2018년 2학기 파이썬프로그래밍응용

프로젝트 최종 보고서

**벽돌 부수기 게임**

작성자: 4조 김영식 (학번:20175119)

1. **서론**
   1. 배경 및 동기

처음 만들어보는 게임이라서 간단하게 만들 수 있는 게임이 무엇이 있을까 생각해보았습니다. 그래서 처음에 나온 것이 리듬게임이 였습니다. 하지만 도트 관련 처리가 상당히 복잡해지고 음악에 따라 각각 만들어야 되는 도트가 달라 직접 노래마다 만들어야 된다고 생각하자 기술적, 시간적 문제가 발생할 것으로 염두가 되어 결국 차선택으로 벽돌 부수기 게임이라는 것을 만들게 되었습니다.

1.2. 코드분석

import pygame, random //pygame과 random이라는 모듈을 가져온다

from pygame.locals import \*//모듈을 가져온다

BLACK = [0, 0, 0] //각각 rgb를 나타낸다

BLUE = [0, 0, 255]

RED = [255, 0, 0]

YELLOW = [255, 255, 0]

GRAY = [192, 192, 192]

WHITE = [255, 255, 255]

WINDOW\_SIZE = [500, 400] //화면의 해상도를 설정할 때 사용

def intro():

pygame.mixer.music.load("Picnic.mp3")//노래를 불러온다

pygame.mixer.music.play(-1)//노래가 시작되고 끝나도 계속 반복

while True:

screen.fill(WHITE)//화면을 하얀색으로 처리

text1 = font.render("게임을 시작하려면 <스페이스> 키를 누르세요.", True, BLUE)

//각각의 매개변수의 의미는 텍스트, 안티앨리어징 여부, 글자 색깔이다

text2 = font.render("그냥 종료하려면 <ESC> 키를 누르세요.", True, RED)

screen.blit(text1, (60, 100))//텍스트를 정해진 위치에 놓는다

screen.blit(text2, (60, 150))

for event in pygame.event.get()://이벤트가 발생하면 받아온다

if event.type == pygame.KEYDOWN://이벤트의 타입이 키보드가 눌린것일 경우

if event.key == K\_SPACE://그 경우가 스페이스일경우

ready()//게임 진행 매서드 호출

elif event.key == K\_ESCAPE://esc일경우

pygame.quit()//파이게임 종료

exit(0)//프로그램 종료

pygame.display.flip()//화면을 출력해준다

def ready():

global player\_pos, ball\_pos, tile\_shape, tile\_color, ball\_move

//함수 밖에서도 사용가능하도록 선언

player\_pos = [150, 250, 100, 20]//플레이어(막대) 위치와 크기설정

ball\_pos = [player\_pos[0] + 50, player\_pos[1] - 10]//공의 위치 설정

tile\_shape = []//타일 생성

tile\_color = []

ball\_move = [0, 0]

for x in [20, 110, 200, 290,380]://벽돌의 x좌표

for y in [50, 80, 110, 140,170]://벽돌의 y좌표

tile\_shape.append([x, y, 80, 20])//타일 생성

tile\_color.append([random.randint(100, 255), random.randint(100, 255), random.randint(100, 255)])

//타일 색깔은 랜덤으로 설정

pygame.mixer.music.stop()//원래 나오던 노래 종료

pygame.mixer.music.load("Picnic.mp3")//노래를 불러온다

pygame.mixer.music.play(-1)//노래 시작 무한반복

play()//매서드 호출

def play():

global player\_pos, ball\_pos, tile\_shape, tile\_color, ball\_move, start\_time, play\_time

//함수 밖에서도 사용가능하도록 선언

moving = False//공이 움직이고있는지 체크해준다 처음엔 false

while True://무한반복

screen.fill(BLACK)//스크린을 검은색으로 한다

tile = []//리스트 생성

for i in range(len(tile\_shape))://벽돌 갯수만큼 돈다

tile.append(pygame.draw.rect(screen, tile\_color[i], tile\_shape[i], 0))

//스크린에다가 사각형 색깔과 좌표 그리고 색칠되도록 설정

if moving://움직이고있을경우

play\_time = round((pygame.time.get\_ticks() - start\_time) / 1000, 1)

//지금까지 진행된 시간을 소숫점 한자리만 나타내어 계산

timer = font.render(str(play\_time) + "초", True, WHITE)

//시간을 문자열로 저장

screen.blit(timer, (300, 20))

