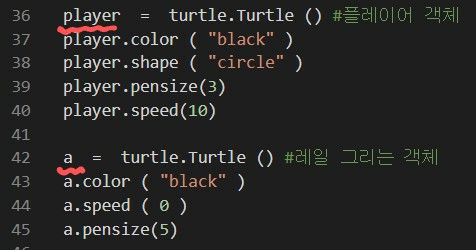
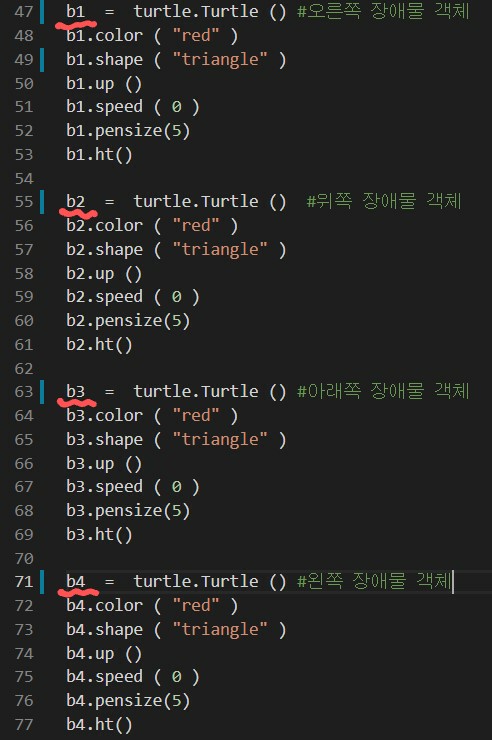


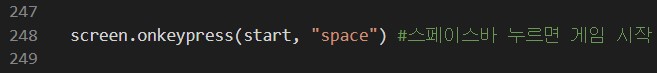
먼저 pygame 모듈과 turtle 모듈을 import합니다. pygame 모듈은 파이썬으로 작성 가능한 게임 등의 멀티미디어 표현을 위한 라이브러리인데요, 게임 개발 도구이지만 이미지 프로세스 또는 조이스틱 입력, 음악 재생 등의 기능만 떼어다 쓸 수도 있다. 저희 프로젝트에서는 음악을 불러오는데 사용했습니다. 그 다음으로 turtle모듈은 커서(거북이)가 이동하며 그림을 그릴 수 있게 해주는 모듈입니다. 리듬게임이다보니 플레이어와 장애물이 이동하는 것을 쉽게 표현하기 위해 사용한 모듈입니다.



turtle 모듈을 이용해 플레이어 객체와 이것이 지나갈 레일을 그리는 객체를 생성합니다.



장애물 객체 4개를 생성합니다.

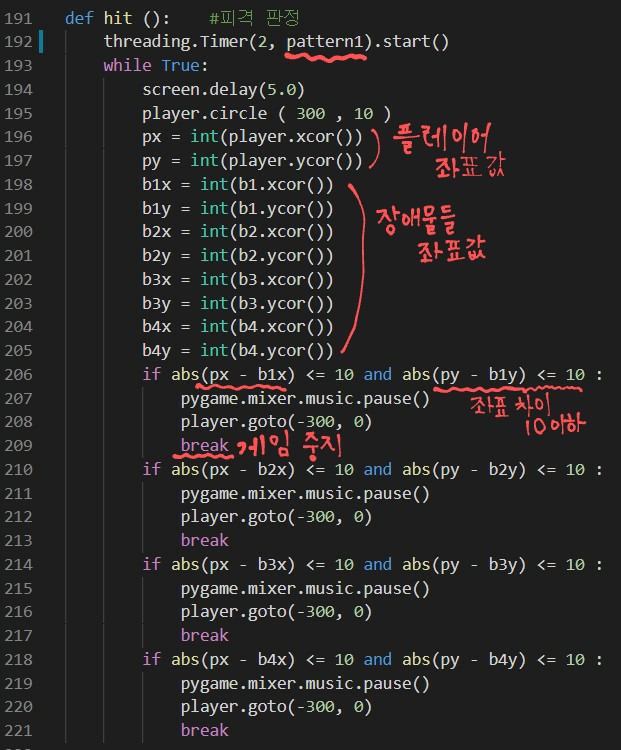


사전 객체 생성은 완료되었고 이제 onkeypress를 사용해 스페이스바를 누르면 start함수를 실행하도록 합니다.

