import java.awt.Graphics2D;

import java.awt.GridLayout;

import java.awt.Image;

import java.awt.MouseInfo;

import java.awt.event.KeyEvent;

import java.util.Scanner;

import javax.swing.ImageIcon;

import javax.swing.JFrame;

import javax.swing.JLabel;

public class Player extends OBJ {

int velX = 0, velY= 0;

public boolean lookRight = true;

public final static String playr = "iPictures/playerright.png";

public final static String playl = "iPictures/playerleft.png";

public final static String cland00 = "iPictures/cland00.png";

public final static String cland2 = "iPictures/cland2.png";

public final static String cland3 = "iPictures/cland3.png";

public final static String cland6 = "iPictures/cland6.png";

public final static String clandidk = "iPictures/clandidk.png";

public final static String cland5 = "iPictures/cland5.png";

public final static String cland7 = "iPictures/cland7.png";

public final static String cland8 = "iPictures/cland8.png";

public final static String cland9 = "iPictures/gigacornL.png";

public final static String cland10 = "iPictures/gigacorn.png";

public final static String cland11 = "iPictures/bossfinal.png";

public final static String cland12 = "iPictures/downfirst.png";

public final static String cland13 = "iPictures/cland13.png";

public final static String cland14 = "iPictures/cland14.png";

public final static String cland15 = "iPictures/cland15.png";

public final static String cland16 = "iPictures/cland16.png";

public final static String cland17 = "iPictures/cland17.png";

public final static String cland18 = "iPictures/cland18.png";

public final static String cland19 = "iPictures/cland19.png";

public final static String cland20 = "iPictures/bonusroom.png";

public final static String map = "iPictures/map.png";

public final static String swoll= "iPictures/Swoll Fairy.png";

public final static String swoll2= "iPictures/Swoll Fairy 2.png";

public final static String giga= "iPictures/gcorn.png";

public final static String megaroto= "iPictures/MegaRototo.png";

public final static String playerwr= "iPictures/cgnomewR.png";

public final static String playerwl= "iPictures/cgnomewL.png";

public final static String gum= "iPictures/Gumball Machine.png";

private JLabel room, bossone,bosstwo,bossthree, hpstats, currentspd,currentatk,currentdef;

private JFrame stats;

public int cland = 0;

public int hit = 0;

public String rototf= "alive",gigatf= "alive",swolltf= "alive";

public boolean gigaalive = true;

public int gigahp = 800;

public int gigamax = 800;

public int stun = 0;

public boolean rotoalive = true;

public int rotohp = 450;

public int rotomax = 450;

public int charge = 0;

public boolean swollalive = true;

public int swollhp = 3000;

public int swollmaxhp = 8000;

//playerStats

public int speed = 50; // \_\_ out of 100, chance of having first attack

public double attack =1; //multiplication factor on damage

public double defense = 1; //multiplication factor on damage taken

public int cloneattack = 0;

public boolean hitperson = true;

public int hyperlength =0;

//items

public int minorhealing = 6;//amount of health restoration pots left

public int fullhealing =3; //heals to maximum hp

public boolean requiumarrow = false;

//PlayerStatus

public boolean isPlayerTurn = false;

public int playerMaxHp = 170;

public int playerHp =170;

public int attackdmg = 0;

public int timestopswoll =0;

public int timestopplay = 0;

public boolean timeeraseab = false;

public boolean timestopab = false;

public int timeeraselength =0;

public boolean rage = false;

public int awakenlen = 4;

public boolean awakeningpot = false;

public boolean candyhpfound = false;

//0 = top, 1=right, 2= down, 3 left

public Player (int x, int y) {

super (x,y);

stats= new JFrame("Info");

stats.setSize(200,400);

room = new JLabel();

bossone = new JLabel();

bosstwo = new JLabel();

bossthree= new JLabel();

hpstats= new JLabel();

currentspd = new JLabel();

currentatk = new JLabel();

currentdef = new JLabel();

stats.add(room);

stats.add(hpstats);

stats.add(bossone);

stats.add(currentspd);

stats.add(bosstwo);

stats.add(currentdef);

stats.add(bossthree);

stats.add(currentatk);

room.setText("00");

bossone.setText("<html>"+" Megarototo: IS "+rototf+"</html>");

bosstwo.setText("<html>"+" Gigacorn: IS "+gigatf+"</html>");

bossthree.setText("<html>"+" Swoll Fairy: IS "+swolltf+"</html>");

hpstats.setText("<html>"+"170 / 170 HP"+"</html>");

currentspd.setText("<html>"+"50 / 100 Speed"+"</html>");

currentatk.setText("<html>"+"Attack Multiplier: 1"+"</html>");

currentdef.setText("<html>"+"Defense Multiplier: 1"+"</html>");

room.setVisible(true);

bossone.setVisible(true);

bosstwo.setVisible(true);

bossthree.setVisible(true);

hpstats.setVisible(true);

currentspd.setVisible(true);

currentatk.setVisible(true);

currentdef.setVisible(true);

GridLayout experimentLayout = new GridLayout(4,2);

stats.setLayout(experimentLayout);

stats.setLocation(600, 200);

stats.setResizable(false);

stats.setVisible(true);

