**Bomber Man**

Introduction

Figure 1 : Main Menu Screen

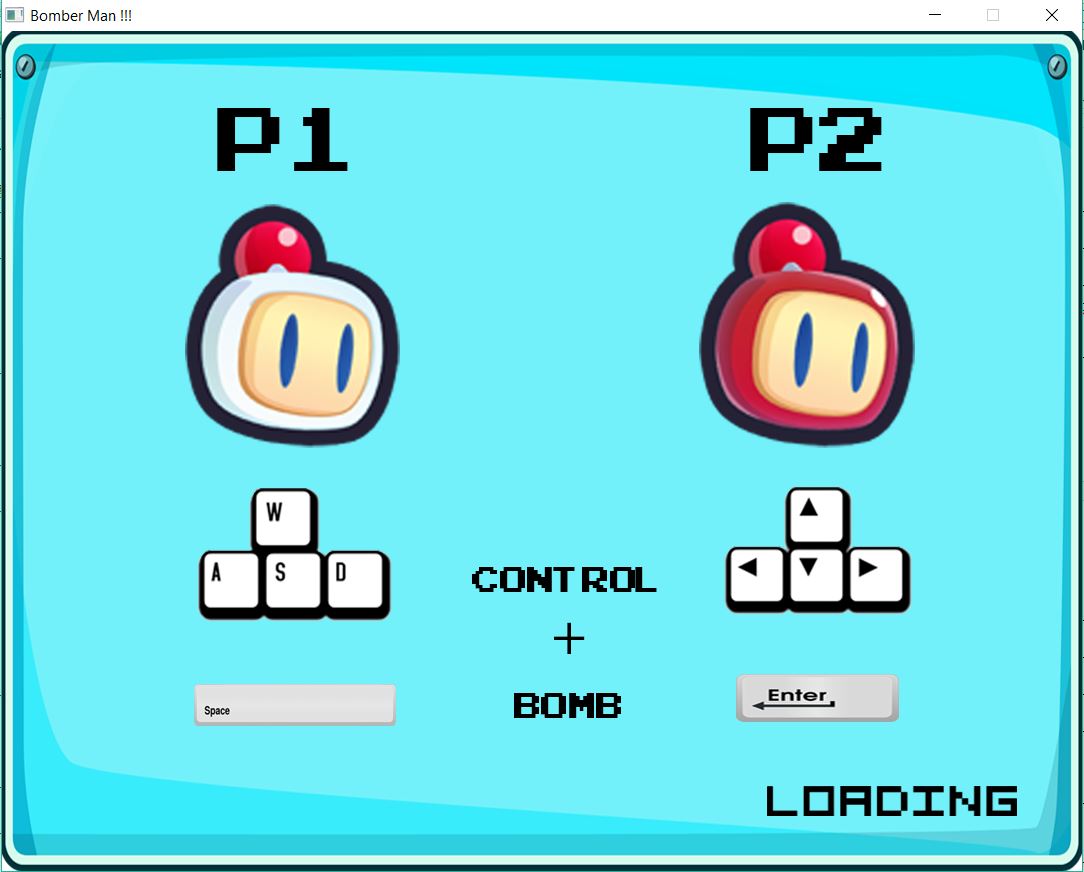
Bomber Man game is a strategic, maze-based video game which requires two players. The main purpose of the game is to eliminate each other and be the last one standing. This game involves strategically placing down bombs, which explode in multiple directions after a certain amount of time, in order to destroy obstacles and kill enemies and other players. The player can pick up various power-ups, giving them benefits such as larger explosions or the ability to place more bombs down at a time. The player is killed if they get caught up in a bomb's explosion, including their own.

