**Programming Methodology Documentation**

Pregmoth Treasure Hunt

**Created by**

Pattaradanai Thanomsittikul 6633185721

Naphat Chartwanchai 6633059321

2110215 Programming Methodology

Semester 2 Year 2023

Chulalongkorn University

**Pregmoth Treasure Hunt**

1. **Introduction**

"Pregmoth Treasure Hunt" is a game where you embark on a quest to uncover hidden treasures by conquering each level. In this strategic adventure, your objective is to maneuver objects to specific targets through careful planning and skillful execution. If you successfully navigate through all the levels, you'll be rewarded with the ultimate treasure hunt experience.

1. **Rules**

2.1 Objective

The objective of the game is to strategically move objects (pieces) onto designated target.

2.2 Player Moves

* The player can move one square in the up, down, left, or right direction.
* The player can push one object at a time, but only if there is an empty square (or a target square) next to the object in the direction of movement.
* Objects cannot be pulled. They can only be pushed into an adjacent empty space.

2.3 Game End

The game ends when all objects are successfully pushed onto the target squares, completing the level. Alternatively, the game ends if the player makes a move that prevents further progress (e.g., pushing an object into a corner where it

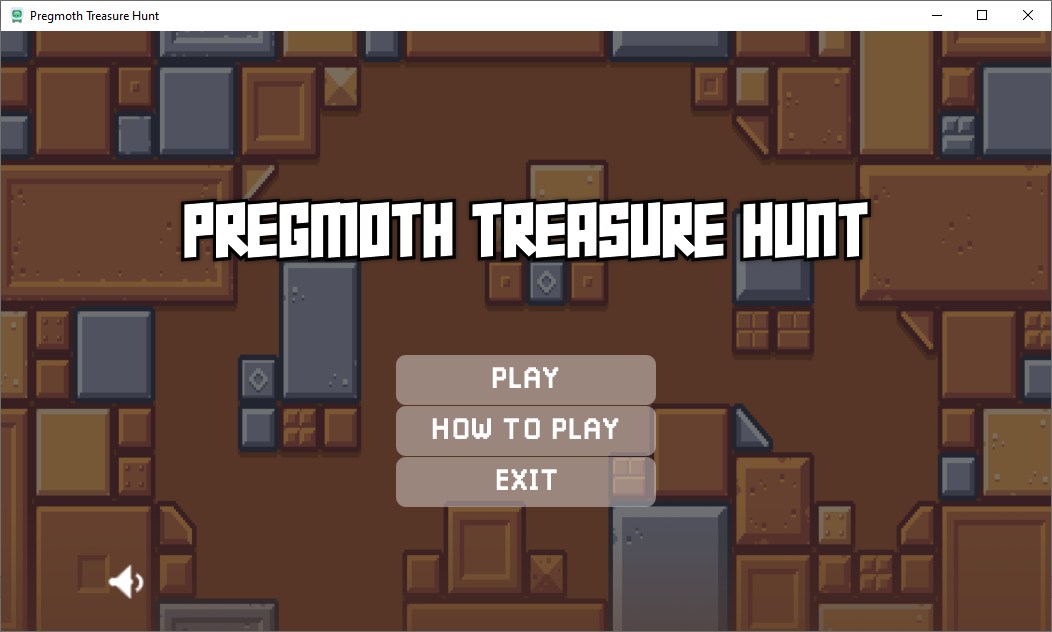
cannot be moved) or the time is running out.

2.4 Project presentation

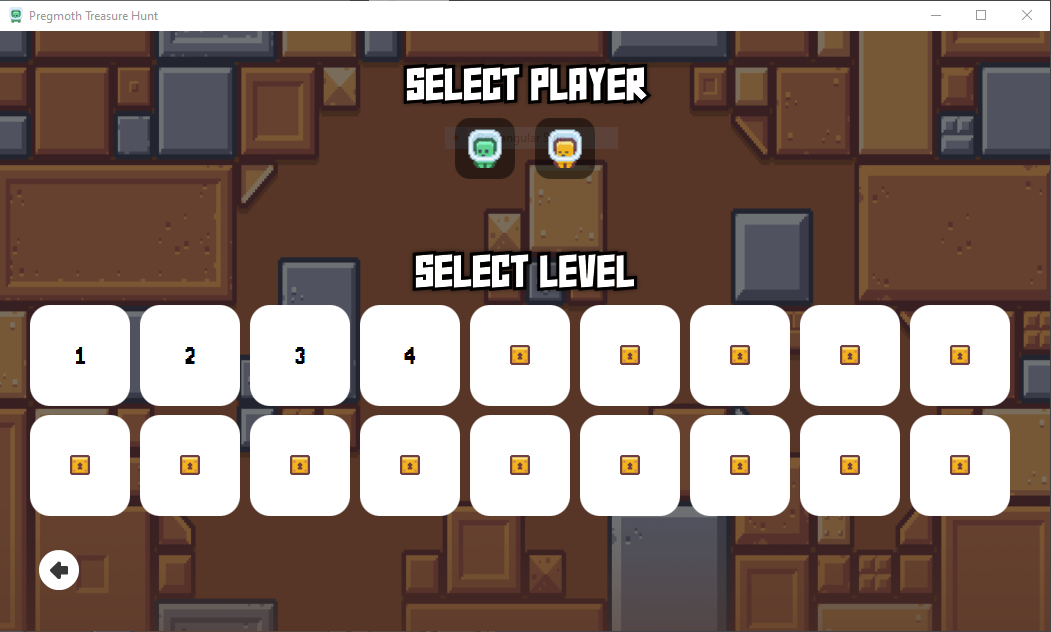
Youtube : https://youtu.be/eE0ky1-deC8?si=FYr77hdA2fobeEoP

