/\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Bailey Thompson

\* Adventure (1.3)

\* 11 November 2016

\* Downloads: Requires jaco mp3 player

\* Info: RPG Game! Made during a hackathon.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

//declaring package

package adventure;

//declaring imports

import jaco.mp3.player.MP3Player;

import java.awt.Color;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.io.File;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

//declaring class

public class Adventure {

//used for GUI

private final JFrame frame = new JFrame("Adventure");

ImageIcon iconAdventure = new ImageIcon("Adventure.png");

JLabel text1 = new JLabel("Welcome To Adventure! This Game Was Created By Bailey Thompson");

JLabel text2 = new JLabel("To Start The Game, Please Pick A Direction To Move In!");

JLabel infoA1 = new JLabel(" Health: 20");

JLabel infoA2 = new JLabel(" Level: 1");

JButton north = new JButton("North");

JLabel infoB1 = new JLabel(" Weapon: Fists");

JLabel infoB2 = new JLabel(" Item: Nothing");

JButton west = new JButton("West"), south = new JButton("South"), east = new JButton("East");

JButton btnWeapon = new JButton("Weapon"), run = new JButton("Run/Leave"), btnItem = new JButton("Item");

JPanel textPanel = new JPanel();

JPanel infoAPanel = new JPanel();

JPanel infoBPanel = new JPanel();

JPanel panelMid = new JPanel(), panelBot = new JPanel(), panelNeg = new JPanel();

//logic variables

int xp, level = 1, currentHP = 20, weapon, item, lastMove, randomNum, encounter, half;

boolean newRound = true;

//declaring main method

public static void main(String[] args) {

//sending to method prepareGUI

Adventure Adventure = new Adventure();

Adventure.prepareGUI();

}

//private void used to prepare gui

private void prepareGUI() {

//playing the sound file n repeat

MP3Player player = new MP3Player();

player.addToPlayList(new File("Adventure.mp3"));

player.setRepeat(true);

player.play();

//creating GUI frame

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setResizable(false);

frame.setSize(400, 400);

frame.setLocationRelativeTo(null);

frame.setIconImage(iconAdventure.getImage());

frame.setVisible(true);

frame.getContentPane().setBackground(Color.gray);

//adding elements to the frame

textPanel.add(text1);

textPanel.add(text2);

textPanel.setLayout(new GridLayout(2, 1));

//adding elements to the frame

infoAPanel.add(infoA1);

infoAPanel.add(infoA2);

infoAPanel.setLayout(new GridLayout(2, 1));

//adding elements to the frame

infoBPanel.add(infoB1);

infoBPanel.add(infoB2);

infoBPanel.setLayout(new GridLayout(2, 1));

//adding elements to the frame

panelMid.add(infoAPanel);

panelMid.add(north);

panelMid.add(infoBPanel);

panelMid.setLayout(new GridLayout(1, 3));

//adding elements to the frame

panelBot.add(west);

panelBot.add(south);

panelBot.add(east);

panelBot.setLayout(new GridLayout(1, 3));

//adding elements to the frame

panelNeg.add(btnWeapon);

panelNeg.add(run);

panelNeg.add(btnItem);

panelNeg.setLayout(new GridLayout(1, 3));

//adding elements to the frame

frame.add(textPanel);

frame.add(panelMid);

frame.add(panelBot);

frame.add(panelNeg);

frame.setLayout(new GridLayout(4, 1, 0, 1));

//disabling buttons

btnWeapon.setEnabled(false);

run.setEnabled(false);

btnItem.setEnabled(false);

//sending to method buttonPress

buttonPress();

}

//method used for button press logic

private void buttonPress() {

//if user presses button north

north.addActionListener((ActionEvent e) -> {

if (newRound == true) {

randomGenerate();

updateGUI();

}

});

//if user presses button south

south.addActionListener((ActionEvent e) -> {

if (newRound == true) {

randomGenerate();

updateGUI();

}

});