Instruction

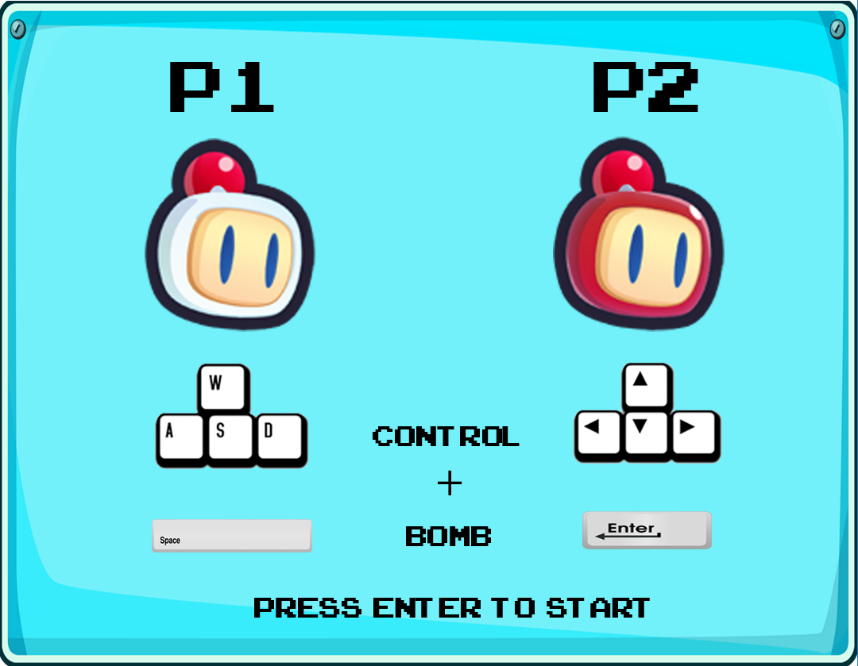
 The game will start with Main Menu Screen as shown in figure 1 and the start button will change color when mouse entered the button as shown in figure 2.

Figure 2 : Start Button

When click on start button the game will go to loading screen as shown in figure 3.

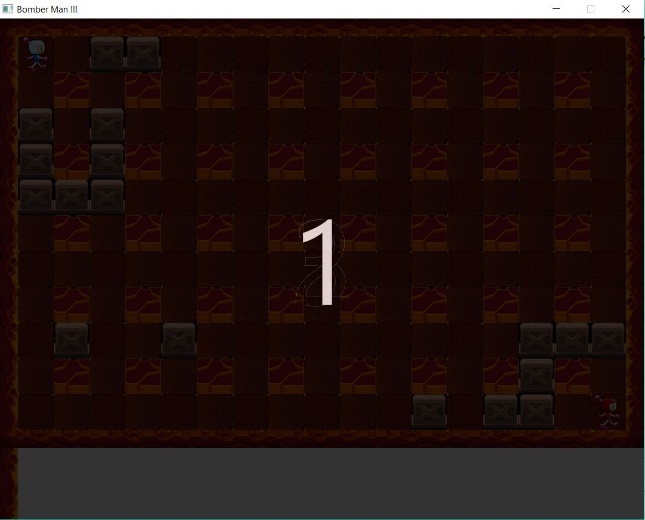
Figure 3 : Loading Screen

The game will wait for 3 second before showing text “Press Enter to Start” (as shown in figure 4) then if you press enter the game will start. (The game will not start if you press enter before the text shows up.)

Figure 4 : Enter to start

This Screen also shows how to control player1 and player2 which are

|  |  |  |
| --- | --- | --- |
| **Action** | **Player 1 (White)** | **Player 2 (Red)** |
| Move up | W | Arrow up |
| Move down | S | Arrow down |
| Move left | A | Arrow left |
| Move right | D | Arrow right |
| Drop bomb | Spacebar | Enter |

Figure 5 : Countdown Screen

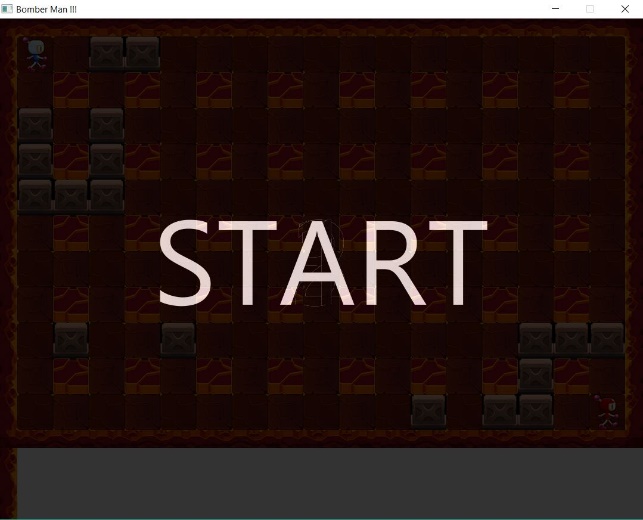
 The game will start by showing a countdown screen counting down from 3 to 1 then will show message “start” as shown in figure 5 and 6.