1. **Example**

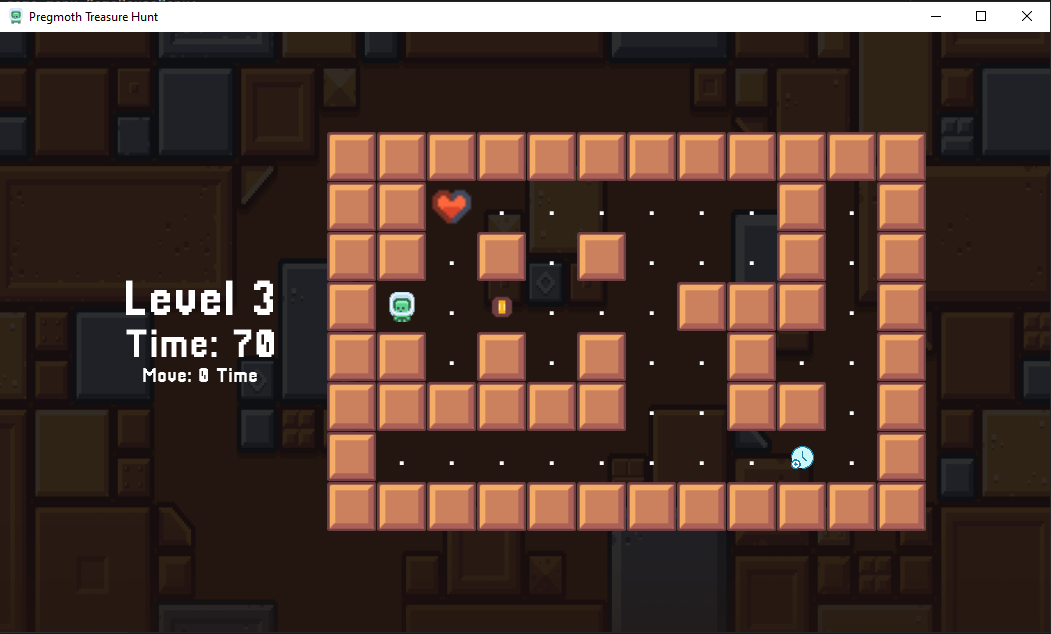
**3.1 Main menu scene**

****

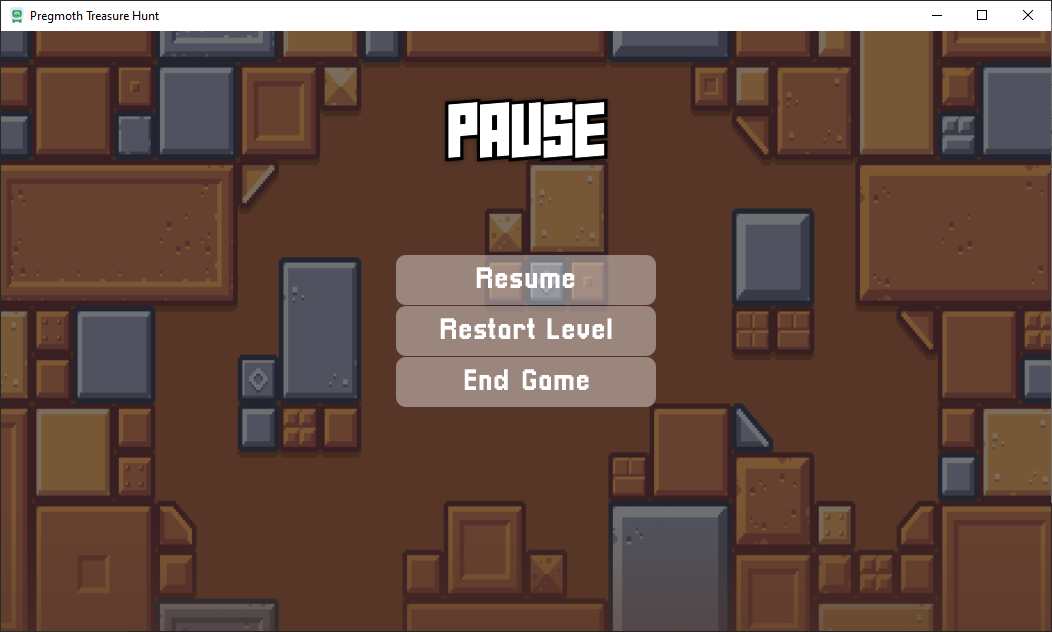
**3.2 Select level and player menu**

****

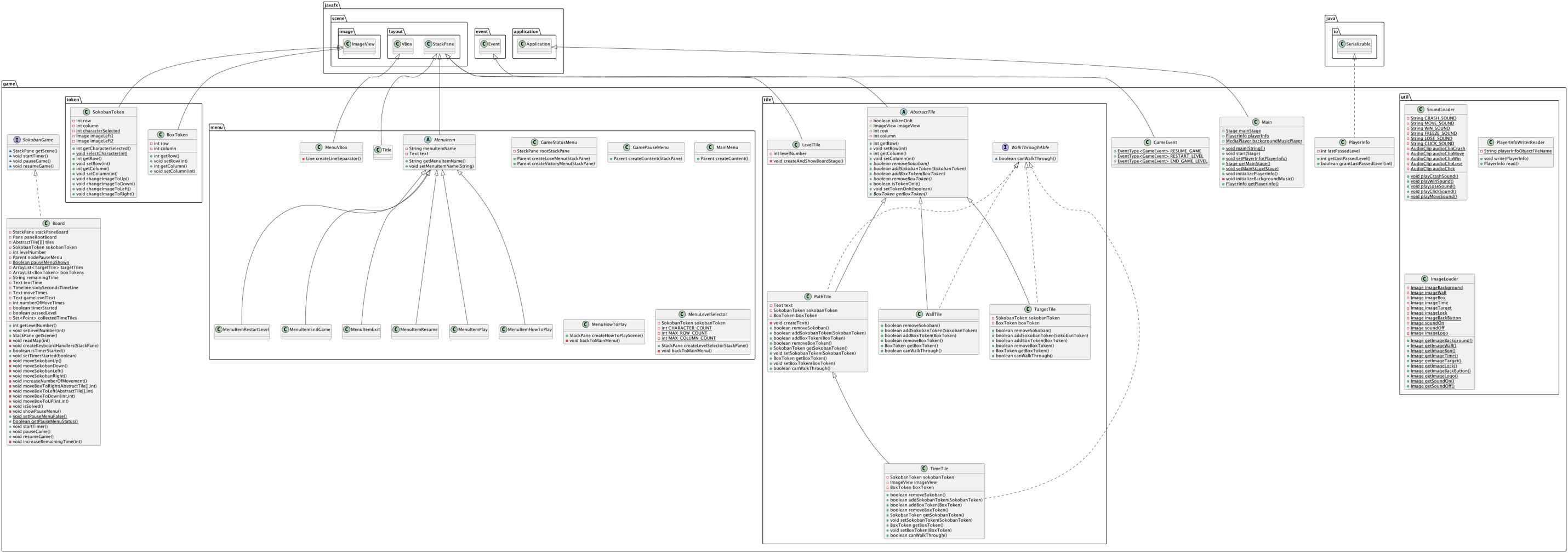
**3.3 Game scene**

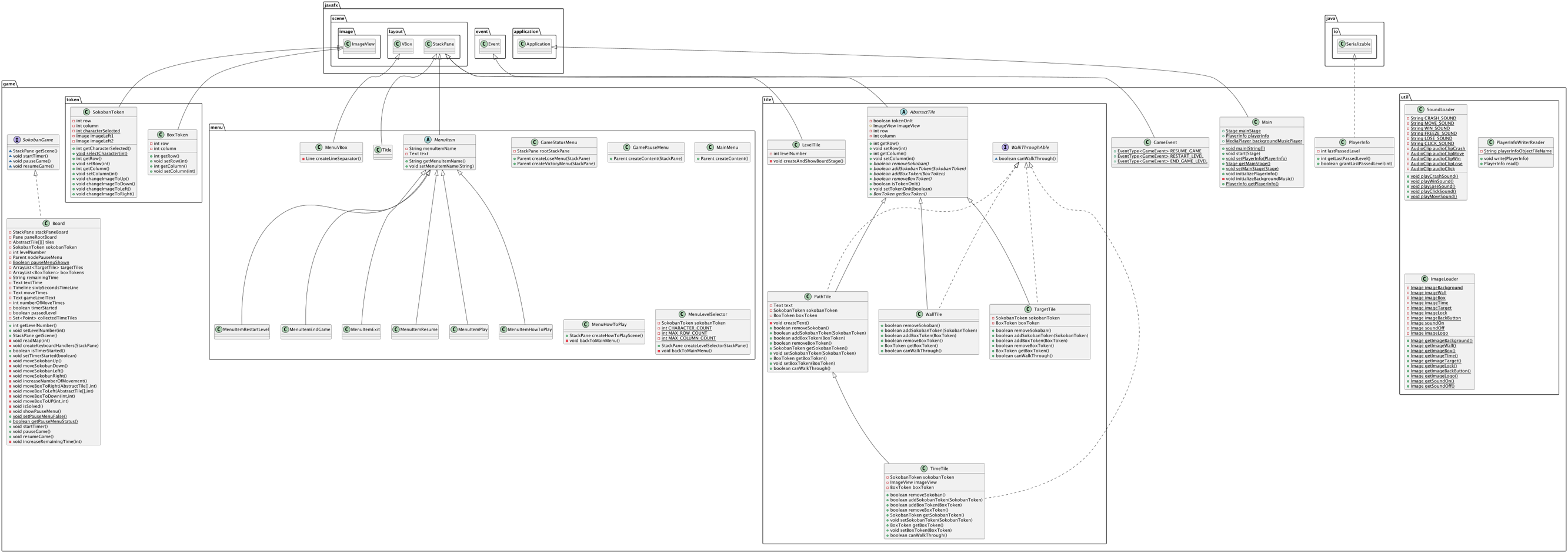
****

**3.4 Menu Pause**

****

1. **Class Diagram**

****

****

1. **Implementation Detail**
   1. **Package game**
      1. Class Main extends Application

***Field***

|  |  |
| --- | --- |
| Name | Description |
| + Stage mainStage | The main stage of application. |
| + PlayerInfo playerInfo | The player’s information. |
| + MediaPlayer backgroundMusicPlayer | The background’s music of application |

***Constructor***

|  |  |
| --- | --- |
| Name | Description |
| + void main (String[] args) | Launch the JavaFX application |

***Method***

|  |  |
| --- | --- |
| Name | Description |
| + void start (Stage primaryStage) | - Initialize player information, background music  - Set application’s icon, title and mouse cursor  - Create a MainMenu scene  - Set to main stage |
| + void initializePlayerInfo () | - Initialize player information by reading from class PlayerInfoWriterReader, if no existing player information creates a new PlayerInfo |