System.out.println("Welcome to the Candyland Paradise Grounds");

System.out.println("The Paradise used to be a place of peace for candy production");

System.out.println("But 3 superpowers one day took over the Grounds and took the sweets for themselves");

System.out.println("The bosses are located in Room 9, 19, and 11. To enter room 9, the other two bosses \nmust be defeated");

System.out.println("To defeat these bosses, items are hidden within the map that can be found with ENTER.");

System.out.println("Movement is : W,A,S,D and ENTER can be used to interact with bosses, items and rooms");

System.out.println("Please restore the Paradise grounds! ");

}

public int bruhiten = 0;

public int bruhstatsindex = 0;

public String [] bruhstats;

public void updoot() {

y += velY;

x += velX;

updateInfoBar();

}

public void draw(Graphics2D g2d) {

int randomx= (int) (240 + (Math.random()\*2)-(Math.random()\*2)), randomy = (int)(50+(Math.random()\*2)-(Math.random()\*2));

g2d.drawImage(backgroundImg(),0,0,null);

g2d.drawImage(getPlayerImg(),x,y,null );

g2d.drawImage(boss(), randomx, randomy, null);

g2d.drawImage(misc(), 500 ,50 , null);

}

public void updateInfoBar() {

String clandnum = "Room #: "+cland;

if(cland == 0 ) {

clandnum = "Room #: "+1;

}

room.setText(clandnum);

if (swollalive ==true) {

swolltf = "ALIVE";

bossthree.setText("<html>"+" Swoll Fairy: IS "+swolltf+"</html>");

}else {

swolltf = "DEAD";

bossthree.setText("<html>"+" Swoll Fairy: IS "+swolltf+"</html>");

}

if (gigaalive ==true) {

gigatf = "ALIVE";

bosstwo.setText("<html>"+" Gigacorn: IS "+gigatf+"</html>");

}else {

gigatf = "DEAD";

bosstwo.setText("<html>"+" Gigacorn: IS "+gigatf+"</html>");

}

if (rotoalive ==true) {

rototf = "ALIVE";

bossone.setText("<html>"+" Megarototo: IS "+rototf+"</html>");

}else {

rototf = "DEAD";

bossone.setText("<html>"+" Megarototo: IS "+rototf+"</html>");

}

String sethp = playerHp+" / "+playerMaxHp +" HP";

hpstats.setText("<html>"+sethp+"</html>");

String setspd = speed+ " / 100 Speed";

currentspd.setText("<html>"+setspd+"</html>");

String setatk = "Attack Multiplier: "+attack;

currentatk.setText("<html>"+setatk+"</html>");

String setdef = "Defense Multiplier: "+defense;

currentdef.setText("<html>"+setdef+"</html>");

}

public Image getPlayerImg() {

ImageIcon cgnomer = new ImageIcon(playr);

ImageIcon cgnomel = new ImageIcon(playl);

ImageIcon cgnomewl = new ImageIcon(playerwl);

ImageIcon cgnomewr = new ImageIcon(playerwr);

if (lookRight == true ) {

if (cland == 17) {

return cgnomewr.getImage();

}else {

return cgnomer.getImage();

}

}else {

if (cland == 17) {

return cgnomewl.getImage();

}else {

return cgnomel.getImage();

}

}

}

public boolean angery;

public Image boss() {

ImageIcon swollashek = new ImageIcon(swoll);

ImageIcon swollashek2 = new ImageIcon(swoll2);

ImageIcon ggacorn = new ImageIcon(giga);

ImageIcon mega = new ImageIcon(megaroto);

if (cland == 9) {

angery = !angery;

if(angery == false) {

return swollashek.getImage();

}else {

return swollashek2.getImage();

}

}

if(cland ==11) {

return ggacorn.getImage();

}

if(cland== 19) {

return mega.getImage();

}else {

return null;

}

}

public Image misc() {

ImageIcon gumballs = new ImageIcon(gum);

if(cland ==13) {

return gumballs.getImage();

}else {

return null;

}

}

public Image backgroundImg() {

ImageIcon cland1 = new ImageIcon(cland00);

ImageIcon cland22 = new ImageIcon(cland2) ;

ImageIcon cland33 = new ImageIcon(cland3) ;

ImageIcon cland66 = new ImageIcon(cland6) ;

ImageIcon cland44 = new ImageIcon(clandidk) ;

ImageIcon cland55 = new ImageIcon(cland5) ;

ImageIcon cland77 = new ImageIcon(cland7) ;

ImageIcon cland88 = new ImageIcon(cland8) ;

ImageIcon cland99 = new ImageIcon(cland11) ;

ImageIcon cland1010 = new ImageIcon(cland9) ;

ImageIcon cland1111 = new ImageIcon(cland10) ;

ImageIcon cland1212 = new ImageIcon(cland12) ;

ImageIcon cland1313 = new ImageIcon(cland13) ;

ImageIcon cland1414 = new ImageIcon(cland14) ;

ImageIcon cland1515 = new ImageIcon(cland15) ;