//if user presses button west

west.addActionListener((ActionEvent e) -> {

if (newRound == true) {

randomGenerate();

updateGUI();

}

});

//if user presses button east

east.addActionListener((ActionEvent e) -> {

if (newRound == true) {

randomGenerate();

updateGUI();

}

});

//if user presses button weapon

btnWeapon.addActionListener((ActionEvent e) -> {

if (newRound == false) {

randomNum = ((int) (Math.random() \* ((100 - 0) + 1))) + 0;

if (randomNum > (100 - (2 \* weapon + 30))) {

//if user wins random test

newRound = true;

btnWeapon.setEnabled(false);

run.setEnabled(false);

btnItem.setEnabled(false);

north.setEnabled(true);

south.setEnabled(true);

west.setEnabled(true);

east.setEnabled(true);

xp += 1;

if (encounter == 5 || encounter == 6) {

text2.setText("You cross with your weapon! What direction do you choose now?");

weapon = 0;

} else {

text2.setText("You killed it! What direction do you choose now?");

}

} else {

//if user loses random test

text2.setText("You failed, due to exhaustion you lost 2HP. Choose another action.");

currentHP -= 2;

}

updateGUI();

}

});

//if user presses button run

run.addActionListener((ActionEvent e) -> {

if (newRound == false) {

randomNum = ((int) (Math.random() \* ((100 - 0) + 1))) + 0;

if (randomNum > 50) {

//if user wins random test

newRound = true;

btnWeapon.setEnabled(false);

run.setEnabled(false);

btnItem.setEnabled(false);

north.setEnabled(true);

south.setEnabled(true);

west.setEnabled(true);

east.setEnabled(true);

xp += 1;

if (encounter == 5 || encounter == 6) {

text2.setText("You ran, but you lost 3HP! What direction do you choose now?");

currentHP -= 3;

} else {

text2.setText("You were able to run away. What direction do you choose now?");

}

} else {

//if user loses random test

text2.setText("You are too tired to run, and lost 2HP. Choose another action.");

currentHP -= 2;

}

updateGUI();

}

});

//if user presses button item

btnItem.addActionListener((ActionEvent e) -> {

if (newRound == false) {

//determines outcome based on what item the user has

if (item > 0 && item < 6) {

if (encounter == 7) {

text2.setText("She stole your weapon! What direction do you choose now?");

weapon = 0;

newRound = true;

btnWeapon.setEnabled(false);

run.setEnabled(false);

btnItem.setEnabled(false);

north.setEnabled(true);

south.setEnabled(true);

west.setEnabled(true);

east.setEnabled(true);

xp += 1;

} else {

text2.setText("You healed yourself. Choose another action.");

currentHP += 4 \* item;

if (currentHP > 20) {

currentHP = 20;

}

}

} else if (item == 0) {

text2.setText("You have no item. Choose another action.");

} else if (encounter == 5 || encounter == 6) {

text2.setText("It had no effect. Choose another action.");

} else {

int bombDamage = ((int) (Math.random() \* ((item - 0) + 1))) + 0;

if (bombDamage > 2) {

text2.setText("You killed it! What direction do you choose now?");

newRound = true;

btnWeapon.setEnabled(false);

run.setEnabled(false);

btnItem.setEnabled(false);

north.setEnabled(true);

south.setEnabled(true);

west.setEnabled(true);

east.setEnabled(true);

xp += 1;

} else {

text2.setText("It had no effect. Choose another action.");

}

}

item = 0;

updateGUI();

}

});

}

//this method generates scenarios

private void randomGenerate() {

//setting text to blank

text2.setText("");

newRound = false;

//enabling buttons

btnWeapon.setEnabled(true);

run.setEnabled(true);

btnItem.setEnabled(true);

//disabling buttons

north.setEnabled(false);

south.setEnabled(false);

west.setEnabled(false);

east.setEnabled(false);

encounter = ((int) (Math.random() \* ((11 - 0) + 1))) + 0;

while ((item != 0 && (encounter == 8 || encounter == 10)) || (weapon != 0 && (encounter == 9 || encounter == 11))) {

encounter = ((int) (Math.random() \* ((11 - 0) + 1))) + 0;

