

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.

2. 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

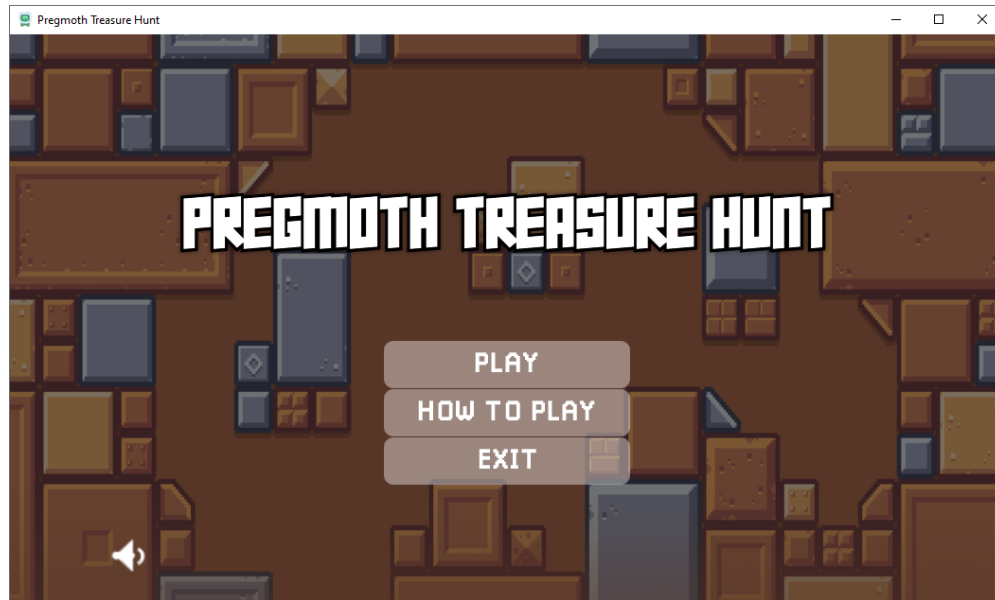
The game ends when all objects are successfully pushed onto the target squares, completing the level. Alternatively, the game ends if the player makes a move that prevents further progress (e.g., pushing an object into a corner where it cannot be moved) or the time is running out.

2.4 Project presentation

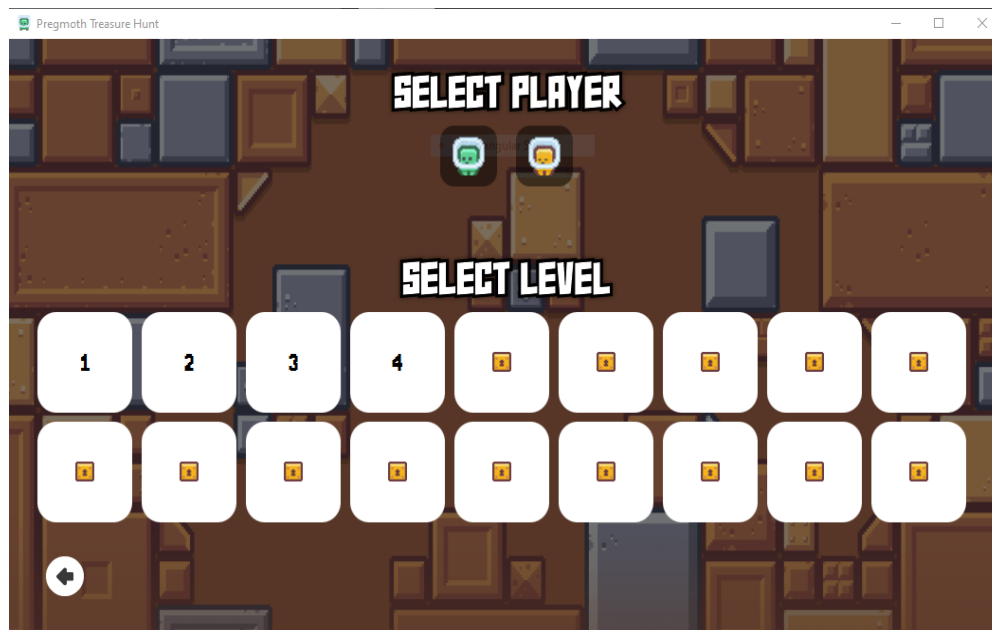
Youtube : <https://youtu.be/eE0ky1-deC8?si=FYr77hdA2fobeEoP>