ImageIcon cland1616 = new ImageIcon(cland16) ;

ImageIcon cland1717 = new ImageIcon(cland17) ;

ImageIcon cland1818 = new ImageIcon(cland18) ;

ImageIcon cland1919 = new ImageIcon(cland19) ;

ImageIcon cland2020 = new ImageIcon(cland20) ;

if (cland ==0) {

return cland1.getImage();

}

if (cland ==2) {

return cland22.getImage();

}

if (cland ==3) {

return cland33.getImage();

}

if (cland ==6) {

return cland66.getImage();

}

if (cland ==4) {

return cland44.getImage();

}

if (cland ==5) {

return cland55.getImage();

}

if (cland ==7) {

return cland77.getImage();

}

if (cland ==8) {

return cland88.getImage();

}

if (cland ==9) {

return cland99.getImage();

}

if (cland ==11) {

return cland1111.getImage();

}

if (cland ==12) {

return cland1212.getImage();

}

if (cland ==13) {

return cland1313.getImage();

}

if (cland ==14) {

return cland1414.getImage();

}

if (cland ==15) {

return cland1515.getImage();

}

if (cland ==16) {

return cland1616.getImage();

}

if (cland ==17) {

return cland1717.getImage();

}

if (cland ==18) {

return cland1818.getImage();

}

if (cland ==19) {

return cland1919.getImage();

}

if (cland ==20) {

return cland2020.getImage();

}

if (cland ==10) {

return cland1010.getImage();

}

else {

return null;

}

}

public boolean found = false;

public int boostnum =0;

public void keyPressed (KeyEvent e) {

int key= e.getKeyCode();

if ( key == KeyEvent.VK\_W) {

velY=-2-boostnum;

}

if(key == KeyEvent.VK\_S) {

velY = 2+boostnum;

}

if (key == KeyEvent.VK\_A) {

velX = -2-boostnum;

lookRight = false;

}

if(key == KeyEvent.VK\_D) {

velX = 2+boostnum;

lookRight = true;

}

if (key==KeyEvent.VK\_ENTER) {

checkBoosters();

}

if (x >488) {

int present = cland;

hit = 1;

cland =checkScreenBound();

if (cland !=111) {

x=0;

}else {

x=240;

y=210;

cland = present;

}

}

if (x <0) {

int present = cland;

hit = 3;

cland =checkScreenBound();

if (cland !=111) {

x=488;

}else {

x=240;

y=210;

cland = present;

}

}

if (y >488) {

int present = cland;

hit = 2;

cland =checkScreenBound();

if(cland!= 111) {

y=0;

}else {

x=240;

y=210;

cland = present;

}

}

if (y <0) {

int present = cland;

hit = 0;

cland =checkScreenBound();

if(cland!=111) {

y=488;

}

else {

x=240;

y=210;

cland = present;

}

}

}

public void checkBoosters() {

if (cland == 5 && x>23 && x<180 && y>145 && y<270) {

if(awakeningpot ==false ) {

System.out.println("Rage potion found");

System.out.println("You: This will be useful to beat the Swoll Fairy");

System.out.println("--------------------------------");

}

awakeningpot = true;

}

if(cland == 7 && x>20 && x<200 &&y>0 &&y<110 &&found == false) {

boostnum+= 2;

found= true;

System.out.println("The Crystal brought you life and your speed has increased");

System.out.println("--------------------------------");

}

if(cland == 8 && x>300 && x<450 && y>0 && y<140 ) {

cland = 20;

x=240;

y=430;

}

if(cland == 20 && x>220 && x<360 && y>450 && y<600 ) {

cland = 8;

x=310;

y=20;

}

if(cland == 9) {

bossFightSwoll();

}

if(cland == 11) {

bossFightGiga();

}

if(cland ==19) {

bossFightRoto();

playerHp = playerMaxHp;

}

if(cland==3 && x>120 && x< 400 && y>150 && y<300) {

if(requiumarrow == false) {

System.out.println("You found a Requium Arrow within the Lollipop");

System.out.println("You save it for later as it exerts a mysterious auora");

System.out.println("--------------------------------");

requiumarrow = true;

}

}

if(cland==20 && x>360 && x< 520 && y>110 && y<240) {

if (candyhpfound ==false) {

candyhpfound = true;

System.out.println("Secret Maximum HP potion found!");

System.out.println("--------------------------------");

fullhealing++;

}

}

}

private int checkScreenBound() {

if (cland == 0) {

if (hit == 0) {

return 6;

}

if (hit ==1) {

return 2;

}

if( hit ==2) {

return 111;

}

if (hit ==3) {

return 111;

}

}

if (cland == 20) {

if (hit == 0) {

return 111;

}

if (hit ==1) {

return 111;

}

if( hit ==2) {

return 111;

}

if (hit ==3) {

return 111;

}

}

if(cland == 2) {

if (hit == 0) {

return 3;

}

if (hit ==3) {

return 0;

}

if (hit == 2|| hit ==1) {

return 111;

}

}

if (cland ==3) {

if (hit ==0) {

return 4;

}

if(hit ==1) {

return 111;

}

if (hit == 2) {

return 2;

