter
+ void fillLootTable(int[][][][] lootTable)	Fill the loot randomly
+ void updateInventoryUI()	Update UI for inventory
+ void removeItemFromInventory(BaseItem item)	Remove item from inventory
+ void removeInventoryUI(char turn)	Update removing item in inventory
+ void updateCertainInventoryUI(char playerTurn, BaseItem item)	Update current UI in inventory
+ void updateInventoryUI()	Update UI in inventory
+ boolean canRemove(BaseItem item)	Check if the item can be removed or not
+ void undoTake(BaseItem item)	Undo the selection of item
+ boolean isCraftable	Check if the item can be crafted or not
+ void addCraftedItem(Baseitem item)	Add crafted item to inventory

2.5.3 class ItemMethods

Fields

Name	Description
+ ItemMethods instance	Instance of class

Methods

Name	Description

+ void eraseLocalPane()	Method for magic eraser(remove item in local cell)
+ void eraseGlobalPane()	Method for upgraded magic eraser(remove item in global cell)
+ void doubleRound()	Method for double spell(player play for 2 rounds)
+ void tripleRound()	Method for triple spell(player play for 3 rounds)
+ void blockCell(int rounds)	Method for ice block(block the local cell)
+ void DestroyItem()	Method for Destroyer Ray(remove item in opponent’s inventory)
+ void tieBreaking()	Method for tie breaker(break the tie global cell)
+ void stuntPlayer()	Method for stunt grenade(disabled opponent’s item)
+ ItemMethods getInstance()	Getter for instance

2.5.4 class UIConstants

Constant for UI

2.5.5 class UseItemHandler

Methods

Name	**Description**
+ void useItem()	Handling item used by player

2.6 Package Player

2.7.1 class Player

Fields

Name	Description
- String name	Name of player
- ArrayList<BaseItem> items	Inventory of player

Methods

Name	Description
+ Player()	constructor
+ getter/setter for every fields	-

2.6 Package main

2.6.1 class Main extends Application

Main class for the game