Bombman Documentation

Created by

Chinnapat Nitipornsri 6332009521

Chariyaporn Umpornpan 6331302121

2110215 Programming Methodology I

Semester 2 Year 2020

Chulalongkorn University

BombMan

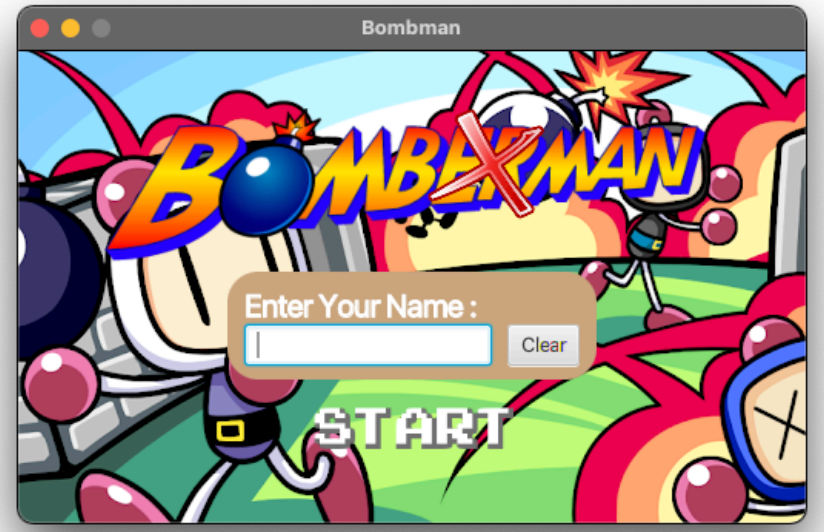
Introduction

BombMan is inspired by famous strategic video game which the player has to defeat enemies and reach an exit to progress through levels. The Bombman Game involves strategically placing down bombs, which explode in multiple directions after a certain amount of time, in order to destroy obstacles and kill enemies.

Rule

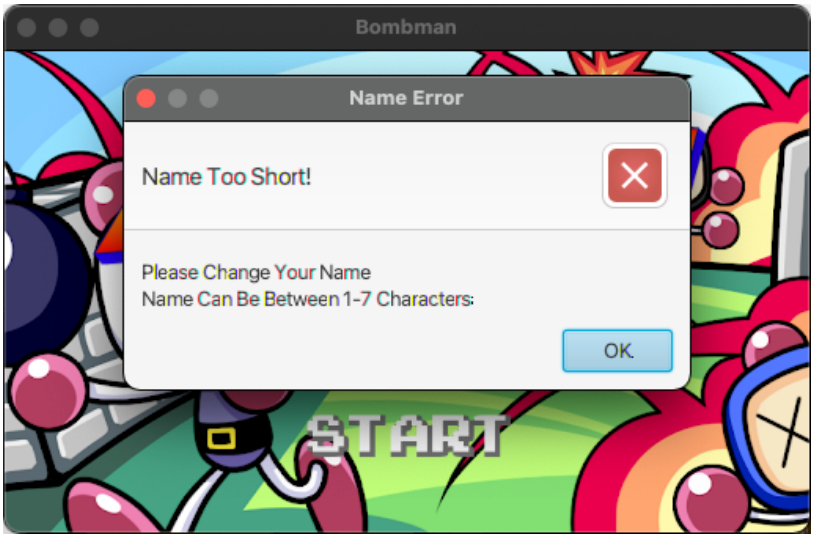
The player has to plant the bomb to destroy the box to search for items which are AddRadius Item (to increase the bomb radius), AddBomb Item (to increase the maximum bomb capacity), Key (to unlock the Door), and Door (to complete the level). Meanwhile, the monsters are walking around the map and hinder player from completing the level, but the player can also use bombs to eliminate those monsters. The higher level player gets, the stronger monsters appear. The player clears the game when passing level 5.