}

if(hit ==3) {

return 6;

}

}

if(cland ==6) {

if(hit == 0) {

return 111;

}

if(hit ==1) {

return 3;

}

if(hit ==2) {

return 0;

}

if(hit == 3) {

return 7;

}

}

if(cland ==8) {

if(hit == 0) {

return 111;

}

if(hit ==1) {

return 7;

}

if(hit ==2) {

return 12;

}

if(hit == 3) {

return 10;

}

}

if(cland ==11) {

if(hit == 0) {

return 111;

}

if(hit ==1) {

return 111;

}

if(hit ==2) {

return 10;

}

if(hit == 3) {

return 111;

}

}

if(cland ==7) {

if(hit == 0) {

return 111;

}

if(hit ==1) {

return 6;

}

if(hit ==2) {

return 111;

}

if(hit == 3) {

return 8;

}

}

if(cland ==12) {

if(hit == 0) {

return 8;

}

if(hit ==1) {

return 111;

}

if(hit ==2) {

return 13;

}

if(hit == 3) {

return 111;

}

}

if(cland ==13) {

if(hit == 0) {

return 12;

}

if(hit ==1) {

return 111;

}

if(hit ==2) {

return 14;

}

if(hit == 3) {

return 111;

}

}

if(cland ==14) {

if(hit == 0) {

return 13;

}

if(hit ==1) {

return 15;

}

if(hit ==2) {

return 111;

}

if(hit == 3) {

return 111;

}

}

if(cland ==15) {

if(hit == 0) {

return 111;

}

if(hit ==1) {

return 16;

}

if(hit ==2) {

return 111;

}

if(hit == 3) {

return 14;

}

}

if(cland ==16) {

if(hit == 0) {

return 111;

}

if(hit ==1) {

return 17;

}

if(hit ==2) {

return 111;

}

if(hit == 3) {

return 15;

}

}

if(cland ==17) {

if(hit == 0) {

return 111;

}

if(hit ==1) {

return 18;

}

if(hit ==2) {

return 111;

}

if(hit == 3) {

return 16;

}

}

if(cland ==18) {

if(hit == 0) {

return 19;

}

if(hit ==1) {

return 111;

}

if(hit ==2) {

return 111;

}

if(hit == 3) {

return 17;

}

}

if(cland ==19) {

if(hit == 0) {

return 111;

}

if(hit ==1) {

return 111;

}

if(hit ==2) {

return 18;

}

if(hit == 3) {

return 111;

}

}

if(cland ==4) {

if(hit == 0) {

return 5;

}

if(hit ==1) {

return 111;

}

if(hit ==2) {

return 3;

}

if(hit == 3) {

return 111;

}

}

if(cland ==10) {

if(hit == 0) {

return 11;

}

if(hit ==1) {

return 8;

}

if(hit ==2) {

return 111;

}

if(hit == 3) {

return 111;

}

}

if(cland ==5) {

if(hit == 0 ) {

if(gigaalive == false && rotoalive == false) {

System.out.println("SwollFairy: Oh? You're approaching me?");

System.out.println("You: I can't beat you without getting closer.");

return 9;

}else {

System.out.println("SwollFairy: You are no match for my Megarototo and Gigacorn...");

return 111;

}

}

if(hit ==1) {

return 111;

}

if(hit ==2) {

return 4;

}

if(hit == 3) {

return 111;

}

}

if(cland ==9) {

if(hit == 0) {

return 111;

}

if(hit ==1) {

return 111;

}

if(hit ==2) {

return 111;

}

if(hit == 3) {

return 111;

}

}

return 1;

}

public void keyReleased (KeyEvent e) {

int key= e.getKeyCode();

if(key == KeyEvent.VK\_W) {

velY =0;

}

if(key == KeyEvent.VK\_S) {

velY = 0;

}

if (key == KeyEvent.VK\_A) {

velX = 0;

}

if(key == KeyEvent.VK\_D) {

velX = 0;

}

}

public void bossFightGiga() {

if ((int) (Math.random()\*100) >speed) {

isPlayerTurn = false;

}else {

isPlayerTurn = true;

}

while (gigahp>0 && playerHp >0) {

if(isPlayerTurn == true) {

isPlayerTurn = false;

int act = 0;

Scanner se = new Scanner(System.in);

System.out.println("-------------------------------");

System.out.println("Gigacorn hp: "+gigahp);

System.out.println("Your Hp: "+playerHp+ " \n");

if(act != 5) {

System.out.println("Actions:");

System.out.println("ATTACK - 1 INSPECT- 2");

System.out.println("BOOST - 3 HEAL - 4");

act = se.nextInt();

}

if(hyperlength >0) {

hyperlength -=1;

attackdmg = (int) ((Math.random()\*30)+5);

gigahp -= attackdmg;

System.out.println("Hyperglycemia did "+ attackdmg+ " damage! ");

}

if (act == 1) {//attack

System.out.println("ATTACKS: ------------------------------");

System.out.println("Sugar Rush: damage 25-45 -1");

System.out.println("Gumball Attack: damage 10-60 - 2");

System.out.println("Clone: raise enemies chance to miss attack for 2 turns- 3");

