**JAVA를 이용한 마우스 피하기 프로그램의 구현**

**155111174 남지훈**

**대구가톨릭대학교 컴퓨터공학전공**

1. 서 론

이번 2학년 1학기 마지막 과제인 텀프로젝트에 ‘마우스 피하기’라는 게임을 만들었고, 이에 관한 관련 연구, 프로그램을 어떻게 구상했는지, 어떻게 구현했는지와 어떻게 동작하는 지를 보여주고, 마지막 결론까지 추려서 설명을 할 것이다.

1. 관련 연구

프로젝트 수행과 관련한 기타 이론이나 내용들을 정리한다.

패키지안에 다수의 클래스를 두었고, 마우스를 이용해서 하는 게임이므로 프레임을 사용하였다. 장애물이 나오는 위치도 지정하고, 마우스와 장애물이 닿으면 어떻게 되는지도 구현되었다.

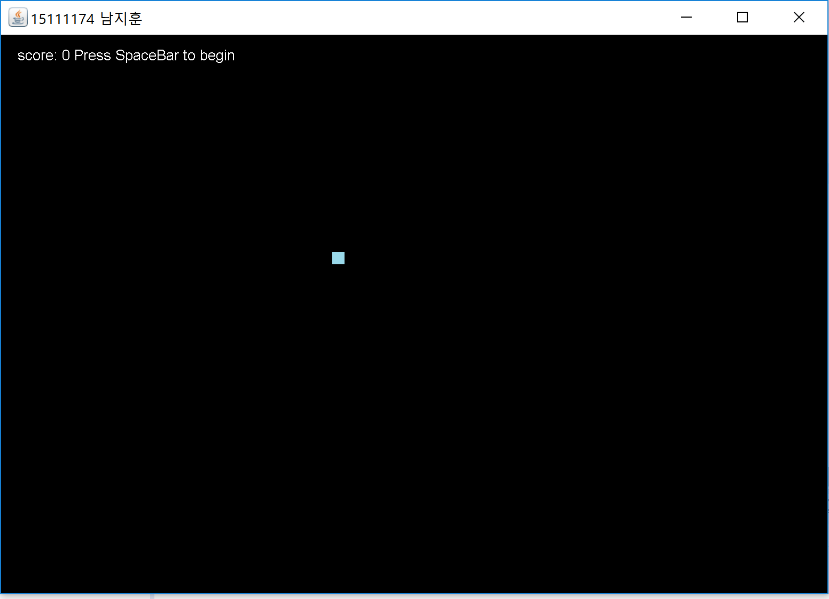
1. 프로그램 설계

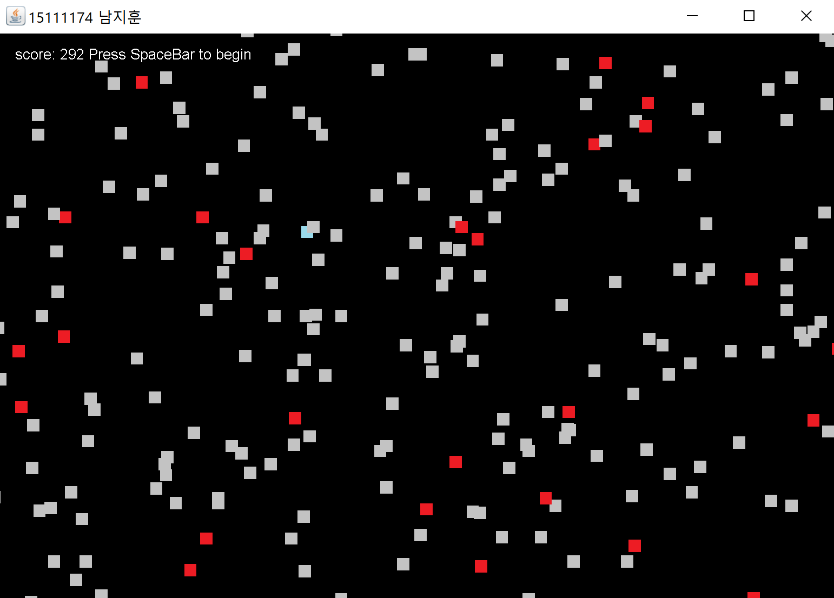
프로젝트를 수행하면서 프로그램을 구성하기 위한 계획과 기획, 구성들에 대해 설명한다.

