

Project: 총알 피하기 UPGRADE

소프트웨어적 사고(1143704-01), 20233020 권민기

(미션1~11) 미션별로 구현한 방법에 대한 간략 설명.

미션	설명	구현
미션1. [5점] 비행기가 총알에 맞았을 때 효과음이 발생하도록 한다.	<p>코드: <code>Pygame.mixer.Channel(0).play(pygame.mixer.Sound(""))</code></p> <p>방법: 충돌 시 사용했던 함수인 <code>collision(a,b)</code>에 넣어, 실행되게끔 했다. 음원은 <code>sellbuyMusic</code>이라는 무료효과음 사이트에서 찾아 사용했다.</p> <p>기타: <code>Pygame.mixer.music.load("")</code>, <code>...play("")</code>를 사용했을 경우, 이미 출력되고 있었던 배경음악이 중단되며, 효과음 이후, 아무런 소리가 나지 않았음을 확인해, 동시에 실행될 수 있게끔, 서로 다른 <code>Channel(0,1)</code>를 설정해, 중단되지 않게끔 구현했다.</p>	○
미션2. [5점] 비행기가 총알에 맞았을 때 터지는 그림 효과가 나타나도록 한다.	<p>코드: <code>def fire(self): self.current_image = self.image[1], player.fire()</code></p> <p>방법: 미션1과 같은 방식으로, 함수 <code>collision(a,b)</code>에 넣어, 같이 실행될 수 있게 만들었다. 이때, 터지는 효과를 두개의 사진을 사용하여, 총알에 부딪히면, 마치 '반짝'하는 모습을 구현하고자 했다.</p> <p>기타: <code>fire(self)</code>라는 함수를 따로 만들어, 관리하기 쉽게 만들었다.</p>	○
미션3. [5점] 배경그림이 비행기의 움직임에 반응하여 같이 움직이도록 한다.	<p>코드: <code>background.update(dt,screen), screen.blit(background.image1, (background.pos1[0], background.pos1[1]))</code></p> <p>방법: <code>player.py</code>처럼 <code>background.py</code> 파일을 따로 만들어, 배경그림을 따로 관리하게 두었으며, <code>player.py</code>에서 우주선이 움직이게 구현했던 방법과 똑같이, 사용자가 원하는 방향키를 누르게 되면, 해당 방향으로 배경그림이 움직인다.</p>	○

	<p>기타: 화면에 보이는 부분을 둘러싸 8개의 추가 복사본을 만들고, 배경그림이 느리게 움직이게 속도를 설정해, 설정해둔 그림의 범위를 벗어나 생기는 오류를 방지했다. (이때, <code>pygame.transform.flip(bg_image1, True, False)</code>으로 그림대칭을 통해, 그림간의 경계가 어색하지 않게 설정했다.)</p>	
<p>미션4. [15점] 비행기가 총알과 여러 번 충돌해야 게임이 종료 되도록 한다.</p>	<p>코드: <code>Health = 70, if collision(a,b): Health -= 2, gameover = True</code></p> <p>방법: Health라는 체력을 담당하는 변수를 만들어, 총알에 충돌하더라도 게임이 바로 종료되지 않게 설정했다. collision의 조건에 따라, Health의 숫자가 감소하며, 여러 번의 충돌 하에 Health의 저장 값이 0이 되면, 게임이 종료된다.</p> <p>기타: While문 안에서 실행되는 코드라, <code>Health -= 2</code>가 반복되어 Health 변수 값을 크게 잡아(70), 한 번에 게임 종료되는 것을 방지했다.</p>	Δ
<p>미션5. [5점] 무적시간동안 비행기가 반짝거리도록 한다.</p>	<p>코드: 구현하지 못했습니다.</p> <p>방법: 미션2와 같이, 살짝 어두운 우주선 그림을 추가로 설정해, 마찬가지로 깜박깜박하게 만들고 싶었지만, 미션2처럼 비행기 그림을 한 번 대체하고 나면, 원래대로 그림을 되돌리는 방법을 찾지 못했습니다.</p> <p>기타: <code>pygame.image.load("images/player.png"), pygame.image.load("images/dark_player.png")</code>이라 연이은 적은 후, 다시 <code>pygame.image.load("images/player.png")</code>를 적어 실행하게 되면, 그대로 원래 우주선의 모양이 유지돼 보여, 대체 방법을 찾지 못했습니다.</p>	X
<p>미션6. [10점] 남은 생명력을 막대기(5점)와 숫자(5점)로 표시한다.</p>	<p>코드: <code>txt1 = "Time: {:.1f}, Bullets: {}, Health: {}".format(time.time()-start_time, len(bullets), Health), draw_text(txt1, 16, (10,10), (255,255,255), screen) ..</code></p> <p>방법: 미션4에서 설정한 변수 Health의 값을 적용했다. 앞서, 만든 <code>draw_text</code> 함수를 이용해, 화면 위에 글씨가 적히게 만들었다.</p>	O

	<p>기타: 막대기는 한글 자음 "ㄹ"을 Health 변수 값만큼 출력해 마치 막대기 '바'처럼 보이게 설정했고, Health값이 위 미션4 실행에 따라 감소하면, 'ㄹ'의 개수도 줄어, 우주선 체력이 줄어드는 것처럼 보이게 만들었다. 추가로, 우주선 바로 위에 체력 바를 띄어 우주선과 같이 움직이게 만들었다.</p>	
<p>미션7. [10점] 총알을 여러 종류로 만들고, 종류별로 크기와 색깔을 다르게 표현한다.</p>	<p>코드: class Bullet1: ... class Bullet2: ... class Bullet3:</p> <p>방법: 앞서 구현한 Bullet 클래스의 수를 늘려, 세부사항(self.radius, self.color) 값을 수정해, 여러 종류의 총알이 출력되게끔 설정했다. 당연히, main.py에 있는 list bullets에 모두 append할 수 있게 추가했다.</p>	O
<p>미션8. [5점] 총알의 종류마다 플레이어와 충돌했을 때 차감되는 생명력을 다르게 한다.</p>	<p>코드: 구현하지 못했습니다.</p>	X
<p>미션9. [10점] 사용자의 가장 오래 버틴 생존 시간을 최대 10개까지 파일에 기록한다.</p>	<p>코드: 구현하지 못했습니다.</p>	X
<p>미션10. [5점] 게임오버시에, 파일에 기록된 생존시간을 화면에 출력한다.</p>	<p>코드: start_time = time.time(), txt1 = "Time: {:.1f}, Bullets: {}, Health: {}".format(time.time()-start_time, len(bullets), Health)</p> <p>방법: 모듈 time를 적용해 시간을 측정할 수 있게 만들었다. start_time 변수를 코드 맨 위에 작성해 게임 실행시의 시각을 값으로 담았고, 미션6에서 했던 것처럼 Health와 같이 좌측 상단에 생존시간을 표시했고, 종료 후 뜨는 Game Over와 함께 생존시간을 표시했다.</p> <p>기타: 소수점 단위를 최소화하기 위해, {:.1f}를 뒤에 적어줌으로써 소수 첫째자리까지 나타나게 설정했다.</p>	O
<p>미션11. [5점] 게임오버시에, 오래 버틴 10개의 기록 안에 현재의 기록이 있을 경우 강조하여 표시한다.</p>	<p>코드: 구현하지 못했습니다.</p>	X