System.out.println("Hyperglycemia: 5-30 damage for 3-6 turns -4");

Scanner attac = new Scanner (System.in);

int attackval = attac.nextInt();

if (attackval == 1) {

attackdmg = (int)(( 25+(Math.random()\*20))\*attack);

System.out.println("You did "+ attackdmg+" damage!");

gigahp -=attackdmg;

}

if (attackval ==2) {

attackdmg = (int)((10+(Math.random()\*50))\*attack);

System.out.println("You did "+ attackdmg+" damage!");

gigahp -=attackdmg;

}

if (attackval == 3) {

System.out.println("You cloned yourself! Gigacorn is dazed!");

cloneattack = 2;

}

if(attackval ==4) {

hyperlength = (int) ((Math.random()\*3)+3);

attackdmg = (int) (((Math.random()\*30)+5)\*attack);

System.out.println("Hyperglycemia has activated! You did "+attackdmg +" damage");

gigahp -= attackdmg;

}

}

if (act == 2) { //item

System.out.println("INSPECTING: Gigacorn stats");

System.out.println("Gigacorn Hp: 800");

System.out.println("Attack 1: 30 with a 20% stun rate for two turns");

System.out.println("Attack 2: 100 raw damage");

System.out.println("Ability: Armour penetration - defense is negligible");

}

if(act == 3) { //boost

System.out.println("Boost:");

System.out.println("Boost Attack - 1 Boost Defense - 2");

System.out.println("Boost Max Health - 3 Boost Speed - 4");

Scanner boost = new Scanner (System.in);

int boostpick = boost.nextInt();

if (boostpick ==1) {

attack = (attack + (attack\*0.2));

System.out.println("Attack factor is now "+attack);

}

if(boostpick == 2) {

defense = (defense+ (defense\*0.15));

if (defense >2.5) {

defense = 2.5;

}

System.out.println("Defense is now: "+defense+" /2.5");

}

if (boostpick ==3 ) {

playerMaxHp += (int) (Math.random()\*30);

System.out.println("Your max Hp is now: "+playerMaxHp +" HP" );

}

if(boostpick == 4) {

speed = (int)(speed + (speed\*0.1));

if (speed>100) {

speed= 100;

}

System.out.println("Your speed is now "+speed);

}

updateInfoBar();

}

if(act ==4) { // heal

Scanner item = new Scanner(System.in);

System.out.println("Items:");

System.out.println("Minor Healing - heals 70 hp -1 Amount: "+minorhealing);

System.out.println("Max Healing - heals to full hp -2 Amount: "+fullhealing);

int healpick = item.nextInt();

if (healpick ==1 && minorhealing >0) {

playerHp+=70;

minorhealing--;

if (playerHp>playerMaxHp) {

playerHp = playerMaxHp;

}

}

if(healpick ==2 && fullhealing >0) {

playerHp = playerMaxHp;

fullhealing--;

}

updateInfoBar();

}

}

if (isPlayerTurn == false) {

isPlayerTurn = true;

hitperson= true;

System.out.println("-------------------------------");

System.out.println("Gigacorn hp: "+gigahp);

System.out.println("Your Hp: "+playerHp+ " \n");

System.out.println( "Gigacorn turn");

if (cloneattack> 0) {

cloneattack--;

int chancev = (int) (Math.random()\*10);

if (chancev> 7) {

hitperson = false;

}else {

hitperson = true;

}

}

if(hitperson == true) {

int pickattack = (int) (Math.random()\*10);

if(gigahp >0) {

if(pickattack > 7) {

System.out.println("Gigacorn lashed with anger you took 100 damage");

playerHp-= 100;

System.out.println("You are now at " + playerHp+" hp" );

}else {

attackdmg = 30;

System.out.println("Gigacorn attacked! You took "+attackdmg+" damage.");

playerHp-=attackdmg;

int stunned = (int)(Math.random()\*10);

if( stunned >8) {

stun = 2;

}

}

}

updateInfoBar();

}

if(hitperson == false) {

System.out.println("Gigacorn aimed at your clone and attacked!");

System.out.println("Gigacorn is confused!");

}

}

}

if(playerHp <= 0) {

System.out.println("Game Over.");

System.exit(0);

}

if(gigahp <=0) {

System.out.println("Rainbows... rainbows... i see... rainbows");

gigaalive = false;

}

}

public void bossFightRoto() {

if ((int) (Math.random()\*100) >speed) {

isPlayerTurn = false;

}else {

isPlayerTurn = true;

}

while (rotohp>0 && playerHp >0) {

if(isPlayerTurn == true) {

isPlayerTurn = false;

int act = 0;

Scanner s = new Scanner(System.in);

System.out.println("-------------------------------");

System.out.println("Megarototo hp: "+rotohp);

System.out.println("Your Hp: "+playerHp+ " \n");

if(act != 5) {

System.out.println("Actions:");

System.out.println("ATTACK - 1 INSPECT- 2");

System.out.println("BOOST - 3 HEAL - 4");

act = s.nextInt();