}

if (encounter == 10 || encounter == 11) {

encounter -= 2;

}

//displaying scenario to user

switch (encounter) {

case 0:

text1.setText("You encounter a dragon. What do you do?");

break;

case 1:

text1.setText("You encounter a furious goose. What do you do?");

break;

case 2:

text1.setText("You encounter EdCom. What do you do?");

break;

case 3:

text1.setText("A wild lizard blocks your path. What do you do?");

break;

case 4:

text1.setText("You encounter Harambe. What do you do?");

break;

case 5:

text1.setText("A forest blocks your path. What do you do?");

break;

case 6:

text1.setText("Quicksand blocks your path. What do you do?");

break;

case 7:

text1.setText("A dying girl blocks your path. What do you do?");

break;

case 8:

half = ((int) (Math.random() \* ((2 - 0) + 1))) + 0;

if (half == 0) {

text1.setText("An elf gave you an item. What do you do?");

} else {

text1.setText("A sand person gave you an item. What do you do?");

}

item = ((int) (Math.random() \* ((9 - 1) + 1))) + 1;

break;

case 9:

half = ((int) (Math.random() \* ((2 - 0) + 1))) + 0;

if (half == 0) {

text1.setText("An old man gave you a weapon. What do you do?");

} else {

text1.setText("A sentient robot gave you a weapon. What do you do?");

}

weapon = ((int) (Math.random() \* ((9 - 1) + 1))) + 1;

break;

}

}

//this method updates the gui

private void updateGUI() {

if (currentHP > 0) {

//updates which weapon the user is displayed as having

switch (weapon) {

case 0:

infoB1.setText(" Weapon: Fists");

break;

case 1:

infoB1.setText(" Weapon: Stick");

break;

case 2:

infoB1.setText(" Weapon: Rock");

break;

case 3:

infoB1.setText(" Weapon: Dagger");

break;

case 4:

infoB1.setText(" Weapon: S. Sword");

break;

case 5:

infoB1.setText(" Weapon: L. Sword");

break;

case 6:

infoB1.setText(" Weapon: L. Bow");

break;

case 7:

infoB1.setText(" Weapon: R. Bow");

break;

case 8:

infoB1.setText(" Weapon: C. Bow");

break;

case 9:

infoB1.setText(" Weapon: Crossbow");

break;

}

//updates which item the user is displayed as having

switch (item) {

case 0:

infoB2.setText(" Item: Nothing");

break;

case 1:

infoB2.setText(" Item: Petty Potion");

break;

case 2:

infoB2.setText(" Item: Small Potion");

break;

case 3:

infoB2.setText(" Item: Average Potion");

break;

case 4:

infoB2.setText(" Item: Effective Potion");

break;

case 5:

infoB2.setText(" Item: Campfire");

break;

case 6:

infoB2.setText(" Item: Petty Bomb");

break;

case 7:

infoB2.setText(" Item: Small Bomb");

break;

case 8:

infoB2.setText(" Item: Average Bomb");

break;

case 9:

infoB2.setText(" Item: Effective Bomb");

break;

}

//setting the level

int previousLevel = level;

level = (int) (Math.floor((xp / 5)) + 1);

//resetting hp

if (previousLevel != level) {

currentHP = 20;

}

//updates what level the user is displayed as having

infoA1.setText(" Health: " + currentHP);

infoA2.setText(" Level: " + level);

} else {

//displaying that the user has died

text1.setText("You died. Restart the application to play again! :(");

text2.setText("On your adventure, you made it to Level " + level);

infoA1.setText(" Health: 0");

//disabling all buttons

btnWeapon.setEnabled(false);

run.setEnabled(false);

btnItem.setEnabled(false);

north.setEnabled(false);

south.setEnabled(false);

west.setEnabled(false);

east.setEnabled(false);

}

}

}