Figure 6 : Start

Then the game will show the maze and player can play the game. (As shown in figure 7)



Figure 7 : The Game Maze

The lower part of the screen showing current status of the range of explosion, number of bomb left to drop, and speed of each player. In the maze there are two types of wall, unbreakable one and breakable one (calls brick). The maze will place the brick randomly each time the game restart but wall will stay at the same position. Every time the brick is break, it will randomly choose item to show up between Boost, Deboost, Stackbomb, Debomb, Upgradebomb, Degradebomb and nothing.

Figure 8 : Player1 Wins

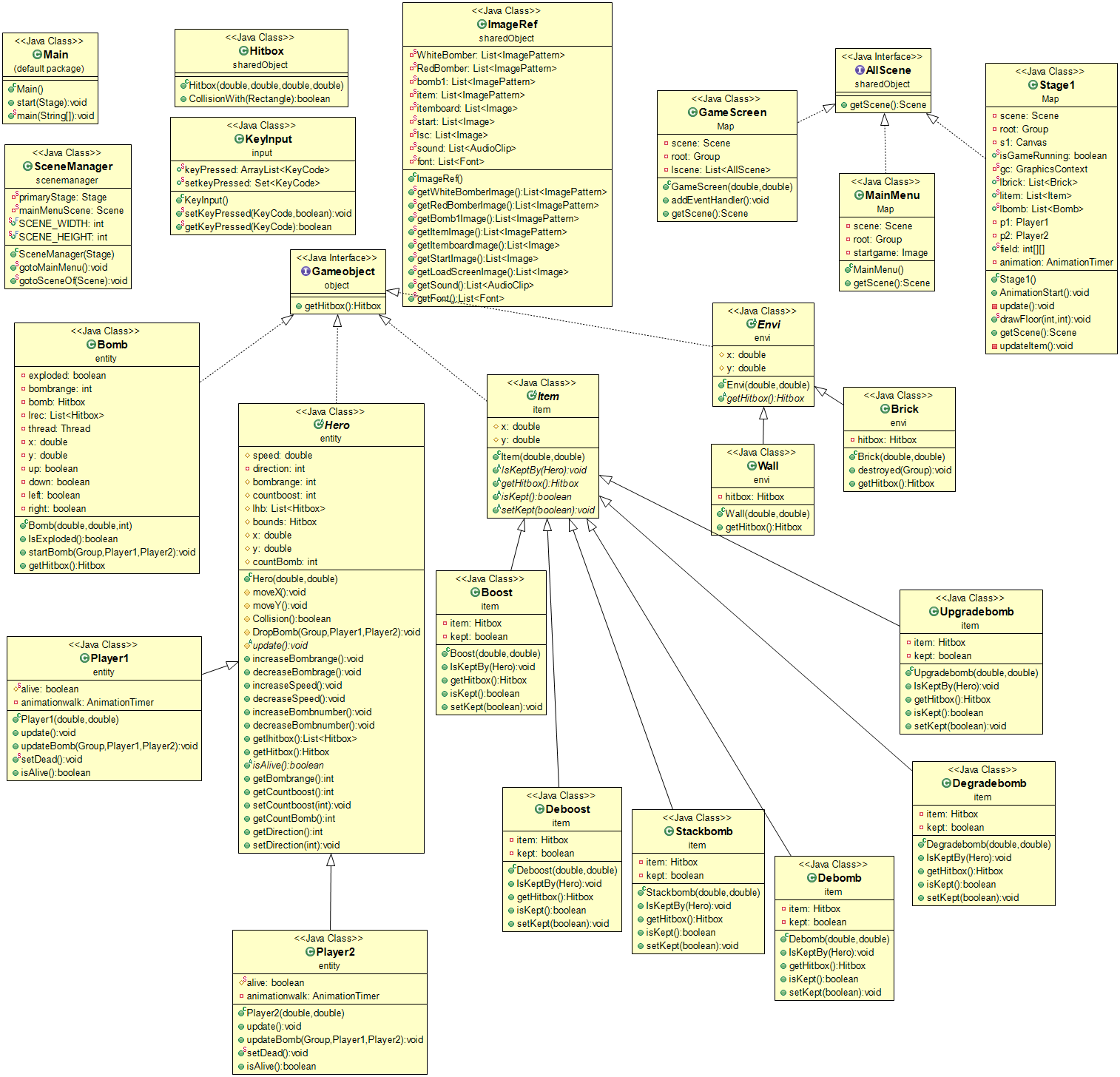
 The game will end if any player is killed. Then the game will show a result screen that telling which player is the winner or if both of the player were killed at the same time as shown in figure 8, 9 and 10.