| + void initializeBackgroundMusic () | Create Media sound and set background music |
| + void PlayerInfo getPlayerInfo () | Get player’s info from PlayerInfo |
| + getMainStage () | Return mainStage |
| + void setMainStage (Stage mainStage) | Set mainStage |
| + void setPlayerInfo (PlayerInfo playerInfo) | Set playerInfo |

* + 1. Class Board implements SokobanGame

***Field***

|  |  |
| --- | --- |
| Name | Description |
| - StackPane stackPaneBoard | The main stack pane for the game board UI. |
| - Pane paneRootBoard | The root pane containing the game board tiles. |
| - AbstractTile[][] tiles | A 2D array representing the game board tiles. |
| - SokobanToken sokobanToken | The token representing the player character on the board. |
| - int levelNumber | The level number of the current game board. |
| - Parent nodePauseMenu | - The node representing the pause menu UI.  - Initialize with null |
| - boolean pauseMenuShown | - Indicates if the pause menu is currently shown.  - Initialize with false |
| - ArrayList<TargetTile> targetTiles | A list of target tiles representing the goal positions for boxes. |
| - ArrayList<BoxToken> boxTokens | A list of box tokens representing movable boxes on the board. |
| - String remainingTIme | - The remaining time left in the game, represented as a string.  - Set to “70” |
| - Text textTime | - The text node displaying the current remaining time on the UI.  - Set to “Time: “ |
| - TimeLine sixtySecondsTimeLine | The timeline for the game timer. |
| - Text moveTimes | - The text node displaying the number of moves made by the player.  - Set to “Move: 0 Time” |
| - Text gameLevelText | The text node displaying the current game level on the UI. |
| - int numberOfMoveTimes | - The number of moves made by the player.  - Set to 0 |
| - boolean timeStarted | - Indicates if the game timer has been started.  - Initialize with false |
| - boolean passedLevel | - Indicates if the current level has been completed.  - Initialize with false |

***Constructor***

|  |  |
| --- | --- |
| Name | Description |
| + Board (int levelNumber) | - Initializes a new instance of the Board class with the specified levelNumber.  - Set the levelNumber field of the Board instance.  - Read map from levelNumber to populate the game board.  - Set Text to to display the current level number on the game board UI. |