텀프로젝트 과제가 주어지고, 이틀 정도 고민을 하였다. C언어, C++언어 텀프로젝트때는 누구나 쉽게, 금방 금방 만들 수 있는 프로그램을 만들었었다. 되새겨 보니 그런 프로그램을 만들면, 쉽게 금방 만들 수 있지만, 그것은 그저 과제를 수행하기 위한 것뿐이라 생각했고, 결국 궁극적인 목적인 나에게 피와 살이 될 수 있는 공부는 되지 않는 느낌이었다, 그래서 이번만큼은 누구나 쉽게 만들 수 있는 프로그램이 아니고, 그렇다고 결코 난이도가 쉽지만은 않으면서, 확실히 공부가 될 수 있는 프로그램을 만들고 싶었다. 그렇게 며칠을 고민하다가 우연히 유튜브에서 마우스 피하기라는 게임을 보게 되었다. 그 게임을 보자마자 나는 ‘저거다!’라고 생각하며 무조건 만들 것이라고 마음먹었다. 하지만 실행에 옮겼지만, 어떻게 할지 갈피조차 잡지 못했다. 과제를 기한 내에 내지 못하겠다고 생각도 했다. 그래도 남자답게 나에게 도움되는 공부한 셈 치자고 마음먹고 편하게 시작했던 것 같다. 서론이 길었는데, 본론으로 가자면, 앞서 말한 것처럼 ‘마우스 피하기’라는 게임을 만들어야 했기 때문에 프레임을 이용하여 시각적인 부분을 구현하였다. 나의 캐릭터와 장애물을 만들고, 장애물에 닿으면 게임이 끝나는 방향으로 설계하였다.

1. 프로그램 구현

위에서 말한 것처럼 프레임을 이용하였고, 나의 캐릭터와 장애물을 이미지를 이용하여 표현하였다. 그리고 게임을 시작할 수 있는 버튼을 만들고, 게임이 시작되면 장애물들이 사방에서 튀어나오게 하였고, 난이도 조절을 위해 장애물을 2가지로 만들었다. 그리고 장애물에 나의 캐릭터가 닿는 순간 게임이 끝이 나도록 하였고, 게임시간도 표시되게 구현하였다.

1. 프로그램의 동작과 결과



1. 결과 분석 및 토의

5장의 동작 결과가 정상적으로 수행된 것인지.. 이상은 없는지.. 완벽한 것인지.. 하나 하나 면밀히 검토하여 분석하고 토의한다.

내가 원하는 대로 게임 시작 버튼을 누르면 사방에서 장애물이 튀어나오고,

시간 체크가 시작된다. 장애물은 비교적 느린 수준과, 비교적 빠른 수준 2가지 장애물이 프레임을 휘젓고 다닌다. 마우스가 장애물에 닿지 않게 피해 다니다가, 장애물과 닿으면

그대로 게임과 경과 시간이 멈춘다.

1. 결론

보고서의 결론을 맺고, 프로젝트를 하면서 느낀 점과 향후 자신의 프로그래밍에 대한 짧은 소견을 밝힌다.