Start scene



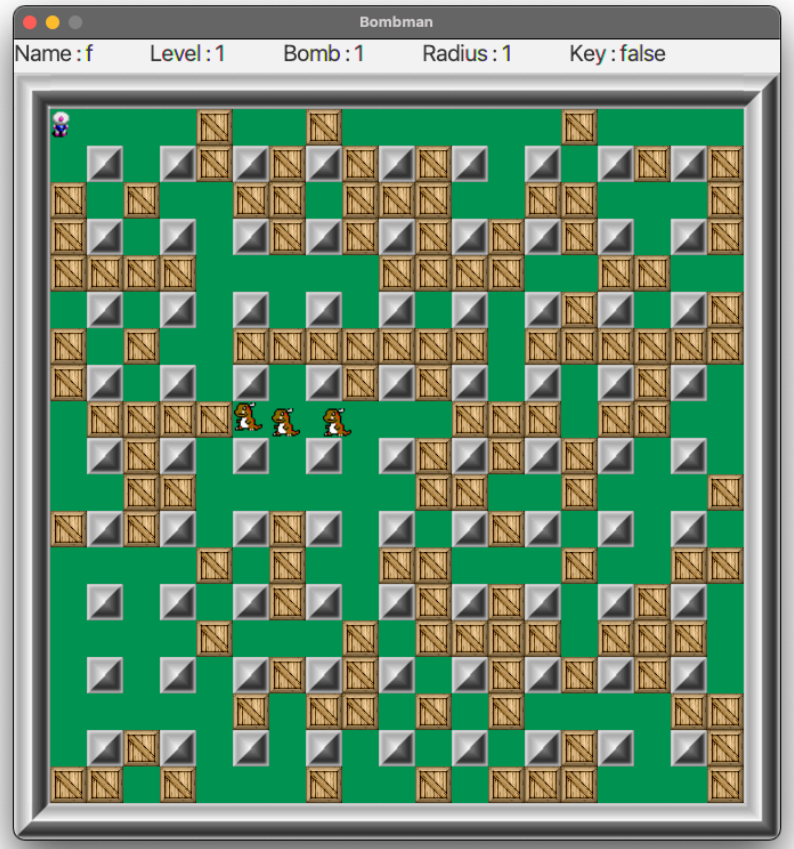
The start scene consists of logo of this game (on top of the scene), input name area (the middle one), and “Start” button (click to start playing game). You can also hit “Enter” after finishing input name to enter the game.

Alert



The user must input name at least 1 character (Name Too Short!) and not more than 7 characters (Name Too Long!). If user's name is not valid, the alert will pop up.

Game scene

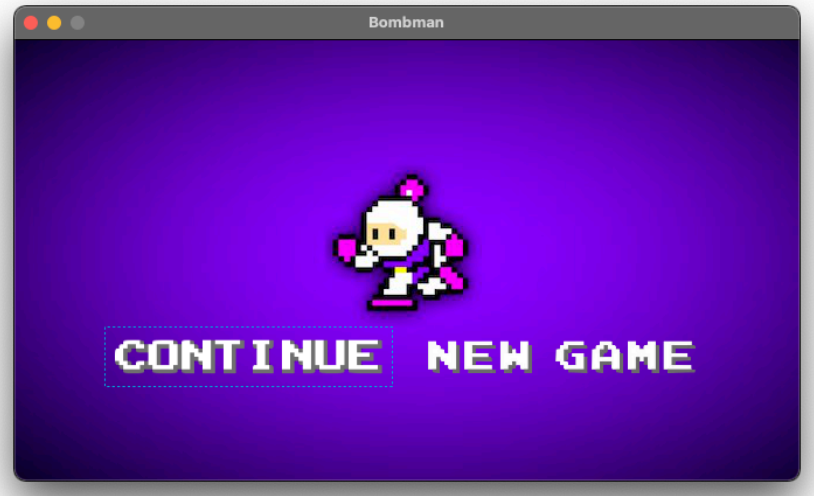


The game scene consists of status panel which shows player's name, level of current stage, player's maximum bomb capacity (Maximum bomb is 5), bomb's radius (Maximum radius is 5), and key status of player. The rest of this window is playing game area. The player should plant the bomb to search for items, key and door to complete the level, while the player has to avoid interact with monsters.

