

과제5 보고서

18011706 김영훈

벽돌 깨기 게임을 구현하기 위해 설계한 클래스는 크게 2분류로 디스플레이를 위한 클래스들은 JFrame을 상속받은 HW5_18011706, JPanel을 상속받은 TitlePanel, PlayingPanel, EndingPanel 들이고, 내부에서 도형을 표현하기 위한 클래스들은 추상 클래스 HW_GameObject, 이를 상속받은 MySphere, MyRectBlock, MyDynamicRect, TargetRect들이 있습니다.

프레임에서는 각각의 패널들을 CardLayout을 통해 관리하고 CardLayout의 show메소드를 통해 패널들을 바꾸도록 구현했습니다. TitlePanel과 EndingPanel은 한 객체로 계속 사용하였고, PlayingPanel은 초기화하여 사용하기보다는 객체를 그때 그때 새로 생성하는게 게임 세션의 의미에 부합한다고 생각하여 프레임에 startSession 메소드를 구현하여 TitlePanel이 호출하도록 하였습니다. 공이 모두 죽어서 PlayingPanel이 끝나는 경우 showEnding 메소드와 프레임의 renewHighScore메소드를 사용해 EndingPanel을 나타내고 하이스코어를 갱신해주었습니다.

도형 클래스들은 과제 힌트영상에서 만든 도형들 외에도 HW_GameObject를 상속받아 게임 내부의 라켓에 해당하는 MyDynamicRect와 깨지는 블록에 해당하는 TargetRect들을 추가로 구현하였습니다. 라켓의 경우 충돌 처리를 양쪽 벽 2개만 처리하면 되기에 인자로 들어오는 객체를 MyRectBlock으로 다운캐스팅하여 포지션을 비교해 해결했습니다. TargetRect 클래스는 생성자에서 20퍼센트의 확률로 색깔을 노랑으로 설정하도록 블록의 색은 구현하였고, PlayingPanel의 collisionProcess에서 반복문을 사용하여 LinkedList를 인덱스를 통해 제어하면서 공이 3개로 늘어나도록 구현하였습니다. MySphere 클래스를 vx의 각도와 vy의 각도의 랜덤 값을 따로 주었고, vx, vy 벡터의 크기를 1로 만들고 speed라는 스칼라 값을 통해 속도를 조절할 수 있도록 수정하였습니다.

벽돌 깨기 게임을 구현하면서 제일 어려운 부분인 충돌처리를 교수님이 강의 영상을 통해 알려주셔서 그나마 쉽게 과제를 수행할 수 있었습니다. 나머지 어려운 부분은 크게 2가지로 나타낼 Panel 객체를 바꾸는 일과 게임의 내용물을 구현하는 일이 있었습니다. 이 2가지 문제에서 제일 어려웠던 점들은 각각 Panel 객체를 바꾸었을 때 바뀐 Panel 객체는 포커스를 가지지 못하는 점과, 공이 노란 블록과 충돌했을 시에 공을 3개로 늘리는 점이었습니다. Panel 객체가 포커스를 가지지 못하는 점은 frame에 띄우고 싶은 패널을 띄우는 show메소드와 창에 떠있는 객체에 포커스를 주는 requestFocusInWidnow메소드를 감싼 메소드인 startSession과 returnTitle 메소드를 구현하여 해결하였고, 공을 3개로 늘리는 기능은 for-each문이나 iterator로는 반복문을 돌면서 자료의 추가가 불가능했기에 for문을 사용하여 직접 index를 제어하는 방식으로 구현하여 해결하였습니다.

다른 과목의 과제들은 주어진 문제에 정확한 답이 있어 공식을 마냥 사용하기만 하면 됐는데, 이번 과제를 통해 프로젝트를 완성하기 위해 배우지 않은 기능을 검색해 알아보기도 하고, 디버

강을 해가면서 구현 되어가는 프로그램을 보면서 성취감을 느끼면서, 군 전역 후 첫 학기에 제가 컴퓨터공학을 배우는 학생인 점을 다시금 알 수 있었습니다.