}

if(hyperlength >0) {

hyperlength -=1;

attackdmg = (int) ((Math.random()\*30)+5);

rotohp -= attackdmg;

System.out.println("Hyperglycemia did "+ attackdmg+ " damage! ");

}

if (act == 1) {//attack

System.out.println("ATTACKS: ------------------------------");

System.out.println("Sugar Rush: damage 25-45 -1");

System.out.println("Gumball Attack: damage 10-60 - 2");

System.out.println("Clone: raise enemies chance to miss attack for 2 turns- 3");

System.out.println("Hyperglycemia: 5-30 damage for 3-6 turns -4");

Scanner attac = new Scanner (System.in);

int attackval = attac.nextInt();

if (attackval == 1) {

attackdmg = (int)(( 25+(Math.random()\*20))\*attack);

System.out.println("You did "+ attackdmg+" damage!");

rotohp -=attackdmg;

}

if (attackval ==2) {

attackdmg = (int)((10+(Math.random()\*50))\*attack);

System.out.println("You did "+ attackdmg+" damage!");

rotohp -=attackdmg;

}

if (attackval == 3) {

System.out.println("You cloned yourself! Megarototo is confused!");

cloneattack = 2;

}

if(attackval ==4) {

hyperlength = (int) ((Math.random()\*3)+3);

attackdmg = (int) (((Math.random()\*30)+5)\*attack);

System.out.println("Hyperglycemia has activated! You did "+attackdmg +" damage");

rotohp -= attackdmg;

}

}

if (act == 2) { //item

System.out.println("INSPECTING: Rototo stats");

System.out.println("Rototo Hp: 450");

System.out.println("Attack 1: 10-40 damage");

System.out.println("Attack 2: 400 damage, requires 4 charges");

System.out.println("Ability: Regeneration - heals 20 hp every turn");

}

if(act == 3) { //boost

System.out.println("Boost:");

System.out.println("Boost Attack - 1 Boost Defense - 2");

System.out.println("Boost Max Health - 3 Boost Speed - 4");

Scanner boost = new Scanner (System.in);

int boostpick = boost.nextInt();

if (boostpick ==1) {

attack = (attack + (attack\*0.2));

System.out.println("Attack factor is now "+attack);

}

if(boostpick == 2) {

defense = (defense+ (defense\*0.15));

if (defense >1.9) {

defense = 1.9;

}

System.out.println("Defense is now: "+defense+" /1.9");

}

if (boostpick ==3 ) {

playerMaxHp += (int) (Math.random()\*30);

System.out.println("Your max Hp is now: "+playerMaxHp +" HP" );

}

if(boostpick == 4) {

speed = (int)(speed + (speed\*0.1));

if (speed>100) {

speed= 100;

}

System.out.println("Your speed is now "+speed);

}

updateInfoBar();

}

if(act ==4) { // heal

Scanner item = new Scanner(System.in);

System.out.println("Items:");

System.out.println("Minor Healing - heals 70 hp -1 Amount: "+minorhealing);

System.out.println("Max Healing - heals to full hp -2 Amount: "+fullhealing);

int healpick = item.nextInt();

if (healpick ==1 && minorhealing >0) {

playerHp+=70;

minorhealing--;

if (playerHp>playerMaxHp) {

playerHp = playerMaxHp;

}

}

if(healpick ==2 && fullhealing >0) {

playerHp = playerMaxHp;

fullhealing--;

}

updateInfoBar();

}

if (act == 5) {

System.out.println("Time has been stopped");

System.out.println("You stand with fear as you can't do anything");

}

}

if (isPlayerTurn == false) {

isPlayerTurn = true;

hitperson= true;

System.out.println("-------------------------------");

System.out.println("Megarototo hp: "+rotohp);

System.out.println("Your Hp: "+playerHp+ " \n");

System.out.println( "Megarototo turn");

if (rotohp<rotomax) {

rotohp +=20;

System.out.println("Rototo healed 20 hp!");

if(rotohp>rotomax) {

rotohp = rotomax;

}

System.out.println("Megarototo hp: "+rotohp);

}

if (cloneattack> 0) {

cloneattack--;

int chancev = (int) (Math.random()\*10);

if (chancev> 7) {

hitperson = false;

}else {

hitperson = true;

}

}

if(hitperson == true) {

if(charge == 4) {

if(rotohp>0) {

int sub = (int) (300/defense);

playerHp -=sub;

charge = 0;

System.out.println("Omae wa mou shindeiru. Adios. "+sub+ " Damage was dealt.");

}else {

int subt = (int) (300/defense);

playerHp -=subt;

System.out.println("You will go down with me...\n Megarototo concentrated and with his last energy, dealt " +subt+" damage." );

}

}else {

int pickattack = (int) (Math.random()\*10);

if(rotohp >0) {

if(pickattack > 6) {

System.out.println("Rototo is concentrating and charging energy. Quite a frightening sight.");

charge++;

}else {

attackdmg = (int) (((Math.random()\*30)+10)/defense);

System.out.println("Rototo attacked! Its a sweet surprise! You took "+attackdmg+" damage.");

playerHp-=attackdmg;

}

}else {

attackdmg = (int) (((Math.random()\*10)+10)/defense);

