

실습 레포트

실습명: 텀 프로젝트 (12 주차 실습)

2022-2 학기 컴퓨터프로그래밍 및 실습
이름: 김건하
학과: 컴퓨터.전자시스템공학부
학번: 201900532

2022. 11. 28

1. 총알 피하기

- 코드:

```
#include<SFML/Graphics.hpp>
#include<iostream>
#include<string>
#include<vector>
#include<time.h>
#include<cmath>
#include<windows.h>
#include"MyCircle.h"

usingnamespacestd;
usingnamespacestd;

intmain() {
    cout <<"hello world"<<endl;
    intnX = 1600; // display size
    intnY = 900;
    RenderWindowwindow(VideoMode(nX, nY), "Moving Ball");
    window.setFramerateLimit(100);

    // Player circle info
    doublepRadius = 10;
    doublepPosX = 800;
    doublepPosY = 450;
    doublepVelocity = 4;

    // Enemy circle info
    doubleeRadius = 3;
    doubleeVelocity = 2;

    MyCircleplayer{ pPosX, pPosY, pRadius }; // set player circle
    player.setVelocity(pVelocity);
    player.setColor(52, 204, 255); // player color set blue

    MyCircleenemy{ 0, 0, eRadius }; // set enemy circle
    enemy.setColor(255, 255, 0); // enemy color set yellow

    // enemy list
    intenemyNum = 20; // enemy number
    vector<MyCircle> enemyLst;
    for(inti = 0; i < enemyNum; i++) {
        enemy.setPosition(10 + i * 80, 10);
        enemy.setVelocity(eVelocity);
        enemyLst.push_back(enemy);
    }

    TexttTime; // display time
    TexttEnemy; // display enemy

    Fontfont;
    intt = 0;
    inte = 0;

    if(!font.loadFromFile("C:\\Users\\wqkqcu\\source\\repos\\Bullet Dodge\\Bullet Dodge\\arial.ttf")) { // check font file route!
        //C:\\Users\\wqkqcu\\source\\repos\\Bullet Dodge\\Bullet Dodge\\arial.ttf"
        return42; // Robust error handling!
    }

    // time text set
    tTime.setFont(font);
    tTime.setCharacterSize(25);
```

```

tTime.setFill(Color::White);
tTime.setPosition(1525, 860);

// enemy num text set
tEnemy.setFont(font);
tEnemy.setCharacterSize(25);
tEnemy.setFill(Color::Magenta);
tEnemy.setPosition(0, 0);

clock_time = clock();

// game loop
while(window.isOpen()) {
    // check event
    Event e;
    while(window.pollEvent(e)) {
        if(e.type == Event::Closed)
            window.close();
    }

    // move player circle by keyboard
    if(Keyboard::isKeyPressed(Keyboard::Up)) {
        player.move(0, -player.getVelocity());
    }
    elseif(Keyboard::isKeyPressed(Keyboard::Down)) {
        player.move(0, player.getVelocity());
    }
    if(Keyboard::isKeyPressed(Keyboard::Left)) {
        player.move(-player.getVelocity(), 0);
    }
    elseif(Keyboard::isKeyPressed(Keyboard::Right)) {
        player.move(player.getVelocity(), 0);
    }

    // move enemy
    for(float dt = 0; dt < 1; dt += 0.001) {

    }

    time = clock();
    time = time / CLOCKS_PER_SEC;
    tTime.setString(to_string(time) + " sec");
    tEnemy.setString("Enemy: " + to_string(enemyNum));

    // erase monitor
    window.clear();

    // draw enemy
    for(int i = 0; i < enemyNum; i++) {
        window.draw(enemyLst[i].getCircle());
    }

    // draw player
    window.draw(player.getCircle());

    // draw text
    window.draw(tTime);
    window.draw(tEnemy);

    // display monitor
    window.display();
}

return 0;
}

// player 가 화면 밖으로 나가지 못하도록
// enemy 를 10초에 1개씩 늘리기(?)

```

```
// enemy 는 한번 생성되면 없어지지 않고 화면 끝에서 튕겨냄  
//
```

- 진행상황 :

enemy 와 player 객체
enemy 객체 vector 리스트
enemy 수 출력 텍스트
시간(sec) 출력 텍스트
player 객체 방향으로 움직임

- 계획 :

enemy 움직임, 속력
player 와 enemy 충돌시 게임종료
10 초에 enemy 객체 한 개씩 증가

player 모양
player 생명

- 어려운 점 :

enemy 객체의 움직임을 어떻게 구현해야 할까
10 초마다 enemy 객체가 하나씩 추가되는 것을 어떻게 구현해야 할까

2. 느낀점

지금까지 단순한 프로그램만 구현해 보다가 SFML 을 이용해 실제로 객체를 움직이는 것을 눈으로 보며 프로그램을 만들다 보니 흥미롭고 지금까지 배운 내용들을 다양하게 활용할 수 있어서 좋았다.