Figure 9 : Player2 Wins

Figure 10 : Draw

Object in game

|  |  |
| --- | --- |
| Boost Item | Increase speed |
| Deboost Item | Decrease speed |
| StackBomb Item | Increase number of bomb that can be drop |
| Debomb Item | Decrease number of bomb that can be drop |
| Upgradebomb Item | Increase explosion range |
| Degradebomb Item | Decrease explosion range |
| Wall | Unbreakable wall |
| Brick | Breakable wall |



1. Package (default package)

1.1 Class Main extends Application

Method

|  |  |
| --- | --- |
| + void start(Stage stage) | -set the stage to cannot be resize  -initialize SceneManager  -use SceneManager to go to Main Menu scene  -play the background sounds |
| + static void main(String[] args) | -make the program start |

2. Package scenemanager

2.1 Class SceneManager

Field

|  |  |
| --- | --- |
| - static Stage primaryStage | -JavaFX Container |
| - static Scene mainMenuScene | -A main menu scene |
| + static final int SCENE\_WIDTH | -Width of the application scene which is 1080 |
| + static final int SCENE\_HEIGHT | -Height of the application scene which is 960 |

Constructor

|  |  |
| --- | --- |
| + SceneManager(Stage stage) | -Initialize MainMenu  -set the primaryStage by the given Stage and mainMenuScene by the scene of MainMenu  -set the title of primaryStage to be  “Bomber Man !!!” and show |

Method

|  |  |
| --- | --- |
| + static void gotoMainMenu() | -set the scene of primaryStage to be mainMenuScene |
| + static void gotoSceneOf(Scene scene) | -set the scene of primaryStage by the given scene |

3. Package sharedObject

3.1Class Hitbox extends Rectangle

Constructor

|  |  |
| --- | --- |
| + Hitbox(double x, double y, double w, double h) | -initialize a Rectangle at position (x,y) with width=w and height=h |

Method

|  |  |
| --- | --- |
| + boolean CollisionWith(Rectangle other) | -check the intersection of this rectangle and the given rectangle  -return true if they are intersect |

3.2 Class ImageRef

Field

|  |  |
| --- | --- |
| - static List<ImagePattern> WhiteBomber | -a list of images for the Player1 |
| - static List<ImagePattern> RedBomber | -a list of images for the Player2 |
| - static List<ImagePattern> bomb1 | -a list of images for the bomb |
| - static List<ImagePattern> item | -a list of images for item |
| - static List<Image> itemboard | -a list of images for item status board |
| - static List<Image> start | -a list of images for start button |
| - static List<Image> lsc | -a list of images for game screen |
| - static List<AudioClip> sound | -a list of sounds for the game |
| - static List<Font> font | -a list of fonts for the game |

Constructor

|  |  |
| --- | --- |
| + ImageRef() | -load all the image and add to the it’s list |

Method

|  |  |
| --- | --- |
| +Getter method for every field |  |

3.3 Interface AllScene

Method

|  |  |
| --- | --- |
| + Scene getScene() | (All the scene must be able to get scene) |

4. Package input

4.1 Class KeyInput

Field

|  |  |
| --- | --- |
| + static ArrayList<KeyCode> keyPressed | -an array list to contain KeyCode |
| + static Set<KeyCode> setkeyPressed | -a hash set of Keycode that isn’t contain the same KeyCode more than once |

Method

|  |  |
| --- | --- |
| + static void setKeyPressed(KeyCode key, boolean pressed) | -if the keyboard is pressed and keyPressed not contain the given KeyCode then add the KeyCode to keyPressed |
| + static boolean getKeyPressed(KeyCode key) | -return true if keyPressed contain the given KeyCode |