Lose scene

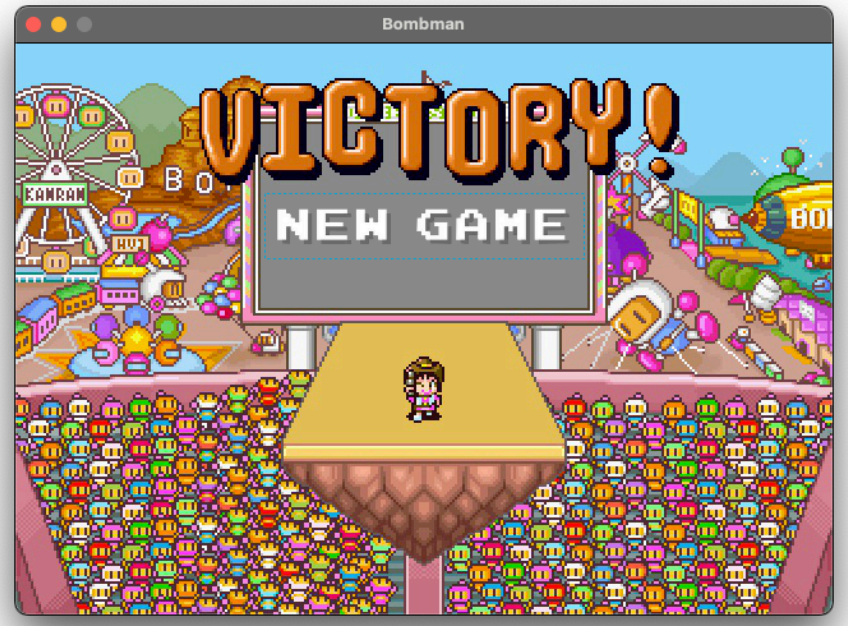
If player is in exploding area or interact with any monsters, the player dies and lose scene appears. There is “New Game” button to start new game.

Next Level scene



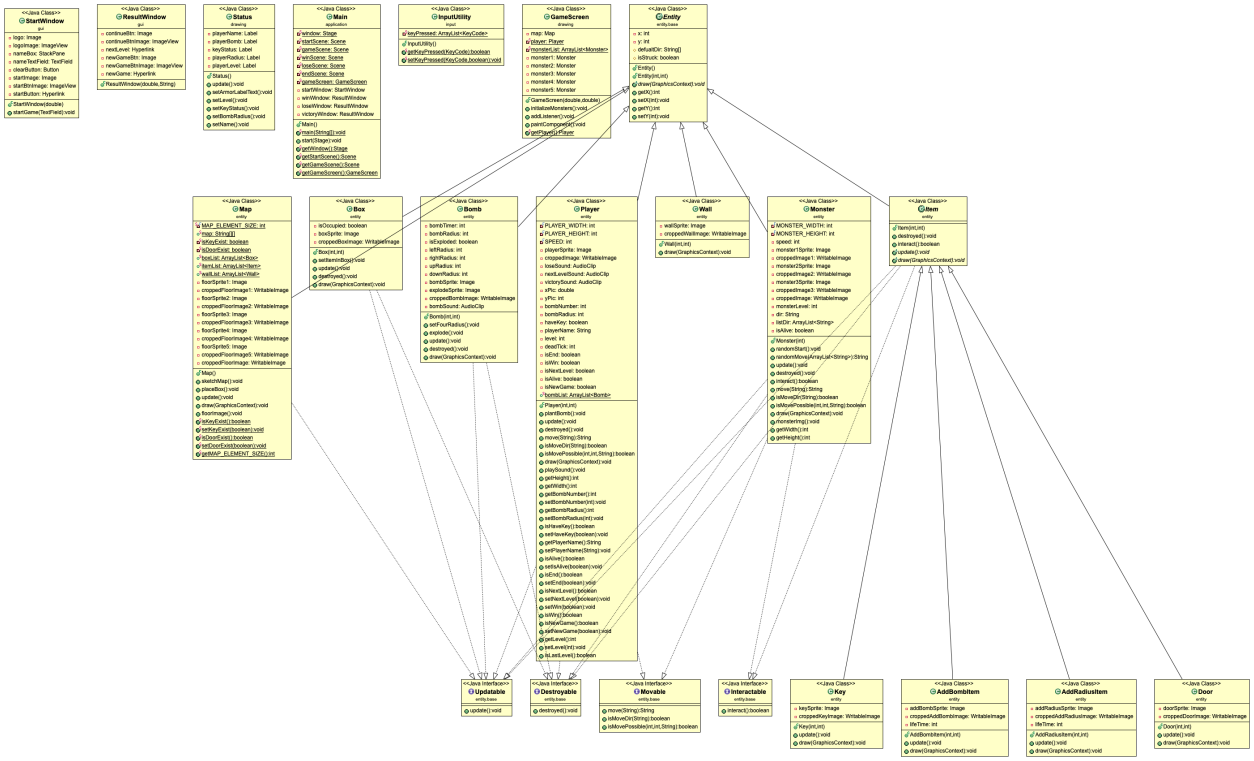
When player clears the stage, which is not the last stage, next-level scene appears. There are two buttons on this scene that are “Continue” button (click to play next level) and “New Game” button (click to start new game).