***Method***

|  |  |
| --- | --- |
| Name | Description |
| + StackPane getScene () | - Constructs and returns the graphical scene for the game board.  - Setting up the game UI elements, handling key events, and managing game state. |
| - void readMap (int levelNumber) | - Reads the map data for a specified level from a text file (test<levelNumber>.txt).  - Constructs the game board. |
| - void createKeyboardHandlers (StackPane scene) | - Sets up keyboard event handlers for controlling the player's movement by using arrow keys.  - Handling the pause menu activation. |
| + boolean isTimerStarted () | Return timerStarted. |
| + void setTimerStarted (boolean timerStarted) | Set timerStarted. |
| - void moveSokobanUp () | - Handle the movement of the Sokoban token in up directions.  - Handle interactions with game tiles (e.g., walls, boxes, targets) during movement. |
| - void moveSokobanDown () | - Handle the movement of the Sokoban token in down directions.  - Handle interactions with game tiles (e.g., walls, boxes, targets) during movement. |
| - void moveSokobanLeft () | - Handle the movement of the Sokoban token in left directions.  - Handle interactions with game tiles (e.g., walls, boxes, targets) during movement. |
| - void moveSokobanRight () | - Handle the movement of the Sokoban token in right directions.  - Handle interactions with game tiles (e.g., walls, boxes, targets) during movement. |
| - void increaseNumberOfMovement () | - Increments the count of Sokoban token movements and updates the display text accordingly. |
| - void moveBoxToRight (AbstractTile [] tile, int sokobanColumn) | - Handle moving boxes based on the right movement of the Player.  - Check for valid moves and interactions with other game tiles. |
| - void moveBoxToLeft (AbstractTile [] tile, int sokobanColumn) | - Handle moving boxes based on the left movement of the Player.  - Check for valid moves and interactions with other game tiles. |
| - void moveBoxToDown (AbstractTile [] tile, int sokobanColumn) | - Handle moving boxes based on the down movement of the Player.  - Check for valid moves and interactions with other game tiles. |
| - void moveBoxToUp (AbstractTile [] tile, int sokobanColumn) | - Handle moving boxes based on the up movement of the Player.  - Check for valid moves and interactions with other game tiles. |
| - void isSolved () | - Checks if the level has been successfully solved  - If solved, it stops the game timer and displays a victory menu. |
| - void showPauseMenu () | - Displays the pause menu when the ESCAPE key is pressed during gameplay. |
| + void setPauseMenuFalse () | Set pauseMenuShown to false. |
| + boolean getPauseMenuStatus () | Return pauseMenuShown. |
| + void startTimer () | - Starts the game timer  - Handles game over conditions when time runs out or a maximum number of moves is reached. |
| + int getLevelNumber () | Get levelNumber. |
| + void setLevelNumber  (int levelNumber) | Set levelNumber. |
| - void increaseRemainingTime (int seconds) | Increase remaining time with specified seconds. |

* + 1. Class GameEvent extends Event

***Field***

|  |  |
| --- | --- |
| Name | Description |
| + EventType<GameEvent> RESUME\_GAME | Represents the event type for resuming the game. |
| + EventType<GameEvent> RESTART\_LEVEL | Represents the event type for restarting the current game level. |
| + EventType<GameEvent> END\_GAME\_LEVEL | Represents the event type for ending the current game level. |

***Method***

|  |  |
| --- | --- |
| Name | Description |
| + GameEvent (EventType<? Extends Event> eventType | Initializes a GameEvent object with a specified EventType. |

* + 1. Class PlayerInfo implements Serializable

***Field***

|  |  |
| --- | --- |
| Name | Description |
| - int lassPassedLevel | - Get the lassPassedLevel from player’s info  - Initialize to 0 |

***Method***

|  |  |
| --- | --- |
| Name | Description |
| + int getLassPassedLevel () | Return lassPassedLevel. |
| + boolean grantLassPassedLevel (int newPassedLevel) | Updates the lastPassedLevel if the provided newPassedLevel is greater than the current lastPassedLevel. |

* + 1. Interface SokobanGame

***Method***

|  |  |
| --- | --- |
| Name | Description |
| StackPane getScene () | Get scene. |
| void startTimer () | Start the timer interface. |
| void pauseGame () | Pause game interface. |
| void resumeGame () | Resume game interface. |

* 1. **Package game.menu**
     1. Class MainMenu

***Method***

|  |  |
| --- | --- |
| Name | Description |
| + Parent createContent () | Creating the visual content of the main menu screen. |

* + 1. Abstract Class MenuItem extends StackPane

***Field***

|  |  |
| --- | --- |
| Name | Description |
| - String menuItemName | Holds the name or label associated with the menu item. |
| - Text text | Represents the graphical text element associated with the menu item. |

***Constructor***

|  |  |
| --- | --- |
| Name | Description |
| + MenuItem () | Initializes a graphical UI for MenuItem. |

***Method***

|  |  |
| --- | --- |
| Name | Description |
| + String getMenuItemName () | Return menuItemName. |
| + void setMenuItemName (String menuItemName) | - Set menuItemName.  - Set text to menuItemName. |

* + 1. Class MenuVBox extends VBox

***Constructor***

|  |  |
| --- | --- |
| Name | Description |
| + MenuVBox (MenuItem… items) | Initializes a graphical UI for MenuVBox. |

***Method***

|  |  |
| --- | --- |
| Name | Description |
| - Line createLineSeparator () | Create a horizontal line separator with specific properties. |

* + 1. Class Title extends StackPane

***Constructor***

|  |  |
| --- | --- |
| Name | Description |
| + Title (String name, int fontSize) | Constructor for the Title class, which represents a styled text title. |

* + 1. Class GamePauseMenu

***Method***

|  |  |
| --- | --- |
| Name | Description |
| + Parent createContent () | Creating the visual content of the main menu screen. |

* + 1. Class GameStatusMenu

***Constructor***

|  |  |
| --- | --- |
| Name | Description |
| + GameStatusMenu () | Creates the menus for game status. |

***Method***

|  |  |
| --- | --- |
| Name | Description |
| + Parent createLoseMenu (StackPane boardStakePane) | Creates the game menu and display when the player loses. |
| + Parent createVictoryMenu (StackPane boardStakePane) | Creates the game menu and display when the player achieves victory. |