5. Package Map

5.1 Class MainMenu implements AllScene

Field

|  |  |
| --- | --- |
| - Scene scene | -a Scene of main menu |
| - Group root | -a Pane that contains objects in the scene |
| - Image startgame | -image of the main menu scene |

Constructor

|  |  |
| --- | --- |
| + MainMenu() | -initialize root, scene  -create canvas and add to be children of root  -load image for startgame and draw on the canvas  -create canvas for start button and draw image from ImageRef  -create event handler on mouse entered start button to change color of the button and change back when exit  -create event handler for mouse click start button to create GameScreen and go to scene of it using SceneManager |

Method

|  |  |
| --- | --- |
| + Scene getScene() | -getter of scene |

5.2 Class GameScreen implements AllScene

Field

|  |  |
| --- | --- |
| - Scene scene | -Scene of GameScreen |
| - Group root | -Pane that contains objects in the scene |
| - List<AllScene> lscene | -List contain all game scene  (in case there are more than one stage) |

Constructor

|  |  |
| --- | --- |
| + GameScreen(double width, double height) | -initialize root and scene  -create canvas and draw image from ImageRef then add to be children of root  -create new thread to draw loading image and wait for 3 second then redraw the gamscreen image and addEventHandler  -create text message |

Method

|  |  |
| --- | --- |
| + void addEventHandler() | -initialize lscene  -create Stage1 and add to lscene  -create event handler on key press  -if press ENTER go to scene of stage1 using  SceneManager and start animation  -if press ESCAPE exit the game |
| + Scene getScene() | -getter of scene |

5.3 Class Stage1 implements AllScene

Field

|  |  |
| --- | --- |
| - Scene scene | -Scene of Stage1 |
| - Group root | -Pane that contains objects in the scene |
| - Canvas s1 | -base canvas of the stage |
| + static boolean isGameRunning | -true if there is no winner yet |
| - static GraphicsContext gc | -graphics context of canvas s1 |
| + static List<Brick> lbrick | -list that contains all exist brick(breakable wall) in the game |
| + static List<Item> litem | -list that contains all item that haven’t been kept by anyone in the game |
| + static List<Bomb> lbomb | -list that contains all bomb that haven’t explode |
| - Player1 p1 | -Player1 of the game |
| - Player2 p2 | -Player2 of the game |
| + static int[][] field | -two dimension array to define position of wall and existing brick in the game |
| - AnimationTimer animation | -animation timer to start the game |

Constructor

|  |  |
| --- | --- |
| + Stage1() | -initialize root, scene, s1, gc, p1, p2 and add p1 and p2 to root  -add s1 to root and request focus  -draw itemboard with image from ImageRef  -create border of the game screen by create hitbox and add to list hitbox of p1 and p2 then add to root  -initialize field and randomly place brick then create wall and brick (add to lbrick for brick) and add their hitbox to p1 and p2 list hitbox and root  -draw floor  -create event handler on key press  - if the KeyCode is N then stop the  background song before play the  background sound again and use  SceneManager to go to scene of  GameScreen  - if the KeyCode is ESCAPE then  exit the game  - else use KeyInput to setKeyPressed  -create event handler on key press to use KeyInput to setKeyPressed  -initialize animation to  - update p1,p2  - updateBomb of p1 and p2  - update the game  - update the item in game |

Method

|  |  |
| --- | --- |
| + void AnimationStart() | -create new thread to make a canvas that show the countdown number before the game start |
| - void update() | -check if any player is dead then stop the animation and create a canvas to show the result of the game |
| + static void drawFloor(int i, int j) | -draw floor according to the given position in field |
| + Scene getScene() | -getter of scene |
| - void updateItem() | - check if any item in litem is collision with  any player if true then called method IsKeptBy  of that item and if any item is already kept then remove it from litem  - update the item status board of each player |

6. Package object

6.1 Interface Gameobject

Method

|  |  |
| --- | --- |
| + Hitbox getHitbox() | (All objects in the game must be able to  get hitbox) |

7. Package envi

7.1 Abstract Class Envi implements Gameobject

Field

|  |  |
| --- | --- |
| # double x | -x position of the object |
| # double y | -y position of the object |

Constructor

|  |  |
| --- | --- |
| + Envi(double x, doubley) | -initialize x,y position by the given number |