“하얗게 불태웠다.”라는 말 밖에 떠오르지 않는다. 처음 과제를 받고 나서 무엇을 만들지 골머리를 싸맨 기억과 우연히 접한 유튜브가 날 여기까지 이끌어준 것, 또 프로젝트를 진행하면서 프로그램을 짰다가 다시 갈아엎었다가를 몇 번이나 했는지 모르겠다. 평소 내가 하던 공부가 ‘부족했구나’를 새삼 느꼈다. 하지만, 이번만큼은 포기하기 싫어서, 나에게 남아있을지도 모르는 마지막 악바리 정신으로 끝까지 만들어 보고자

프로젝트 마지막 날까지 골머리를 싸매면서 결국 완성시켰다.

근 한달동안 프로그램을 만들면서 공부도 엄청 많이 되었고, 평소 나는 프로그래밍에 재능이 없다고 늘 생각해왔지만, ‘할 수 있다’ 라는 자신감을 가지게 되었다.

1. 참고문헌

[1] 황기태, 김효수, “명품 JAVA Programming”, 생능출판사, 2013.

[2] 신용권, “이것이 자바다”, 한빛미디어, 2015.

[3] 윤성우, “열혈C++프로그래밍” , 오렌지미디어, 2010.

[4] <https://www.opentutorials.org/> 오픈튜토리얼

[5] <https://stackoverflow.com/> 스택오버플로우

[6] 생활코딩

1. 별첨 (소스코드와 주석)

MAIN

package Avoid\_Mouse;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

import java.awt.event.MouseEvent;

import java.awt.event.MouseMotionListener;

import java.util.Random;

public class Main extends JFrame implements Runnable, MouseMotionListener, KeyListener {

private int frameWidth = 700;

private int frameHeight = 500;

final int MAX\_MONSTER = 500;

private Hero Hero = new Hero();

private Monster[] Monster = new Monster[MAX\_MONSTER];

private long MonsterCreationTime = 0;

private int Score = 0;

private Random random = new Random();

private Thread thread;

public Main() {

init();

setLayout(null);

setTitle("15111174 남지훈");

setBounds(200, 200, frameWidth, frameHeight);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// 윈도우 종료시 종료

setVisible(true);

thread = new Thread(this);

thread.start();

addMouseMotionListener(this);

addKeyListener(this);

}

public void run() {

while (true) {

update();

repaint();

try {

Thread.sleep(50);

} catch (InterruptedException e) {

System.out.println("Error in executing thread: " + e);

}

}

}

public void init() {

Hero.init();

}

public void update() {

if (Hero.getLifeState() == Hero.DEAD)

return;

Hero.update();

for (int i = 0; i < MAX\_MONSTER; i++) {

if (Monster[i] != null) {

Monster[i].update(Hero);

if (Monster[i].getLifeState() == Monster[i].DEAD) {

Monster[i] = null;

}

}

}

if (MonsterCreationTime <= System.currentTimeMillis() - 20) {

Score++;

MonsterCreationTime = System.currentTimeMillis();

for (int i = 0; i < MAX\_MONSTER; i++) {

// 죽은 몬스터 체크

if (Monster[i] == null) {

int nRandomMonster = random.nextInt(5);

if (nRandomMonster == 0) {

Monster[i] = new FastMonster();

} else {

Monster[i] = new NormalMonster();

}

Monster[i].init();

int nRandomAppear = random.nextInt(4);

if (nRandomAppear == 0) {

Monster[i].appear(-5, random.nextInt(500));

} else if (nRandomAppear == 1) {

Monster[i].appear(705, random.nextInt(500));

} else if (nRandomAppear == 2) {

Monster[i].appear(random.nextInt(700), 505);

} else if (nRandomAppear == 3) {

Monster[i].appear(random.nextInt(700), -5);

}

break;

}

}

}

}

public void paint(Graphics g) {

g.setColor(Color.black);

g.fillRect(0, 0, frameWidth, frameHeight);

Hero.render(g);

for (int i = 0; i < MAX\_MONSTER; i++) {

if (Monster[i] != null) {

Monster[i].render(g);

}

}

g.setColor(Color.white);

g.drawString("score: " + Score + " Press SpaceBar to begin", 20, 50);