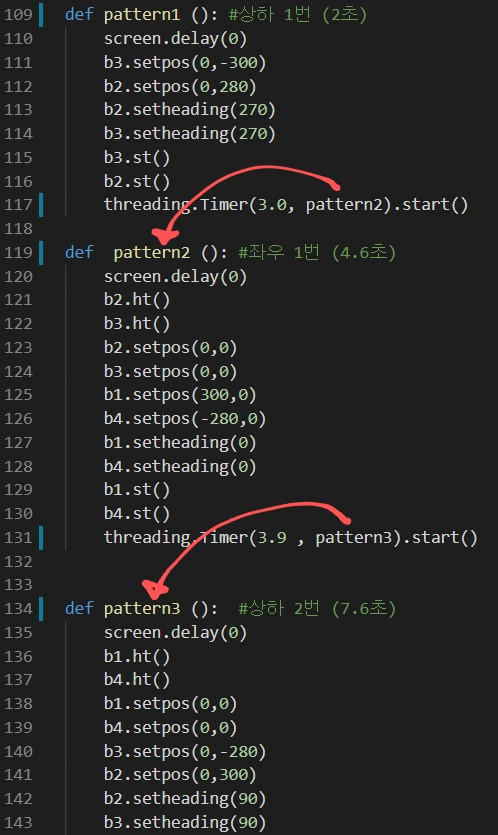
텍스트이(가) 표시된 사진

자동 생성된 설명

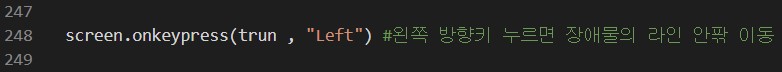
start 함수를 실행하면 pygame을 이용해 음악이 나오게 되고 hit함수를 실행하게 됩니다.



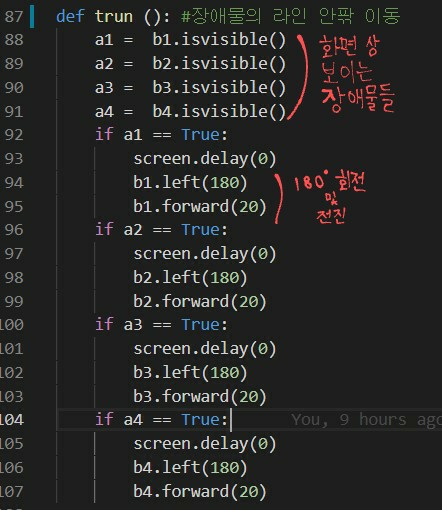
hit함수가 실행되면 pattern1함수를 실행하게 되고, 이 함수를 실행시키는 역할만 하는 것이 아니라 while문을 이용해 게임 전반에 걸쳐 지속적으로 플레이어와 장애물들의 좌표값을 뽑아내 그 둘의 x좌표 혹은 y좌표의 차이가 각각 10보다 작아지면 게임을 중지합니다.



hit함수에 의해 pattern1함수가 실행되는데, 여기서 말하는 패턴이란 장애물들의 위치와 그것들이 나타나는 시점을 말합니다. 장애물들을 레일 위의 특정 위치에 나타냈다가 그것들을 숨기는 식으로 패턴을 구성했습니다. pattern1 함수가 실행되면 장애물들의 첫번째 패턴을 만들게 되고 곧바로 pattern2 함수를 실행시킵니다. 이 pattern2 함수가 실행되면 장애물들의 두번째 패턴을 만들게 되고 또 곧바로 pattern3 함수를 실행시킵니다. 이렇게 패턴 함수들을 연쇄적으로 연결했습니다. 그리고 씽크 조절을 여기서 하게 되는데 threading.Timer에 그 다음 패턴 함수를 몇 초 뒤에 실행시킬지를 입력해서 맞추게 됩니다.



사전 객체 생성과 패턴 형성이 완료되었습니다. 이제 장애물이 라인 안팎을 이동해서 플레이어가 살아남을 수 있도록 합니다. onkeypress를 사용해 왼쪽 방향키를 누르면 trun 함수를 실행하도록 합니다.



turn함수가 실행되면 화면 상에 보이는 장애물들의 방향을 180도 회전시키고 동시에 앞쪽으로 20만큼 이동시킵니다. 이로써 장애물들이 라인을 완벽하게 안팎으로 갈아타도록 합니다.