Method

|  |  |
| --- | --- |
| + Hitbox getHitbox() | -getter of hitbox |

7.2 Class Wall extends Envi

Field

|  |  |
| --- | --- |
| - Hitbox hitbox | -hitbox of the wall |

Constructor

|  |  |
| --- | --- |
| + Wall(double x, double y) | -initialize x,y position by the given number  -initialize hitbox  -load the image for wall and set to hitbox fill  -set visible to the hitbox |

Method

|  |  |
| --- | --- |
| + Hitbox getHitbox() | -getter of hitbox |

7.3 Class Brick extends Envi

Field

|  |  |
| --- | --- |
| - Hitbox hitbox | -hitbox of the brick |

Constructor

|  |  |
| --- | --- |
| + Brick(double x, double y) | -initialize x,y position by the given number  -initialize hitbox  -load the image for brick and set to hitbox fill  -set visible to the hitbox |

Method

|  |  |
| --- | --- |
| + void destroyed(Group root) | -set visible of hitbox to be false  -randomly create item on the destroyed brick position, set visible to be true and add to given root and litem in Stage1 |
| + Hitbox getHitbox() | -getter of hitbox |

8. Package item

8.1 Abstract Class Item implements Gameobject

Field

|  |  |
| --- | --- |
| # double x | -x position of the item |
| # double y | -y position of the item |

Constructor

|  |  |
| --- | --- |
| + Item(double x, double y) | -initialize x,y by the given number |

Method

|  |  |
| --- | --- |
| + abstract void IsKeptBy(Hero hero) | (activate item if the item is kept by any player) |
| + abstract Hitbox getHitbox() | (getter of the hitbox) |
| + abstract boolean isKept() | (check if the item is already kept by someone) |
| + abstract void setKept(boolean k) | (setter of boolean kept) |

8.2 Class Boost extends Item

Field

|  |  |
| --- | --- |
| - Hitbox item | -hitbox of this item |
| - boolean kept | -is true if this item is already kept by someone |

Constructor

|  |  |
| --- | --- |
| + Boost(double x, double y) | -initialize x,y position by the given number  -initialize item and set fill to be the image of  the item from ImageRef |

Method

|  |  |
| --- | --- |
| + void IsKeptBy(Hero hero) | -if item hasn’t been kept by anyone then check the intersection if it is 100% (so that when increase speed the player still be able to move correctly) increaseSpeed the hero, set visible of item to be false and kept is true  -increase that player countboost by 1 |
| + Hitbox getHitbox() | -getter of item (hitbox) |
| + boolean isKept() | -getter of kept |
| + void setKept(boolean k) | -setter of kept |

8.3 Class Deboost extends Item

Field

|  |  |
| --- | --- |
| - Hitbox item | -hitbox of this item |
| - boolean kept | -is true if this item is already kept by someone |

Constructor

|  |  |
| --- | --- |
| + Deboost(double x, double y) | initialize x,y position by the given number  -initialize item and set fill to be the image of  the item from ImageRef |

Method

|  |  |
| --- | --- |
| + void IsKeptBy(Hero hero) | -if item hasn’t been kept by anyone then check the intersection if it is 100% (so that when increase speed the player still be able to move correctly) decreaseSpeed the hero, set visible of item to be false and kept is true  -decrease that player countboost by 1 if it’s more than 0 |
| + Hitbox getHitbox() | -getter of item (hitbox) |
| + boolean isKept() | -getter of kept |
| + void setKept(boolean k) | -setter of kept |

8.4 Class Stackbomb extends Item

Field

|  |  |
| --- | --- |
| - Hitbox item | -hitbox of this item |
| - boolean kept | -is true if this item is already kept by someone |

Constructor

|  |  |
| --- | --- |
| + Stackbomb(double x, double y) | initialize x,y position by the given number  -initialize item and set fill to be the image of  the item from ImageRef |

Method

|  |  |
| --- | --- |
| + void IsKeptBy(Hero hero) | -if item hasn’t been kept by anyone then increaseBombnumber the hero, set visible of item to be false and kept is true |
| + Hitbox getHitbox() | -getter of item (hitbox) |
| + boolean isKept() | -getter of kept |
| + void setKept(boolean k) | -setter of kept |

8.5 Class Debomb extends Item

Field

|  |  |
| --- | --- |
| - Hitbox item | -hitbox of this item |
| - boolean kept | -is true if this item is already kept by someone |