* + 1. Class MenuItemEndGame extends MenuItem

***Constructor***

|  |  |
| --- | --- |
| Name | Description |
| + MenuItemEndGame (StackPane boardStackPane) | Create a menu item for ending the game. |

* + 1. Class MenuItemExit extends MenuItem

***Constructor***

|  |  |
| --- | --- |
| Name | Description |
| + MenuItemExit () | Create a menu item for exiting the game. |

* + 1. Class MenuItemHowToPlay extends MenuItem

***Constructor***

|  |  |
| --- | --- |
| Name | Description |
| + MenuItemHowToPlay () | Create a menu item for how to play the game. |

* + 1. Class MenuItemPlay extends MenuItem

***Constructor***

|  |  |
| --- | --- |
| Name | Description |
| + MenuItemPlay () | Create a menu item for playing the game. |

* + 1. Class MenuItemRestartLevel extends MenuItem

***Constructor***

|  |  |
| --- | --- |
| Name | Description |
| + MenuItemRestartLevel(StackPane boardStackPane) | Create a menu item for restarting the game. |

* + 1. Class MenuItemResume extends MenuItem

***Constructor***

|  |  |
| --- | --- |
| Name | Description |
| + MenuItemResume(StackPane boardStackPane) | Create a menu item for resuming the game. |

* + 1. Class MenuLevelSelector

***Field***

|  |  |
| --- | --- |
| Name | Description |
| - SokobanToken sokobanToken | - Represents a token associated with the game. |
| - int CHARACTER\_COUNT | - The number of character options available for selection.  - Set to 2. |
| - int MAX\_ROW\_COUNT | - Specifies the maximum number of rows for displaying level tiles.  - Set to 3. |
| - int MAX\_COLUMN\_COUNT | - Specifies the maximum number of columns for displaying level tiles.  - Set to 9. |

***Method***

|  |  |
| --- | --- |
| Name | Description |
| + StackPane createHowToPlayScene () | Creates a StackPane representing the "How to Play" scene with instructions and a back button. |
| - void backToMainMenu() | Navigate back to the main menu. |

* + 1. Class MenuHowToPlay

***Method***

|  |  |
| --- | --- |
| Name | Description |
| + StackPane createLevelSelectorStackPane () | Contains the level selector, player selector section in this menu. |
| - void backToMainMenu() | Navigate back to the main menu. |

* 1. **Package game.tile**
     1. Abstract Class AbstractTile extends StackPane

***Field***

|  |  |
| --- | --- |
| Name | Description |
| - boolean tokenOnIt | Indicates the token on the tile. |
| # ImageView imageView | An image view associated with this tile. |
| - int row | Represents the row index of the tile. |
| - int column | Represents the column index of the tile. |

***Constructor***

|  |  |
| --- | --- |
| Name | Description |
| + AbstractTile (int row, int column, double translateX, double translateY) | Constructs an abstract tile with specified row and column indices, along with translation coordinates (translateX and translateY) for positioning. |

***Method***

|  |  |
| --- | --- |
| Name | Description |
| + int getRow () | Retrieves the row index of the tile. |
| + void setRow (int row) | Sets the row index of the tile. |
| + int getColumn () | Retrieves the column index of the tile. |
| + void setColumn (int column) | Sets the column index of the tile. |
| *+ boolean removeSokoban ()* | Abstract method to remove a Sokoban token from the tile. |
| *+ boolean addSokobanToken (SokobanToken sokobanToken)* | Abstract method to add a Sokoban token to the tile. |
| *+ boolean addBoxToken (BoxToken boxToken)* | Abstract method to add a Box token to the tile. |
| *+ boolean removeBoxToken ()* | Abstract method to remove a Box token from the tile. |
| + boolean isTokenOnIt () | Checks if there is a token on the tile. |
| + void setTokenOnIt (boolean tokenOnIt) | Sets whether there is a token on the tile. |
| *+ BoxToken getBoxToken ()* | Abstract method to retrieve the Box token on the tile. |

* + 1. Class LevelTile extends StackPane

***Field***

|  |  |
| --- | --- |
| Name | Description |
| - int levelNumber | Holds the number representing the level associated with this tile. |

***Constructor***

|  |  |
| --- | --- |
| Name | Description |
| + LevelTile (String value,double translateX,double translateY) | - Initializes a LevelTile with the given value  - Sets its translation coordinates. |

***Method***

|  |  |
| --- | --- |
| Name | Description |
| - void createAndShowBoardStage () | - Creates and displays the game board associated with this level tile when the tile is clicked.  - Sets up event filters for game events like END\_GAME\_LEVEL and RESTART\_LEVEL to handle changes in the game state. |

* + 1. Class PathTile extends AbstractPane implements WalkThroughAble

***Field***

|  |  |
| --- | --- |
| Name | Description |
| - Text text | Represents the text element associated with the PathTile, displaying a dot ".". |
| - SokobanToken sokobanToken | Holds the SokobanToken object associated with the PathTile, representing a Sokoban player character. |
| - Boxtoken boxToken | Holds the BoxToken object associated with the PathTile, representing a box token that can be pushed in the game. |