Victory scene



When player clear the last stage (level 5), victory scene appears. There is “New Game” button to start new game.

Class Diagram



1. Package application

1.1 Class Main extends Application

1.1.1 Fields

- Stage window	Game's window
- Scene startScene	Start scene of the game that let user input his/her name before entering the game
- Scene gameScene	Game scene
- Scene winScene	Win scene that show when user completes the game on each level
- Scene loseScene	Game over scene
- Scene endScene	Win scene that show when user completes the game on all levels.

- GameScreen gameScreen	User game screen.
- StartWindow startWindow	Start scene window.
- ResultWindow winWindow	Win scene window
- ResultWindow loseWindow	Lose scene window.
- ResultWindow victoryWindow	End scene window.

1.1.2 Methods

+ void start(Stage stage)	Contains each scene.
+ getter for each field	Getter methods for window, startScene, gameScene, and GameScreen.

2. Package drawing

2.1 Class GameScreen extends Canvas

2.1.1 Fields

- Map map	Game's map.
- Player player	Player.
- ArrayList<Monster> monsterList	List of monsters in each level.
- Monster monster1	1st monster.
- Monster monster2	2nd monster.
- Monster monster3	3rd monster.
- Monster monster4	4th monster (only exists in last level).
- Monster monster5	5th monster (only exists in last level).

2.1.2 Constructor

+ GameScreen(double width, double height)	- Set the information of the super class. - Initialize map, player, monsterList, and monsters. - Add Listener.

2.1.3 Methods

+ void initializeMonsters()	Initialize all monsters for each level.
+ void addListener()	Receive user's keypress.
+ void paintComponent()	Draw game screen on GraphicsContext.
+ Player getPlayer()	Get player.

2.2 Class Status extends HBox

2.2.1 Fields

- Label playerName	Player's name.
- Label playerBomb	The amount of bombs player has.
- Label keyStatus	Determine whether player has key or not.
- Label playerRadius	Player's bomb radius.
- Label playerLevel	Player's level.

2.2.2 Constructor

+ Status()	Initialize all fields.

2.2.3 Methods

+ void update()	Update values of all fields.
+ setter for each field	Set value of all fields.

3. Package gui

3.1 Class StartWindow extends VBox

3.1.1 Fields

- Image logo	Game logo's image.
- ImageView logoImage	Game logo's image with specific properties.
- StackPane nameBox	Contains text field for input username and clear button.
- TextField nameTextField	Let user input his/her name.
- Button clearButton	Clear the input name in nameTextField.
- Image startImage	Start button's image.
- ImageView startBtnImage	Start button's image with specific property.
- Hyperlink startButton	Start Button that links user to the game scene.

3.1.2 Constructor

+ StartWindow(double spacing)	Set spacing, alignment, initialize and add buttons to window.

3.1.3 Method

+ void startGame(TextField nameTextField)	Start button handler. If the input name is valid, it will links user to the game scene. Otherwise, the alert will appear which warns the user to change his/her name to a valid name before entering the game.

