Sick Bombz - Developer’s Guide

Creator: Paul Oh

# Overview

This game is a 2 player competitive game where both players try to kill each other by planting bombs that will detonate after a delay. This game is written using Java and uses the Java GUI to display and play the game.

Both players must enter their names before starting the game.

Choose a map for both of the players to play on.

Press “Play” to start the game.

Otherwise, you can press “Credits”, “Informations” to see who developed this game and how to play the game.

When one of the players lives reach 0 during the game, that player loses the game and the opposing player wins.

There is a minigame when one of the players during the game touches a Leaf Totem. The player then is directed to a minigame with Verbal Interactions between user and program occurs. More about this minigame is explained below.

An end frame appears.

Click the “Menu” button to go back to the StartFrame.

# Features //classes

Main

* This is the main class with only the main method to initiate the game
  + This is the class that you must run to initiate the game
* It creates a start frame and sets it visible to start the game

StartFrame

* This is technically the “menu” frame and extends the JFrame class
* Variables:
  + DrawArea: mainImg, title, bomb1
    - These are different images with dev set dimensions
  + JPanel: mainPanel
    - To hold the left side of the frame
  + JLayeredPane: lpane
    - This is the entirety of the frame, where you can add panels over each other
  + JLabel: player1Label, player2Label
    - The texts that says “Player 1:” and “Player 2:”
  + JButton: play, info, cred
    - Leads to corresponding pages
  + JTextField: player1Text, player2Text
    - Where players put their names
  + JComboBox: items
    - This is the list of maps that the user can choose
    - The item on the combobox is then send through as String
* Constructor:
  + No parameters
  + Creates a 1000 x 500 JFrame that contains a cool image and title at the right
  + It contains the JTextField for both players to enter their names before starting the game
  + There are 3 JButton and 1 JComboBox on the mainPanel, type of JPanel
    - Play: Click this button to play
    - Maps: Choose the different types of maps before starting the game
    - Information: Click this button to view how to play the game
    - Credits: Click this button to view who made this game
    - mainPanel is in GridBagLayout, so GridBagConstraints should be followed
* void actionPerformed:
  + Parameter: ActionEvent: e
  + Disposes the current frame
  + If the play button was pressed
    - Open Board: window
  + Else If the info button was pressed
    - Open Information: window
  + Else //the credits button must have been pressed
    - Open Credits: window

EndFrame

* Extends JFrame and implements ActionListener
* Displays the Winner of this game using JTextArea: results
* The JTextArea: results is added to the JPanel: resultsPanel, and that JPanel is added to JLayeredPane: lpane
* lpane is added to the entire EndFrame
* setBounds are used to set where the variables
* Press JButton: menu to go back to the game
  + void actionPerformed:
    - Parameter: ActionEvent: e
    - Disposes the current frame
    - Displays the StartFrame

Board

* This is an extension of a JFrame and is the supposed “game” part of SickBombz
* A map is first created on the board with all the requirements
* The board has 20 rows and 15 columns
* Both players can move around to plant bombs and collect items
  + Players are printed on a glass pane which was added to board
* On the right side of the board, user stats are displayed and updated
* Variables
  + Field, user, user 2 - map and two Character objects
  + String filename - name to the images of the characters, changes as they face different directions
  + userBombs, user2Bombs - arraylists of bombs that keep track of each player’s bombs on the field
  + Other variables are components for the stats bar

Map

* Variables:
  + 2D array of block objects
  + Height, width for block dimensions
* Constructor:
  + Reads a saved text file for characters and creates a map according to it
  + Separates a string into an array of strings and creates a block at each index depending on what character is in the string
* In the print method, the map displays each of the blocks using g.drawImage
* Explode has a timer on 4 seconds for the bomb to detonate
* Another timer for 1 second allows the explosion to be displayed
* Explosions were modified to stop at walls or if they have broken a crate
  + Breaking a crate uses a random object to spawn items by chance
* Checks if any player is in the explosion range and reduces health if so
  + If health of any player drops to 0 the board is disposed and an EndFrame is made

DrawArea

* Extends JPanel
* This is the second type of DrawArea, the one that’s used by Paul
* There is another DrawArea contained in the Board class which is Maxx’s DrawArea
* This DrawArea accepts int: width, height, and String: name
  + The name determines the file name to get the image
* The image is loaded using BufferedImage: loadImage method
  + Parameter is String: name //the same name from constructor
  + ImageIO.read to read the image
* void paintComponent method
  + g.drawImage to display the image on the DrawArea