System.out.println("Rototo used his final bit of energy to attack! You took "+attackdmg+" damage.");

playerHp-=attackdmg;

System.out.println("You stand as a stronger warrior ready to face the Swoll Fairy. You have "+playerHp+ " health left.");

}

}

updateInfoBar();

}

if(hitperson == false) {

System.out.println("Megarototo aimed at your clone and attacked!");

System.out.println("Megarototo is confused!");

}

}

}

if(playerHp <= 0) {

System.out.println("Game Over.");

System.exit(0);

}

if(rotohp <=0) {

System.out.println("You will... not... succeed");

rotoalive = false;

}

if (rotohp <0 && playerHp<0) {

System.out.println("Megarototo: You fought well... but It seems like you also cannot fight");

System.out.println("Your consciousness fades and you slowly feel your energy slipping away");

}

}

public void bossFightSwoll() {

if ((int) (Math.random()\*100) >speed) {

isPlayerTurn = false;

}else {

isPlayerTurn = true;

}

while (swollhp>0 && playerHp >0) {

if(isPlayerTurn == true) {

isPlayerTurn = false;

int act = 0;

Scanner s = new Scanner(System.in);

System.out.println("-------------------------------");

System.out.println("Swoll Fairy hp: "+swollhp);

System.out.println("Your Hp: "+playerHp+ " \n");

if (timestopswoll >0) {

System.out.println("Time has been stopped");

System.out.println("You stand with fear as you can't do anything");

timestopswoll--;

if(timestopab == true && timestopswoll >0 ) {

System.out.println("You start moving in stopped time. The Swoll Fairy is shocked");

timestopswoll = 0;

timestopplay= (int)(Math.random()\*2)+2;

System.out.println("Swoll Fairies time resumes, and you stop time for "+timestopplay+ " turns");

}

}

if(timestopswoll == 0) {

System.out.println("Actions:");

System.out.println("ATTACK - 1 INSPECT- 2");

System.out.println("BOOST - 3 HEAL - 4");

act = s.nextInt();

}

if(hyperlength >0) {

hyperlength -=1;

attackdmg = (int) (((Math.random()\*30)+5)\*(attack/12));

if ( awakenlen >0) {

attackdmg = attackdmg\*10;

}

swollhp -= attackdmg;

System.out.println("Hyperglycemia did "+ attackdmg+ " damage! ");

}

if (act == 1) {//attack

System.out.println("ATTACKS: ------------------------------");

System.out.println("Sugar Rush: damage 25-45 -1");

System.out.println("Gumball Attack: damage 10-60 - 2");

System.out.println("Clone: raise enemies chance to miss attack for 2 turns- 3");

System.out.println("Hyperglycemia: 5-30 damage for 3-6 turns -4");

Scanner attac = new Scanner (System.in);

int attackval = attac.nextInt();

if (attackval == 1) {

attackdmg = (int)(( 25+(Math.random()\*20))\*(attack/12));

if (awakenlen >0) {

attackdmg = attackdmg\*10;

awakenlen--;

}

System.out.println("You did "+ attackdmg+" damage!");

swollhp -=attackdmg;

}

if (attackval ==2) {

attackdmg = (int)((10+(Math.random()\*50))\*(attack/12));

if (awakenlen >0) {

attackdmg = attackdmg\*10;

awakenlen--;

}

System.out.println("You did "+ attackdmg+" damage!");

swollhp -=attackdmg;

}

if (attackval == 3) {

System.out.println("You cloned yourself! The Swoll fairy is confused!");

cloneattack = 2;

}

if(attackval ==4) {

hyperlength = (int) ((Math.random()\*3)+3);

attackdmg = (int) (((Math.random()\*30)+5)\*(attack/6));

if (awakenlen >0) {

attackdmg = attackdmg\*10;

awakenlen--;

}

System.out.println("Hyperglycemia has activated! You did "+attackdmg +" damage");

swollhp -= attackdmg;

}

}

if (act == 2) { //item

System.out.println("INSPECTING: Swoll Fairy stats");

System.out.println("Swoll Fairy Hp: 2000");

System.out.println("Attack 1: Raging flurry- multiple strong punches Damage 15-50 per hit.");

System.out.println("Attack 2: Candy Explosion - a explosive gumball does 120 Damage");

System.out.println("Ability 1: The World- Time stop for 2-3 turns");

System.out.println("Ability2 : Extreme rage- Hp gets restored to 5000, attack damage is increased");

System.out.println("Note: The Swoll Fairy's candy coated armour allows attack factorials\n to be reduced to 12 times it's original amount ");

}

if(act == 3) { //boost

System.out.println("Boost:");

System.out.println("Boost Attack - 1 Boost Defense - 2");

System.out.println("Boost Max Health - 3 Boost Speed - 4");

Scanner boost = new Scanner (System.in);

int boostpick = boost.nextInt();

if (boostpick ==1) {

attack = (attack + (attack\*0.3));

System.out.println("Attack factor is now "+attack);

}

if(boostpick == 2) {

defense = (defense+ (defense\*0.15));

if (defense >15) {

defense = 15;