3.2 Class ResultWindow extends HBox

3.2.1 Fields

- Image continueBtn	NextLevel button's image.
- ImageView continueBtnImage	NextLevel button's image with specific properties.
- Hyperlink nextLevel	NextLevel button that links user to next level game scene.
- Image newGameBtn	NewGame button's image.
- ImageView newGameBtnImage	NewGame button's image with specific properties.
- Hyperlink newGame	NewGame button that links user to start scene to start new game.

3.2.2 Constructor

+ ResultWindow(double spacing, String condition)	- Set spacing, alignment, initialize and add buttons to window. - If the condition is “win”, the window will contain both nextLevel and newGame buttons. But if the condition is “lose” or “victory”, the window will contain only newGame button.

4. Package entity.base

4.1 Class Entity

4.1.1 Fields

- int x	Object’s x position.
- int y	Object’s y position.
# String[] defualtDir	Default object’s direction.
# boolean isStruck	Indicate whether object collides with other objects or not.

4.1.2 Constructor

+ Entity()	Default constructor.
+ Entity(int x, int y)	Initialize object’s position.

4.1.3 Methods

+ void draw(GraphicsContext gc)	

4.2 Interface Destroyable

+ void destroyed()	Destroy object.

4.3 Interface Interactable

+ boolean intereact()	Interaction between objects.

4.4 Interface Movable

+ String move(String dir)	Move object.
+ boolean isMoveDir(String dir)	Check that whether the target position is valid or not.
+ boolean isMovePossible(int targetX, int targetY, String sprite)	Check that whether the target position is the sprite's position or not.

4.5 Interface Updatable

+ void update()	Update object.

5. Package entity

5.1 Class Map extends Entity implements Updatable

5.1.1 Fields

- int MAP_ELEMENT_SIZE	Map's element size.
+ String[][] map	Floor, Walls, Items, and Bombs index on the map.

- boolean isKeyExist	Exist Status of the key.
- boolean isDoorExist	Exist Status of the door.
+ ArrayList<Box> boxList	List of all boxes.
+ ArrayList<Item> itemList	List of all items.
+ ArrayList<Wall> wallList	List of all walls.
- Image floorSprite1	Level 1 Floor Image.
- WritableImage croppedFloorImage1	Cropped Level 1 Floor Image.
- Image floorSprite2	Level 2 Floor Image.
- WritableImage croppedFloorImage2	Cropped Level 2 Floor Image.
- Image floorSprite3	Level 3 Floor Image.
- WritableImage croppedFloorImage3	Cropped Level 3 Floor Image.
- Image floorSprite4	Level 4 Floor Image.
- WritableImage croppedFloorImage4	Cropped Level 4 Floor Image.
- Image floorSprite5	Level 5 Floor Image.
- WritableImage croppedFloorImage5	Cropped Level 5 Floor Image.
- WritableImage croppedFloorImage	Cropped Floor Image.

5.1.2 Constructor

+ Map()	- Initailize all elements. - Set isKeyExist and isDoorExist to be false. - Sketch the map. - Place all boxes.

5.1.3 Methods

+ void sketchMap()	Indicate the position of walls and floors.
+ void placeBox()	Place boxes on the map's floor following sketched map.
+ void update()	Update each box and item.
+ void draw(GraphicsContext gc)	Draw floors, walls, boxes, and items on GraphicsContext.
+ void floorImage()	Set floors' color.
+ Getter / setter	Get and set value of each field.

5.2 Class Wall extends Entity

5.2.1 Fields

- Image wallSprite	Wall Image.
- WritableImage croppedWallImage	Cropped Wall Image.

5.2.2 Constructor

+ Wall(int x, int y)	Set the information of the super class. Set this wall's image.

5.2.3 Methods

+ void draw(GraphicsContext gc)	Draw this wall on GraphicsContext.

5.3 Class Box extends Entity implements Updatable, Destroyable

5.3.1 Fields

- boolean isOccupied	Occupied status of this box.
- Image boxSprite	Box Image.
- WritableImage croppedBoxImage	Cropped Box Image.

5.3.2 Constructor

+ Box(int x, int y)	- Set the information of the super class. - Set Item in this box.

5.3.3 Methods

+ void setItemInBox()	Set isOccupied.
+ void update()	This box is destroyed if it is in exploding area.
+ void destroyed()	- Emerge the item if isOccupied is true. - Remove this box from Map.boxList.
+ void draw(GraphicsContext gc)	Draw this box on GraphicsContext.