}

public static void main(String[] args) {

new Main();

}

public void mouseMoved(MouseEvent e) {

Hero.setPos(e.getX(), e.getY());

}

@Override

public void mouseDragged(MouseEvent arg0) {

}

@Override

public void keyPressed(KeyEvent arg0) {

}

@Override

public void keyReleased(KeyEvent arg0) {

if (arg0.getKeyCode() == 32) {

Hero.setLifeState(Hero.ALIVE);

for (int i = 0; i < MAX\_MONSTER; i++) {

Monster[i] = null;

}

Score = 0;

}

}

@Override

public void keyTyped(KeyEvent arg0) {

}

}

HERO

package Avoid\_Mouse;

import java.awt.Graphics;

import java.awt.Image;

import javax.swing.ImageIcon;

public class Hero {

private Image imgHero;

private int posX;

private int posY;

private int lifeState;

final int ALIVE = 0;

final int DEAD = 1;

public void init(){

imgHero = new ImageIcon("Hero.png").getImage();

posX = 0;

posY = 0;

lifeState = DEAD;

}

public void update(){

}

public void render(Graphics g){

g.drawImage(imgHero, posX-5, posY-5, null);

}

public void setPos(int \_posX, int \_posY){

posX = \_posX;

posY = \_posY;

}

public void setLifeState(int \_lifeState){

lifeState = \_lifeState;

}

public int getLifeState(){

return lifeState;

}

public int getPosX(){

return posX;

}

public int getPosY(){

return posY;

}

}

MONSTER

package Avoid\_Mouse;

import java.awt.Graphics;

import java.awt.Image;

import java.util.Random;

import javax.swing.ImageIcon;

public class Monster {

protected Image imgMonster;

protected float posX;

protected float posY;

protected int target\_posX;

protected int target\_posY;

protected double target\_angle;

private int lifeState;

final int ALIVE = 0;

final int DEAD = 1;

private Random random = new Random();

public void init(){

posX = 0;

posY = 0;

lifeState = DEAD;

}

public void update(Hero hero){

if(posX <-10 || posX > 710 || posY > 510 || posY < -10)

{

lifeState = DEAD;

}

//충돌했는지 확인

float horizontalDistance = hero.getPosX()-posX;

float verticalDistance = hero.getPosY()-posY;

float distance = (float) Math.sqrt(horizontalDistance\*horizontalDistance + verticalDistance\*verticalDistance);

if(distance <= 10){

hero.setLifeState(hero.DEAD);

}

}

public void appear(int \_posX, int \_posY){

posX = \_posX;

posY = \_posY;

lifeState = ALIVE;

target\_posX = random.nextInt(700);

target\_posY = random.nextInt(500);

target\_angle = Math.atan2(target\_posY-posY, target\_posX-posX);

}

public void render(Graphics g){

g.drawImage(imgMonster, (int)posX-5, (int)posY-5, null);

}

public int getLifeState(){

return lifeState;

}

public float getPosX(){

return posX;

}

public float getPosY(){

return posY;

}

}

NORMALMONSTER

package Avoid\_Mouse;

import javax.swing.ImageIcon;

public class NormalMonster extends Monster {

public void init(){

super.init();

imgMonster = new ImageIcon("Monster.png").getImage();

}

public void update(Hero hero){

super.update(hero);

posX += Math.*cos*(target\_angle)\*2;

posY += Math.*sin*(target\_angle)\*2;

}

}

FASTMONSTER

package Avoid\_Mouse;

import javax.swing.ImageIcon;

public class FastMonster extends Monster {

public void init(){

super.init();

imgMonster = new ImageIcon("FastMonster.png").getImage();

}

public void update(Hero hero){

super.update(hero);

posX += Math.*cos*(target\_angle)\*5;

posY += Math.*sin*(target\_angle)\*5;

}

}