3. Example

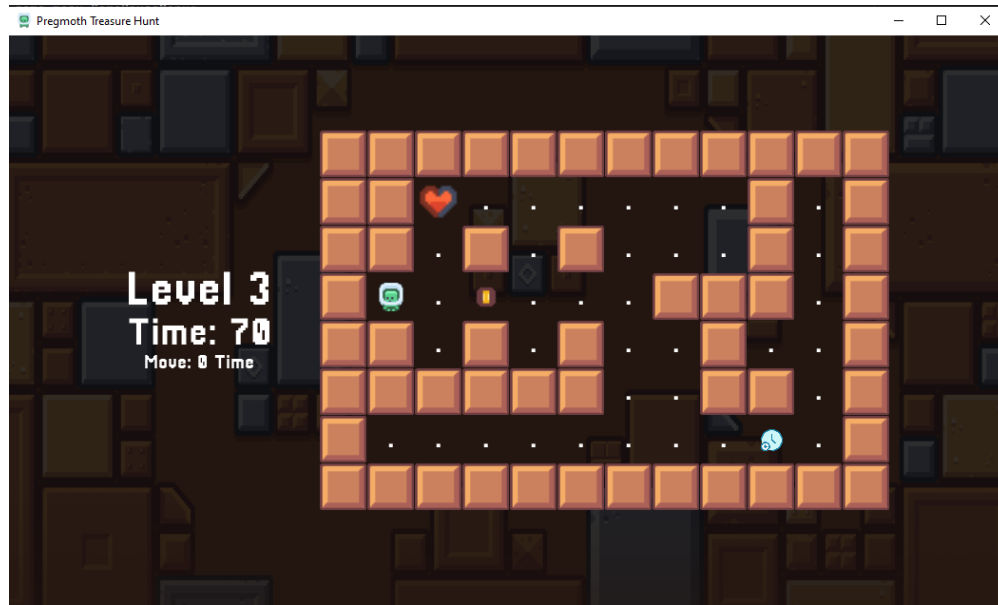
3.1 Main menu scene



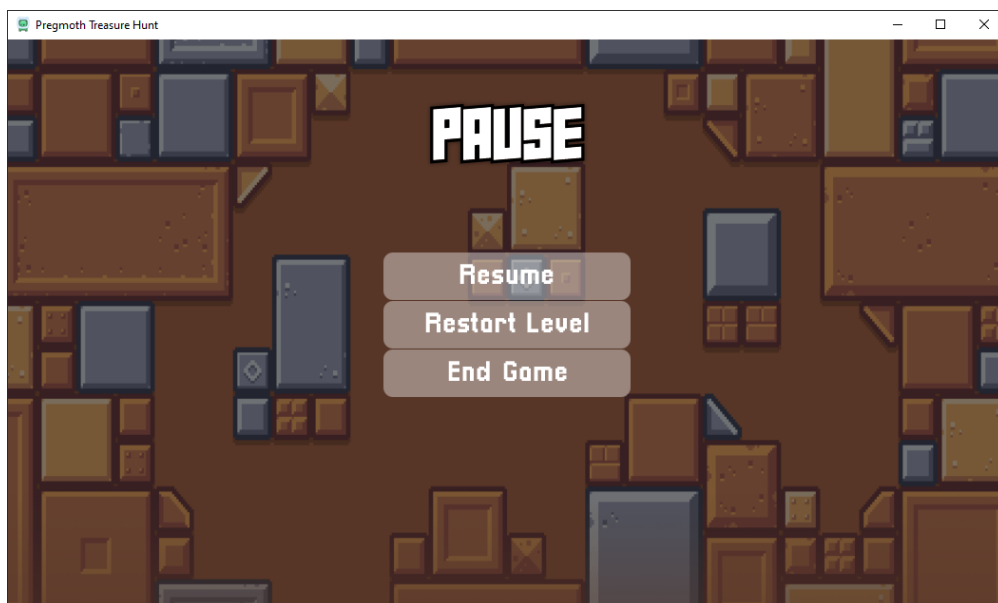
3.2 Select level and player menu



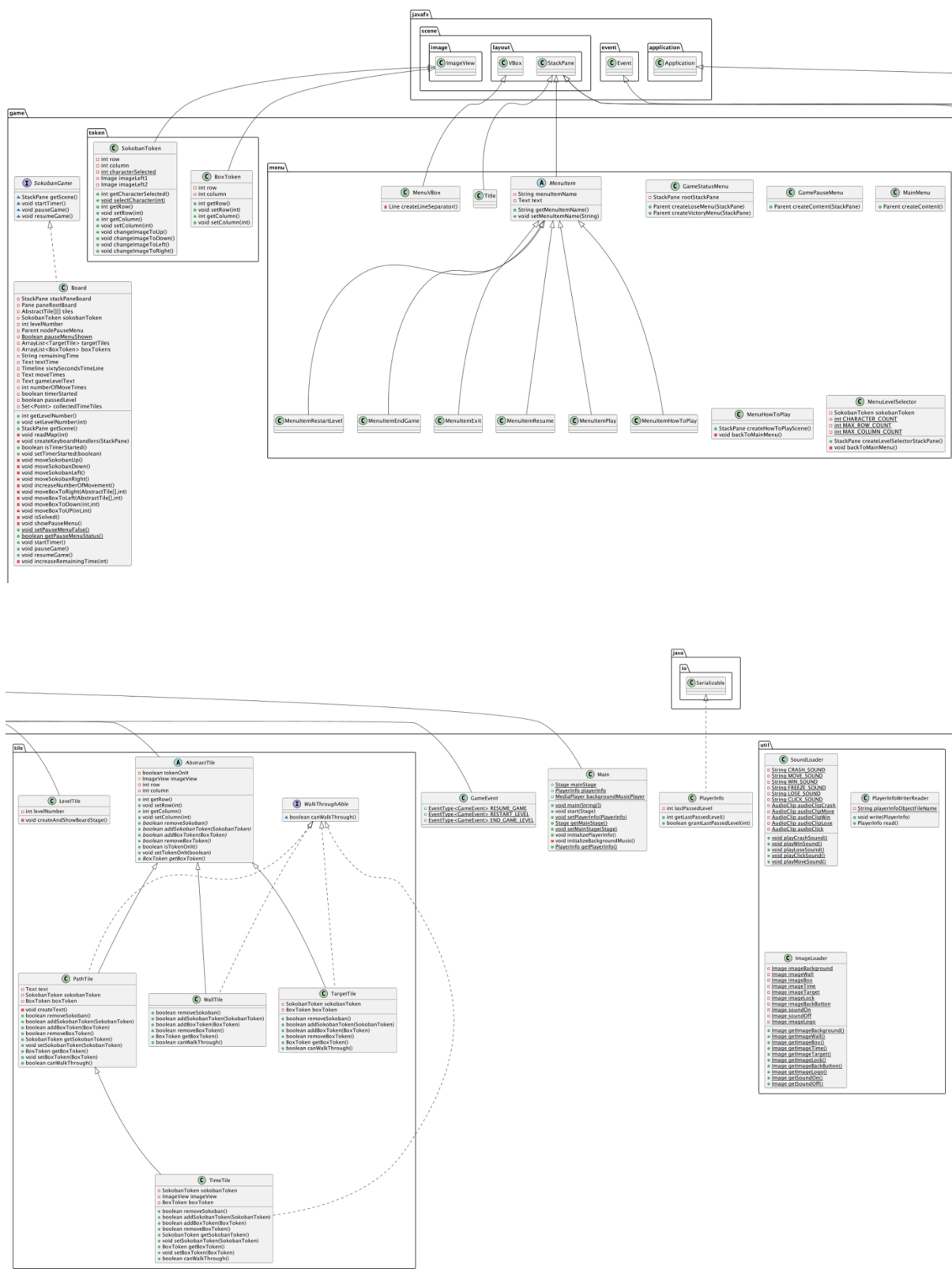
3.3 Game scene



3.4 Menu Pause



4. Class Diagram



5. Implementation Detail

5.1 Package game

5.1.1 Class Main extends Application

Field

Name	Description
<u>+ Stage mainStage</u>	The main stage of application.
<u>+ PlayerInfo playerInfo</u>	The player's information.
<u>+ MediaPlayer backgroundMediaPlayer</u>	The background's music of application

Constructor

Name	Description
<u>+ void main (String[] args)</u>	Launch the JavaFX application