5.4 Class Item extends Entity implements Updatable, Destroyable, Interactable

5.4.1 Constructor

+ Item(int x, int y)	Set the information of the super class.

5.4.2 Methods

+ void destroyed()	Remove this Item from Map.itemList.
+ boolean interact()	Check and return that whether the player interact with this item or not.
+ void update()	Update this Item.
+ void draw(GraphicsContext gc)	Draw this Item.

5.5 Class AddBombItem extends Item

5.5.1 Fields

- Image addBombSprite	AddBombItem Image.
- WritableImage croppedAddBombImage	Cropped AddBombItem Image.
- int lifeTime	Delay time before item emerges.

5.5.2 Constructor

+ AddBombItem(int x, int y)	- Set the information of the super class. - Initialize lifeTime to be 0.

5.5.3 Methods

+ void update()	This item is destroyed if it is in exploding area or the player interact with it.
+ void draw(GraphicsContext gc)	Draw this item on GraphicsContext.

5.6 Class AddRadiusItem extends Item

5.6.1 Fields

- Image addRadiusSprite	AddRadiusItem Image.
- WritableImage croppedAddRadiusImage	Cropped AddRadiusItem Image.
- int lifeTime	Delay time before item emerges.

5.6.2 Constructor

+ AddRadiusItem(int x, int y)	- Set the information of the super class. - Initialize lifeTime to be 0.

5.6.3 Methods

+ void update()	This item is destroyed if it is in exploding area or the player interact with it.
+ void draw(GraphicsContext gc)	Draw this item on GraphicsContext.

5.7 Class Door extends Item

5.7.1 Fields

- Image doorSprite	Door Image.
- WritableImage croppedDoorImage	Cropped Door Image.

5.7.2 Constructor

+ Door(int x, int y)	Set the information of the super class.

5.7.3 Methods

+ void update()	If the player has key and interact with this Door.
+ void draw(GraphicsContext gc)	Draw this door on GraphicsContext.

5.8 Class Key extends Item

5.8.1 Fields

- Image keySprite	Key Image.
- WritableImage croppedKeyImage	Cropped Key Image.

5.8.2 Constructor

+ Key(int x, int y)	Set the information of the super class.

5.8.3 Methods

+ void update()	Destroy this key if the player interacts with it.
+ void draw(GraphicsContext gc)	Draw this key on GraphicsContext.

5.9 Class Monster extends Entity implements Updatable, Destroyable, Interactable, Movable

5.9.1 Fields

- int MONSTER_WIDTH	Monster's width.
- int MONSTER_HEIGHT	Monster's height.
- int speed	Monster's speed.
- Image monster1Sprite	Monster level 1's image.
- WritableImage croppedImage1	Monster level 1's cropped image.
- Image monster2Sprite	Monster level 2's image.
- WritableImage croppedImage2	Monster level 2's cropped image.
- Image monster3Sprite	Monster level 3's image.
- WritableImage croppedImage3	Monster level 3's cropped image.
- WritableImage croppedImage	Monster cropped image.
- int monsterLevel	Monster's level.
- String dir	Move direction.
- ArrayList<String> listDir	Directions' list.
- boolean isAlive	Monster's life status.

5.9.2 Constructor

+ Monster(int monsterLevel)	- Initialize isAlive to be true. - Set start position and move direction. - Initialize level's monster. - Initialize speed of this monster to equal level's monster.

5.9.3 Methods

+ void randomStart()	Set start position of the monster.
+ String randomMove()	Set monster's direction.
+ void update()	Update movement, life status and interaction with player of the monster.
+ void destroyed()	Check monster's life status.
+ boolean interact()	Check monster's interaction with player.
+ String move(String dir)	Move monster.
+ boolean isMovedir(String dir)	Check that whether the target position is valid or not.
+ boolean isMovePossible(int targetx, int targety, String sprite)	Check that whether the target position is the sprite's position or not.
+ void draw(GraphicsContext gc)	Draw monster.
+ void monsterImg()	Choose croppedImage by monster's level.
+ getter for monster's size	Get monster's width and height