}

System.out.println("Defense is now "+defense);

}

if (boostpick ==3 ) {

playerMaxHp += (int) (Math.random()\*70);

System.out.println("Your max Hp is now: "+playerMaxHp +" HP" );

}

if(boostpick == 4) {

speed = (int)(speed + (speed\*0.1));

if (speed>100) {

speed= 100;

}

System.out.println("Your speed is now "+speed);

}

updateInfoBar();

}

if(act ==4) { // heal

Scanner item = new Scanner(System.in);

System.out.println("Items:");

System.out.println("Minor Healing - heals 70 hp -1 Amount: "+minorhealing);

System.out.println("Max Healing - heals to full hp -2 Amount: "+fullhealing);

if(requiumarrow == true ) {

System.out.println("Requium Arrow- grants one special ability- 3");

}

if(awakeningpot == true) {

System.out.println("Awakening Potion- gives you inhuman powers for 4 turns -4");

}

int healpick = item.nextInt();

if (healpick ==1 && minorhealing >0) {

playerHp+=70;

minorhealing--;

if (playerHp>playerMaxHp) {

playerHp = playerMaxHp;

}

}

if(healpick ==2 && fullhealing >0) {

playerHp = playerMaxHp;

fullhealing--;

}

if(healpick ==3 && requiumarrow == true) {

requiumarrow = false;

System.out.println("You take a deep breath and hold the golden arrow");

System.out.println("You embrace it with your soul, and beg for a miracle");

System.out.println("With "+ playerHp+ " left, your resolve stays strong");

System.out.println("You can feel the power of the arrow, and are given a choice");

System.out.println("The ability to move within stopped time -1");

System.out.println("The ability to erase time -2");

Scanner choose = new Scanner (System.in);

int choice = choose.nextInt();

if (choice == 1) {

System.out.println("You are now able to move within stopped time.\n Your determination to win is stronger than ever.");

timestopab = true;

}

if(choice ==2 ) {

System.out.println("You now have the ability to undo the Swoll fairy's actions every 2 turns");

timeeraseab = true;

}

}

if (healpick == 4 && awakeningpot == true) {

awakeningpot = false;

}

timestopplay --;

updateInfoBar();

}

}

if (isPlayerTurn == false) {

if (timestopplay <= 0) {

timeeraselength--;

isPlayerTurn = true;

hitperson= true;

System.out.println("-------------------------------");

System.out.println("Swoll Fairy hp: "+swollhp);

System.out.println("Your Hp: "+playerHp+ " \n");

System.out.println( "Swoll fairy turn");

if (cloneattack> 0) {

cloneattack--;

int chancev = (int) (Math.random()\*10);

if (chancev> 7) {

hitperson = false;

}else {

hitperson = true;

}

}

if(hitperson == true) {

int pickattack = (int) (Math.random()\*10);

if(swollhp >500) {

if(pickattack > 5 && timestopswoll <= 0) {

int length = (int)(Math.random()\*1.9) +2;

timestopswoll = length;

System.out.println("THE WORLD! TIME HAS STOPPED. "+timestopswoll+" TURNS REMAINING BEFORE TIME RESUMES");

}else {

if(pickattack >5) {

int totaldmg=0;

for (int i = 0; i< (int)(Math.random()\*4.9)+3; i++ ) {

System.out.println("MUDA ");

attackdmg = (int) (Math.random()\*35) +15;

if(rage == true) {

attackdmg = (int) (attackdmg\*(Math.random()+1));

}

System.out.println(" (Damage: "+attackdmg+")");

playerHp -= attackdmg;

totaldmg += attackdmg;

}

System.out.println("You took "+totaldmg+" damage");

timestopswoll --;

if (timeeraselength <=0 &&timeeraseab == true) {

timeeraselength =2;

playerHp+= totaldmg;

System.out.println("You erased time and the attack never existed. Your health is back to "+ playerHp);

}

}else {

timestopswoll--;

System.out.println("EMBRACE FOR THE CANDY EXPLOSION");

attackdmg = ((int)( 120/defense));

if (rage == true) {

attackdmg = (int) (attackdmg\*(Math.random()+1));

}

System.out.println("The explosion did "+attackdmg+" damage.");

playerHp-=attackdmg;

if (timeeraselength ==0 &&timeeraseab == true) {

timeeraselength = 2;

playerHp+= attackdmg;

System.out.println("You erased time and the attack never existed. Your health is back to "+ playerHp);

}

}

}

}

}

if(rage ==false && swollhp <= 500) {

System.out.println("I WILL NOT LOSE... THIS IS THE START OF THE END");

swollhp = swollmaxhp;

rage= true;

}

}else {

if(timestopplay>0) {

System.out.println("The Swoll fairy can't do anything in stopped time");

}

}

}

if(hitperson == false) {

System.out.println("Swoll fairy aimed at your clone and attacked!");

System.out.println("Swoll fairy missed the attack and is confused!");

}

}

if(playerHp <= 0) {

System.out.println("Game Over.");

System.exit(0);

}

if(swollhp <=0) {

System.out.println("You will never control... these lands... ");

swollalive = false;

}

}

}