<객체지향프로그래밍2 1차 오프라인과제>

12171661 윤혁

1.게임 창 생성

sfml로 만든 “Lunar Lander”라는 게임의 프로세스를 설명하겠습니다. main함수부터 따라가면 RendowWindow클래스의 window에 사이즈 600,480크기의 videomode화면을 생성하고 그 화면은“Lunar Lander by 12171661 윤혁”이라는 이름을 가집니다. 그런 다음 window객체의 메소드를 이용해 앱이 모니터의 frequency대신 주어진 프레임비율 60으로 수행시킵니다.

2.게임 화면 구성

게임 화면을 구성하기 위해 texture객체를 생성해 loadfromfile함수로 배경화면의 주소를 가져오고, sprite객체(게임의 구성요소)로 이미지를 포장하여 처리합니다. 그리고 코드 하단에 window.clear()로 화면을 지우고 window.draw로 배경을 그립니다. 게임을 시작하기에 앞서 직접 만든 클래스 생성자로 우주선을 생성해줍니다. 우주선은 x좌표가 300, y좌표, 속도와 연료는 직접 인스턴스로 받습니다. 배경화면과 똑같이 texture객체를 2개 생성하고 각각 우주선, 불꽃에 해당하는 이미지를 sprite객체로 포장하여 처리합니다. 이 또한 window. draw로 그려줍니다. 우주선의 시작위치는 x가300, y는 입력 값으로 설정합니다. 불꽃의 위치는 x+20, y+50으로 설정됩니다. 또한 텍스트를 표현하기위해 폰트도 OpenSans-Bold.ttf로 설정해줍니다. 이는 draw함수에서 폰트 크기는 20, 위치는 (10,100)좌표로 설정하고 window.draw로 글씨를 나타냅니다. 폰트파일이 없을 시에는 “폰트 파일을 오픈할 수 없음!”을 띄웁니다.

3.게임 루프

while(window.isopen())🡪게임 화면이 열려있는동안 계속 게임을 실행합니다. event 객체를 생성하고 (while(window.pollEvent(e))🡪게임이벤트가 감지되면 실행합니다. 만약 키보드에서 위 방향키를 누르면 3이 전달되는데 update함수를 통해 연료는 누를 때마다 3씩 감소하고 속도는 속도-3+0.8로 변합니다. 높이 y도 속도에 비례해 증가합니다. 만약 y가 착륙하면, 즉 450보다 커지면 더 이상 증가하지 않고 450으로 고정됩니다. 우주선의 위치가 변화된 y값을 반영해 재설정됩니다. 불꽃 또한 마찬가지입니다. 이렇게 계속 버튼을 누르면 연료가 감소하기 때문에 연료가 0이되면, 더 이상 감소하지 않고 함수 인수도 0이 됩니다. 사실상 게임 종료입니다. 그리고 화면 왼쪽에 우주선의 높이, 속도, 남은 연료의 양을 지정된 font의 text로표시해줍니다.