Constructor

|  |  |
| --- | --- |
| + Debomb(double x, double y) | initialize x,y position by the given number  -initialize item and set fill to be the image of  the item from ImageRef |

Method

|  |  |
| --- | --- |
| + void IsKeptBy(Hero hero) | -if item hasn’t been kept by anyone then decreaseBombnumber the hero, set visible of item to be false and kept is true |
| + Hitbox getHitbox() | -getter of item (hitbox) |
| + boolean isKept() | -getter of kept |
| + void setKept(boolean k) | -setter of kept |

8.6 Class Upgradebomb extends Item

Field

|  |  |
| --- | --- |
| - Hitbox item | -hitbox of this item |
| - boolean kept | -is true if this item is already kept by someone |

Constructor

|  |  |
| --- | --- |
| + Upgradebomb(double x, double y) | initialize x,y position by the given number  -initialize item and set fill to be the image of  the item from ImageRef |

Method

|  |  |
| --- | --- |
| + void IsKeptBy(Hero hero) | -if item hasn’t been kept by anyone then increaseBombrange the hero, set visible of item to be false and kept is true |
| + Hitbox getHitbox() | -getter of item (hitbox) |
| + boolean isKept() | -getter of kept |
| + void setKept(boolean k) | -setter of kept |

8.7 Class Degradebomb extends Item

Field

|  |  |
| --- | --- |
| - Hitbox item | -hitbox of this item |
| - boolean kept | -is true if this item is already kept by someone |

Constructor

|  |  |
| --- | --- |
| + Degradebomb(double x, double y) | -initialize x,y position by the given number  -initialize item and set fill to be the image of  the item from ImageRef |

Method

|  |  |
| --- | --- |
| + void IsKeptBy(Hero hero) | -if item hasn’t been kept by anyone then decreaseBombrange the hero, set visible of item to be false and kept is true |
| + Hitbox getHitbox() | -getter of item (hitbox) |
| + boolean isKept() | -getter of kept |
| + void setKept(boolean k) | -setter of kept |

9. Package entity

9.1 Class Bomb implements Gameobject

Field

|  |  |
| --- | --- |
| - boolean exploded | -true if the bomb is exploded |
| - int bombrange | -range that the bomb explode from the player that place the bomb |
| - Hitbox bomb | -hitbox of the bomb |
| - List<Hitbox> lrec | -list contains effect of the exploded bomb |
| - Thread thread | -thread when start the bomb |
| - double x | -x position of the bomb |
| - double y | -y position of the bomb |
| - boolean up | -false if the exploded bomb hit wall or brick on the up direction so that it won’t exploded though wall or brick |
| - boolean down | -false if the exploded bomb hit wall or brick on the down direction so that it won’t exploded though wall or brick |
| - boolean left | -false if the exploded bomb hit wall or brick on the left direction so that it won’t exploded though wall or brick |
| - boolean right | -false if the exploded bomb hit wall or brick on the right direction so that it won’t exploded though wall or brick |

Constructor

|  |  |
| --- | --- |
| + Bomb(double x, double y, int bombrange) | -initialize x,y position and bombrange by the given number, set exploded to be false  -initialize bomb at the x,y position and set fill to be image from ImageRef  -add this Bomb to lbomb in Stage1 |

Method

|  |  |
| --- | --- |
| + boolean IsExploded() | -getter of exploded |
| + void startBomb(Group root, Player1 p1, Player2 p2) | -add bomb to root  -create new thread that will set bomb to be visible and wait for 2.340 second then make the bomb explode by  -set the bomb to be invisible  -remove bomb from lbomb in Stage1 and  list hitbox in Player1 and Player2  -play the bomb sound from ImageRef  -create effect of the exploded bomb  which will not explode the wall and not  go through wall and brick (check from  Stage1.field) and add the effect to lrec  -if the brick is explode, update the field and  draw floor on that position  -check if any player intersect with  any effect in lrec then that player will die  -wait for 0.290 second  -check if brick is exploded then remove it  from Player1 and Player2 list hitbox  -if the game is still running then destroyed  the brick  -recheck if any player reenter to the effect  of the bomb before it disappeared |
| + Hitbox getHitbox() | -getter of bomb (hitbox) |