Method

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

5.1.2 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
- <u>boolean pauseMenuShown</u>	- 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 ()	<ul style="list-style-type: none"> - Handle the movement of the Sokoban token in up directions. - Handle interactions with game tiles (e.g., walls, boxes, targets) during movement.
- void moveSokobanDown ()	<ul style="list-style-type: none"> - Handle the movement of the Sokoban token in down directions. - Handle interactions with game tiles (e.g., walls, boxes, targets) during movement.
- void moveSokobanLeft ()	<ul style="list-style-type: none"> - Handle the movement of the Sokoban token in left directions. - Handle interactions with game tiles (e.g., walls, boxes, targets) during movement.
- void moveSokobanRight ()	<ul style="list-style-type: none"> - 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)	<ul style="list-style-type: none"> - 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)	<ul style="list-style-type: none"> - 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)	<ul style="list-style-type: none"> - 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.

5.1.3 Class GameEvent extends Event

Field

Name	Description
+ EventType<GameEvent> <u>RESUME_GAME</u>	Represents the event type for resuming the game.
+ EventType<GameEvent> <u>RESTART_LEVEL</u>	Represents the event type for restarting the current game level.
+ EventType<GameEvent> <u>END_GAME_LEVEL</u>	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.

5.1.4 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.

5.1.5 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.

5.2 Package game.menu

5.2.1 Class MainMenu

Method

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

5.2.2 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.

5.2.3 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.

5.2.4 Class Title extends StackPane

Constructor

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

5.2.5 Class GamePauseMenu

Method

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

5.2.6 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.

5.2.7 Class MenuItemEndGame extends MenuItem

Constructor

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

5.2.8 Class MenuItemExit extends MenuItem

Constructor

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

5.2.9 Class MenuItemHowToPlay extends MenuItem

Constructor

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

5.2.10 Class MenuItemPlay extends MenuItem

Constructor

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

5.2.11 Class MenuItemRestartLevel extends MenuItem

Constructor

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

5.2.12 Class MenuItemResume extends MenuItem

Constructor

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

5.2.13 Class MenuLevelSelector

Field

Name	Description
- SokobanToken sokobanToken	- Represents a token associated with the game.
- <u>int CHARACTER_COUNT</u>	- The number of character options available for selection. - Set to 2.
- <u>int MAX_ROW_COUNT</u>	- Specifies the maximum number of rows for displaying level tiles. - Set to 3.
- <u>int MAX_COLUMN_COUNT</u>	- 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.

5.2.14 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.

5.3 Package game.tile

5.3.1 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.

5.3.2 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.

5.3.3 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.

5.3.4 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.

5.3.5 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.

5.3.6 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.

5.3.7 Interface WalkThroughAble

Method

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

5.4 Package game.token

5.4.1 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.

5.4.2 Class SokobanToken extends ImageView

Field

Name	Description
- int row	Represents the current row position.
- int column	Represents the current column position.
<u>- int characterSelected</u>	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 changelImageToUp ()	Set the image of the character in up direction.
+ void changelImageToDown ()	Set the image of the character in down direction.
+ void changelImageToLeft ()	Set the image of the character in left direction.
+ void changelImageToRight ()	Set the image of the character in right direction.

5.5 Package game.util**5.5.1 Class ImageLoader****Field**

Name	Description
- <u>Image imageBackground</u>	Represents the background image used in the game.
- <u>Image imageWall</u>	Represents the image used for walls.
- <u>Image imageBox</u>	Represents the image used for boxes.
- <u>Image imageTarget</u>	Represents the image used for target tiles.

- <u>Image imageLock</u>	Represents the image used for locked levels.
- <u>Image imageBackButton</u>	Represents the image used for the back button.
- <u>Image soundOn</u>	Represents the image used for the sound on icon.
- <u>Image soundOff</u>	Represents the image used for the sound off icon.
- <u>Image imageLogo</u>	Represents the logo image used in the game.
- <u>Image imageTime</u>	Represents the image used for extra time.

Method

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

5.5.2 Class SoundLoader

Field

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

Method

Name	Description
<u>+ void playCrashSound ()</u>	Plays the crash sound.
<u>+ void playWinSound ()</u>	Plays the win sound.
<u>+ void playLoseSound ()</u>	Plays the lose sound.
<u>+ void playClickSound ()</u>	Plays the click sound.
<u>+ void playMoveSound ()</u>	Plays the move sound.

5.5.3 Class PlayerInfoWriterReader

Field

Name	Description
<u>- String playerInfoObjectFileName</u>	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.