5.10 Class Player extends Entity implements Updatable, Destroyable, Movable

5.10.1 Fields

- int PLAYER_WIDTH	Player’s width.
- int PLAYER_HEIGHT	Player's height.
- int SPEED	Player’s speed.
- Image playerSprite	Player Image.
- WritableImage croppedImage	Cropped Player Image.
- AudioClip loseSound	Lose Screen Sound Effect.
- AudioClip nextLevelSound	Next-level Screen Sound Effect.
- AudioClip victorySound	Victory Screen Sound Effect.
- double xPic	Horizontal Player Image Position.
- int yPic	Vertical Player Image Position.
- int bombNumber	Bomb capacity (Maximum capacity is 5).
- int bombRadius	Radius of each bomb (Maximum radius is 5).
- boolean haveKey	Having Key Status
- String playerName	Player’s name.
- int level	Player’s level (maximum level is level 5).
- int deadTick	Dead Scene Timer
- boolean isEnd	End Status
- boolean isWin	Win Status
- boolean isNextLevel	Next-level Status
- boolean isAlive	Player’s life status.

- boolean isNewGame	New game Status
+ ArrayList<Bomb> bombList	List of bombs.

5.10.2 Constructor

+ Player(int x, int y)	- Set the information of the super class. - Initialize xPic, deadTick to be 0. - Initialize croppedImage, bombList. - Initialize bombNumber, bombRadius, and level to be 1. - Initialize haveKey, isEnd, isWin, and isNewGame to be false. - Initialize isAlive to be true.

5.10.3 Methods

+ void plantBomb()	Plant bomb on the current player's position.
+ void update()	- Move player by the input direction(A = move left, W = move up, S = move down, and D = move right) - Update the bombs. - Check Alive Condition. - Play Sound Effect.
+ void destroyed()	If the player is in Exploding Area, isAlive is false then the dead scene starts.
+ String move(String dir)	Move player to the proper position.
+ boolean isMoveDir(String dir)	Check that whether the target position is valid or not.

+ boolean isMovePossible(int targetx, int targety, String sprite)	Check that whether the target position is the sprite's position or not.
+ void draw(GraphicsContext gc)	Draw the player movement or dead scene and bombs on GraphicsContext.
+ void playSound()	Play the proper sound effect to the next screen.
+ Getter / setter	Get and set value of each field.

5.11 Class Bomb extends Entity implements Updatable, Destroyable

5.11.1 Fields

- int bombTimer	This bomb's lifetime.
- int bombRadius	The default bombRadius of player.
- boolean isExploded	Exploding status of this bomb.
- int leftRadius	Real left distance of exploding area.
- int rightRadius	Real right distance of exploding area.
- int upRadius	Real up distance of exploding area.
- int downRadius	Real down distance of exploding area.
- Image bombSprite	Bomb Image.
- Image explodeSprite	Exploding Area Image.
- WritableImage croppedBombImage	Cropped Bomb Image.

- AudioClip bombSound	Exploding Sound.

5.11.2 Constructor

+ Bomb(int x, int y)	- Set the information of the super class. - Initailize bombTimer to be 250. - Set bombRadius from player's bombRadius. - Set is Exploded to be false. - Set Four Radius.

5.11.3 Methods

+ void setFourRadius()	The exploding radiuses equal bombRadius if it is not blocked by the walls.
+ void explode()	Explode the Exploding Area. Play explode sound.
+ void update()	This Bomb is exploded when bombTimer equals 18 and is destroyed when bombTimer equals 0.
+ void destroyed()	- Remove this bomb from Player.bombList. - Return Exploding Area to be normal floors.
+ void draw(GraphicsContext gc)	Draw this bomb on GraphicsContext.

6. Package input

6.1 Class InputUtility

6.1.1 Field

- ArrayList<KeyCode> keyPressed	ArrayList of keycodes that user presses on the keyboard.

6.1.2 Method

+ boolean getKeyPressed(KeyCode keycode)	Return whether keycode is pressed or not.
+ void setKeyPressed(KeyCode keycode, boolean pressed)	Add keycode in keyPressed if the key is pressed and remove keycode from keyPressed if the user releases the key.