***Constructor***

|  |  |
| --- | --- |
| Name | Description |
| + PathTile (int row, int column, double translateX, double translateY, SokobanToken sokobanToken) | Constructs a PathTile with a SokobanToken at the specified position. |
| + PathTile (int row, int column, double translateX, double translateY, BoxToken boxToken) | Constructs a PathTile with a BoxToken at the specified position. |
| + PathTile (int row, int column, double translateX, double translateY) | Constructs an empty PathTile at the specified position. |

***Method***

|  |  |
| --- | --- |
| Name | Description |
| - void createText () | Initializes the text element for the PathTile. |
| + boolean removeSokoban () | Removes the SokobanToken from the PathTile and updates its state. |
| + boolean addSokobanToken (SokobanToken sokobanToken) | Adds a SokobanToken to the PathTile and updates its state. |
| + boolean addBoxToken (BoxToken boxToken) | Adds a BoxToken to the PathTile and updates its state. |
| + boolean removeBoxToken () | Removes the BoxToken from the PathTile and updates its state. |
| + SokobanToken getSokobanToken () | Returns the SokobanToken associated with the PathTile. |
| + void setSokobanToken (SokobanToken sokobanToken) | Sets the SokobanToken for the PathTile. |
| + BoxToken getBoxToken () | Returns the BoxToken associated with the PathTile. |
| + void setBoxToken (BoxToken boxToken) | Sets the BoxToken for the PathTile. |
| + boolean canWalkThrough () | Sets walk through to true. |

* + 1. Class TargetTile extends AbstractPane implements WalkThroughAble

***Field***

|  |  |
| --- | --- |
| Name | Description |
| - SokobanToken sokobanToken | Represents the Sokoban token associated with the TargetTile, indicating the presence of a Sokoban player character. |
| - BoxToken boxToken | Represents the box token associated with the TargetTile, indicating the presence of a box. |

***Constructor***

|  |  |
| --- | --- |
| Name | Description |
| + TargetTile (int row, int column, double translateX, double translateY) | Represent a target tile in the game. This tile displays an image of a target and can contain either a Sokoban token or a box token. |

***Method***

|  |  |
| --- | --- |
| Name | Description |
| + boolean removeSokoban () | Removes the Sokoban token from the tile if present. |
| + boolean addSokobanToken (SokobanToken sokobanToken) | Adds a Sokoban token to the tile. |
| + boolean addBoxToken (BoxToken boxToken) | Adds a box token to the tile. |
| + boolean removeBoxToken () | Removes the box token from the tile if present. |
| + BoxToken getBoxToken () | Returns the box token associated with the tile. |
| + boolean canWalkThrough () | Sets walk through to true. |

* + 1. Class WallTile extends AbstractPane implements WalkThroughAble

***Constructor***

|  |  |
| --- | --- |
| Name | Description |
| + WallTile (int row, int column, double translateX, double translateY) | Represents a wall tile within the game. Wall tile is impassable and does not allow tokens such as Sokoban tokens or box tokens to be added or removed. |

***Method***

|  |  |
| --- | --- |
| Name | Description |
| + boolean removeSokoban () | Returns false because Sokoban tokens cannot be removed from wall tiles. |
| + boolean addSokobanToken (SokobanToken sokobanToken) | Returns false because Sokoban tokens cannot be added to wall tiles. |
| + boolean addBoxToken (BoxToken boxToken) | Returns false because box tokens cannot be added to wall tiles. |
| + boolean removeBoxToken () | Returns false because box tokens cannot be removed from wall tiles. |
| + BoxToken getBoxToken () | Returns null because wall tiles do not contain box tokens. |
| + boolean canWalkThrough () | Sets walk through to false. |

* + 1. Class TimeTile extends PathTile implements WalkThroughAble

***Field***

|  |  |
| --- | --- |
| Name | Description |
| - ImageView imageView | Represents the time elements image. |
| - SokobanToken sokobanToken | Holds the SokobanToken object associated with the PathTile, representing a Sokoban player character. |
| - Boxtoken boxToken | Holds the BoxToken object associated with the PathTile, representing a box token that can be pushed in the game. |

***Constructor***

|  |  |
| --- | --- |
| Name | Description |
| + TimeTile (int row, int column, double translateX, double translateY) | Constructs an empty TimeTile at the specified position. |

***Method***

|  |  |
| --- | --- |
| Name | Description |
| + boolean removeSokoban () | Removes the SokobanToken from the PathTile and updates its state. |
| + boolean addSokobanToken (SokobanToken sokobanToken) | Adds a SokobanToken to the TimeTile and updates its state. |
| + boolean addBoxToken (BoxToken boxToken) | Adds a BoxToken to the TimeTile and updates its state. |
| + boolean removeBoxToken () | Removes the BoxToken from the TimeTile and updates its state. |
| + SokobanToken getSokobanToken () | Returns the SokobanToken associated with the TimeTile. |
| + void setSokobanToken (SokobanToken sokobanToken) | Sets the SokobanToken for the TimeTile. |
| + BoxToken getBoxToken () | Returns the BoxToken associated with the TimeTile. |
| + void setBoxToken (BoxToken boxToken) | Sets the BoxToken for the TimeTile. |
| + boolean canWalkThrough () | Sets walk through to true. |