Bomb

* The bomb contains image, x and y locations, width and height, and range as variables
* Range is the explosion range where if the player is within it, they lose a life, and if there is a crate there, the crate disappears
* The explosion range is an integer of how many tiles it can spread over to on up, down, left, right sides
* Constructor accepts, x, y, width, height, and range and reads an image of the bomb from the Graphics file
* print method using g.drawImage to display the bomb placed on a block

Block

* Variables:
  + Boolean: breakable, crossable
  + BufferedImage: img
  + Char: type, used for saving as text file and easy identification of blocks
* Constructor:
  + Parameter: String: s
    - This string is to indicate the type of block
* There are 3 main different types of blocks
  + Wall: breakable = false; type = ‘W’; crossable = false
  + Ground: breakable = false; type = ‘G’; crossable = true
  + Crate: breakable = true; type = ‘C’; crossable = false
* For Wall, you cannot destroy or move past it
* For Ground, this is the place where players move around
* For Crate, the players cannot move past it, but they can destroy the crate if it is within the range of the bomb explosion
  + The crate also stops the bomb explosion from extending further out than the crate
  + There is a chance of spawning an item after the crate is destroyed
* For Bush, this is just a made up block when we first created the game
  + This isn’t used in any of the real maps and works as the same as the wall
* Block objects were also made for the explosion graphics (cannot be crossed) and items (can be crossed)

Character

* Variables:
  + X,y, for positon
  + Range, currentBomb, totalBomb to keep track of the character’s bombs
  + Health
* Constructor:
  + Initializes starting position based on different maps
  + Range and totalBombs also changes by map
* So each character has their own icons and own images
* There are separate images for moving left, right, up, down
  + Separate controls for each players described in the information page of the game and the user’s guide
* Move method checks the block in front of the player for crossability and items
* Stepping on items changes instance variables and updates stats bar

LeafTotem

* Verbal Interaction between user and the program
* This is a minigame that Extends JFrame and implements ActionListener
* The player must type the entire sentence with each letter an alphabet after
  + So “a” -> “b”
  + The capital words do count and any punctuation and spaces can remain the same
* The player got it right receives an extra life
* Variables:
  + String[]: s, ans - Problems and the answers
  + JTextField: user //where the user types in their answer
* Constructor:
  + Displays the directions, question, has a JTextField for an answer, JButton: enter to enter the answer
  + Directions and questions are displayed through JLabel: title, ques, prob
* void actionPerformed:
  + Goes to LeafTotemResult class after disposing this frame

LeafTotemResult

* Parameter is a boolean: right
  + This checks whether the answer was correct
* If right == false, a message says that nothing happens and the game progresses
* If right == true, a message says that the player receives one more life
* The user receives a life or not depending on their answer

Information

* Extends JFrame implements ActionListener
  + This contains the DrawArea: mainImg //background image
  + DrawArea: title //title displayed
* Displays the rules of playing this game
  + This is done by using JTextArea along with String
  + The information was written down onto the String: left, right and was setText by the JTextArea: textAreaL, textAreaR
* Press JButton: menu to go back to the game
  + void actionPerformed:
    - Parameter: ActionEvent: e
    - Disposes the current frame
    - Displays the StartFrame

Credits

* Extends JFrame implements ActionListener
  + This contains the DrawArea: mainImg //background image
  + DrawArea: title //title displayed
* Displays the developers of this game
  + JLabel: m //Maxx Wu
  + JLabel: p //Paul Oh
  + JLabel: s //Saiyam Patel
* Press JButton: menu to go back to the game
  + void actionPerformed:
    - Parameter: ActionEvent: e
    - Disposes the current frame
    - Displays the StartFrame

# Suggestions for Improvement

* More players could be added
  + 4 player mode where 4 users can play the game simultaneously
  + Extra keyboards may have to be added in
* Single player versions can be added by implementing AI players or other objectives
* More items can be added to enhance the game
  + Boots: Faster movement speed around for the players
  + Rain: All bombs detonated cannot be exploded
  + Cut Time: The bomb explodes 1 second earlier
  + Projectile items
* Detonation indicators such as blinking or changing colour can be added to bombs
* Bombs can be designed so they detonate early if hit by other bombs
* Players can kick the bombs around across the map and hit the other player to explode