9.2 Abstract Class Hero implements Gameobject

Field

|  |  |
| --- | --- |
| # double speed | -speed of that player |
| - int direction | -current direction that player is facing  0 = up  1 = right  2 = down  3 = left |
| # int bombrange | -current bombrange of that player |
| # int countboost | -counter of item boost effect to that player |
| # List<Hitbox> lhb | -list contain hitbox of thing that the player can’t collision with |
| # Hitbox bounds | -hitbox of that player |
| # double x | -initial x position of that player |
| # double y | -initial y position of that player |
| # int countBomb | -current number of bomb that can be place by that player at that time |

Constructor

|  |  |
| --- | --- |
| + Hero(double x, double y) | -initialize x,y by the given number |

Method

|  |  |
| --- | --- |
| # void moveX() | -move the player in the direction the player is facing (increase or decrease x position of bound by that player speed) |
| # void moveY() | -move the player in the direction the player is facing (increase or decrease y position of bound by that player speed) |
| # boolean Collision() | -return true if that player collision with any object in lhb |
| # void DropBomb(Group root, Player1 p1, Player2 p2) | -drop the bomb in the square that the player is in or behind if the player is in between 2 squares |
| # abstract void update() | (update the move of each player) |
| + void increaseBombrange() | -increase that player bombrange by 1 |
| + void decreaseBombrange() | -decrease that player bombrange by 1 if it is more than 1 |
| + void increaseSpeed() | -increase speed from 3->4->5->6->7.5->10 |
| + void decreaseSpeed() | -decrease speed from 10->7.5->6->5->4->3->2 |
| + void increaseBombnumber() | -increase bombcount by 1 |
| + void decreaseBombnumber() | -decrease bombcount by 1 if it is more than 1 |
| + abstract boolean isAlive() | -getter of that player alive |
| + Getter method for field lhb,bounds,countboost,bombrange,countBomb,direction and Setter method for countboost and direction |  |

9.3 Class Player1 extends Hero

Field

|  |  |
| --- | --- |
| # static boolean alive | -true if player1 is still alive |
| - AnimationTimer animationwalk | -animation of player1 walking |

Constructor

|  |  |
| --- | --- |
| + Player1(double x, double y) | -initialize x,y position by the given number  -initialize bombrange,countBomb,countboost = 1, speed = 3 and alive  -set direction to be 1 (facing right)  -create bounds, set visible, set fill by image from ImageRef  -create animationwalk to set fill to a new image depends on time and direction |

Method

|  |  |
| --- | --- |
| + void update() | -if press W move up, D move right, S move down, A move left  -start animationwalk and stop when keyPressed is empty  -set direction |
| + void updateBomb(Group root, Player1 p1, Player2 p2) | - if press SPACE drop bomb if count bomb > 0  -create a thread to decrease countBomb for 3 second the increase back  -add every bomb in Stage1 lbomb to lhb if player 1 isn’t in the position that drop the bomb (catch IndexOutOfBoundsException because lbomb size may change during loop) |
| + static void setDead() | -set player1 to be not alive |
| + boolean isAlive() | -getter of alive |

9.4 Class Player2 extends Hero

Field

|  |  |
| --- | --- |
| # static boolean alive | -true if player2 is still alive |
| - AnimationTimer animationwalk | -animation of player2 walking |

Constructor

|  |  |
| --- | --- |
| + Player2(double x, double y) | -initialize x,y position by the given number  -initialize bombrange,countBomb,countboost = 1, speed = 3 and alive  -set direction to be 3 (facing left)  -create bounds, set visible, set fill by image from ImageRef  -create animationwalk to set fill to a new image depends on time and direction |

Method

|  |  |
| --- | --- |
| + void update() | -if press UP move up, RIGHT move right,  DOWN move down, LEFT move left  -start animationwalk and stop when keyPressed is empty  -set direction |
| + void updateBomb(Group root, Player1 p1, Player2 p2) | - if press SPACE drop bomb if count bomb > 0  -create a thread to decrease countBomb for 3 second the increase back  -add every bomb in Stage1 lbomb to lhb if player 2 isn’t in the position that drop the bomb (catch IndexOutOfBoundsException because lbomb size may change during loop) |
| + static void setDead() | -set player2 to be not alive |
| + boolean isAlive() | -getter of alive |