* + 1. Interface WalkThroughAble

***Method***

|  |  |
| --- | --- |
| Name | Description |
| + boolean canWalkThrough () | Check tile availability to walk through. |

* 1. **Package game.token**
     1. Class BoxToken extends ImageView

***Field***

|  |  |
| --- | --- |
| Name | Description |
| - int row | Represents the row position of the BoxToken. |
| - int column | Represents the column position of the BoxToken. |

***Constructor***

|  |  |
| --- | --- |
| Name | Description |
| + BoxToken (int row, int column) | Initializes a BoxToken object with a specified row and column. |

***Method***

|  |  |
| --- | --- |
| Name | Description |
| + int getRow () | Returns the current row position. |
| + void setRow (int row) | Sets the row position. |
| + int getColumn () | Returns the current column position. |
| + void setColumn (int column) | Sets the column position. |

* + 1. Class SokobanToken extends ImageView

***Field***

|  |  |
| --- | --- |
| Name | Description |
| - int row | Represents the current row position. |
| - int column | Represents the current column position. |
| - int characterSelected | Keeps track of the selected character. |
| - Image imageLeft1 | Represents the image for the character1 facing left. |
| - Image imageLeft2 | Represents the image for the character2 facing left. |

***Constructor***

|  |  |
| --- | --- |
| Name | Description |
| + SokobanToken (int row, int column) | Initializes a SokobanToken object with a specified row and column. |

***Method***

|  |  |
| --- | --- |
| Name | Description |
| + int getCharacterSelected () | Returns the currently selected character. |
| + static void selectCharacter (int characterIndex) | Sets the characterSelected to the specified index. |
| + int getRow () | Returns the current row position. |
| + void setRow (int row) | Sets the row position. |
| + void setColumn (int column) | Sets the column position |
| + int getColumn () | Returns the current column position. |
| + void changeImageToUp () | Set the image of the character in up direction. |
| + void changeImageToDown () | Set the image of the character in down direction. |
| + void changeImageToLeft () | Set the image of the character in left direction. |
| + void changeImageToRight () | Set the image of the character in right direction. |

* 1. **Package game.util**
     1. Class ImageLoader

***Field***

|  |  |
| --- | --- |
| Name | Description |
| - Image imageBackground | Represents the background image used in the game. |
| - Image imageWall | Represents the image used for walls. |
| - Image imageBox | Represents the image used for boxes. |
| - Image imageTarget | Represents the image used for target tiles. |
| - Image imageLock | Represents the image used for locked levels. |
| - Image imageBackButton | Represents the image used for the back button. |
| - Image soundOn | Represents the image used for the sound on icon. |
| - Image soundOff | Represents the image used for the sound off icon. |
| - Image imageLogo | Represents the logo image used in the game. |
| - Image imageTime | Represents the image used for extra time. |

***Method***

|  |  |
| --- | --- |
| Name | Description |
| + Image getImageBackground () | Returns the background image. |
| + Image getImageWall () | Returns the wall image. |
| + Image getImageBox () | Returns the box image. |
| + Image getImageTarget () | Returns the target image. |
| + Image getImageLock () | Returns the lock image. |
| + Image getImageBackButton () | Returns the back button image. |
| + Image getImageLogo () | Returns the logo image. |
| + Image getSoundOn () | Returns the image for sound on. |
| + Image getSoundOff () | Returns the image for sound off. |
| + Image getImageTime () | Returns the image for extra time. |

* + 1. Class SoundLoader

***Field***

|  |  |
| --- | --- |
| Name | Description |
| - AudioClip audioClipCrash | Represents the audio clip for crash sounds. |
| - AudioClip audioClipMove | Represents the audio clip for move sounds. |
| - AudioClip audioClipWin | Represents the audio clip for win sounds. |
| - AudioClip audioClipLose | Represents the audio clip for lose sounds. |
| - AudioClip audioClick | Represents the audio clip for click sounds. |

***Method***

|  |  |
| --- | --- |
| Name | Description |
| + void playCrashSound () | Plays the crash sound. |
| + void playWinSound () | Plays the win sound. |
| + void playLoseSound () | Plays the lose sound. |
| + void playClickSound () | Plays the click sound. |
| + void playMoveSound () | Plays the move sound. |

* + 1. Class PlayerInfoWriterReader

***Field***

|  |  |
| --- | --- |
| Name | Description |
| - String playerInfoObjectFileName | The filename used for saving and loading PlayerInfo objects. |

***Method***

|  |  |
| --- | --- |
| Name | Description |
| + void write (PlayerInfo playerInfo) | Writes a PlayerInfo object to a file using object serialization. |
| + PlayerInfo read () | Reads a PlayerInfo object from a file using object deserialization. |