//스크린을 이용하여 전달한 좌표에 문자열 출력

if ball\_pos[0] <= 10 or ball\_pos[0] >= 460: //x좌표 오른쪽 왼쪽

ball\_move[0] = ball\_move[0] \* (-1)//공의 튕기는 각도를 반대방향으로 한다

if ball\_pos[1] <= 10://y좌표가 10이하 일경우

ball\_move[1] = ball\_move[1] \* (-1)//공의 튕기는 각도를 반대방향으로 한다

elif ball\_pos[1] >= 310://y좌표가 310보다 커질경우 막대아래로 갔다고 판단하여

moving = False//공이 움직이지 않는다고 처리하고

end()//매서드 호출

ball\_pos[0] = ball\_pos[0] + ball\_move[0]//각각의 처리에따라서 공의 위치를 변환한다

ball\_pos[1] = ball\_pos[1] + ball\_move[1]//x,y

//아직 공이 움직이고 있지 않을경우

ball = pygame.draw.circle(screen, YELLOW, ball\_pos, 10, 0)//노란색공을 만든다

player = pygame.draw.rect(screen, GRAY, player\_pos, 0)//플레이어가 움직일 수 있는 막대를 만든다

if len(tile\_shape) == 0://벽돌의 갯수가 0개가 되면

pygame.display.flip()//화면을 출력해주고

moving = False//공이 움직이지 않는다고 처리하고

end()//매서드 호출

else://갯수가 0이 아닌경우

for i in range(len(tile\_shape))://벽돌 갯수만큼 반복

if ball.colliderect(tile[i])://공과 벽돌이 부딪힐경우

effect.play()//효과음 출력

tile\_shape.pop(i)//타일 갯수 1감소

tile\_color.pop(i)

//벽돌의 가로 부분에 맞았을 경우

if ball\_pos[0] <= tile[i][0] or ball\_pos[0] >= tile[i][0] + tile[i][2]:

ball\_move[0] = ball\_move[0] \* (-1)//반대 방향으로 이동

//벽돌의 세로 부분에서 맞았을경우

if ball\_pos[1] <= tile[i][1] or ball\_pos[1] >= tile[i][1] + tile[i][3]:

ball\_move[1] = ball\_move[1] \* (-1)//반대 방향으로 이동

if player.colliderect(ball)://막대와 공이 부딪혔을경우

ball\_move[0] = int((ball\_pos[0] - (player\_pos[0] + 50)) / 10)//공의 x좌표를 이동

if player.y <= ball\_pos[1]://공의 y좌표가 플레이어의 y좌표보다 클경우

ball\_move[1] = ball\_move[1] \* (-1)//공의 y좌표를 반대로 이동

for event in pygame.event.get()://이벤트 처리

if event.type == pygame.KEYDOWN://이벤트 타입이 키눌린것일경우

if event.key == K\_LEFT://왼쪽이 눌렸을 경우

if player\_pos[0] >= 20://막대의 x좌표가 20이상일경우

player\_pos[0] = player\_pos[0] - 20//막대를 왼쪽으로 20이동

if not moving://공이 움직이고 있지 않을경우

ball\_pos[0] = ball\_pos[0] - 20//공의 x좌표를 20감소 시켜서 막대와 함께 이동하게한다

elif event.key == K\_RIGHT://오른쪽이 눌렸을경우

if player\_pos[0] <= 360://막대의 x좌표가 360이상일경우

player\_pos[0] = player\_pos[0] + 20//막대를 오른쪽으로 20이동

if not moving://공이 움직이고 있지 않을경우

ball\_pos[0] = ball\_pos[0] + 20/공의 x좌표를 20증가 시켜서 막대와 함께 이동하게한다

elif event.key == K\_SPACE and not moving://공이 움직이지 않고 스페이스를 눌렀을 경우

ball\_move[1] = -5//공이 위로 올라가게한다

moving = True//공을 움직이는것으로 변경

start\_time = pygame.time.get\_ticks()//시간 재기 시작

elif event.type == pygame.QUIT://이벤트 타입이 종료일경우

pygame.quit()//파이게임 종료

exit(0)//프로그램 종료

pygame.display.flip()//화면에 출력

clock.tick(30)//프레임 조정

def end():

pygame.mixer.music.stop()//나오고있던 노래 종료

while True://무한반복문

pygame.draw.rect(screen, WHITE, [50, 100, 300, 100], 0)//사각형을 화면에 그린다

if len(tile\_shape) == 0://벽돌의 갯수가 0개일경우

msg = font.render("성공!! 당신의 기록은 %.1f초입니다." % (play\_time), True, BLUE)//성공메세지 저장

else://벽돌이 모두 깨지지 않았을경우

msg = font.render("It's end game...", True, RED)//게임 종료 메세지 저장

screen.blit(msg, (60, 110))//메세지를 정해진 위치에 출력

text1 = font.render("다시 하려면 <스페이스> 키를 누르세요.", True, BLACK)//두개의 매세지를 저장

text2 = font.render("종료하려면 <ESC> 키를 누르세요.", True, BLACK)

screen.blit(text1, (60, 140))//각각의 위치에 출력

screen.blit(text2, (60, 170))

for event in pygame.event.get():이벤트가 발생할경우 처리

if event.type == pygame.KEYDOWN://발생한 이벤트타입이 키가 눌린경우

if event.key == K\_SPACE://스페이스인경우

ready()//게임 다시시작 매서드 호출

elif event.key == K\_ESCAPE://esc인경우

pygame.quit()//파이게임 종료

exit(0)//프로그램 종료

elif event.type == pygame.QUIT://이벤트 타입이 종료일경우

pygame.quit()//파이게임 종료

exit(0)//프로그램 종료

pygame.display.flip()//화면에 출력

clock.tick(15)//프레임 조정

//

pygame.init()//파이게임 라이브러리 초기화

pygame.mixer.init()

screen = pygame.display.set\_mode(WINDOW\_SIZE,FULLSCREEN | DOUBLEBUF)

//화면 해상도를 위에 선언한 크기로 초기화, 더블버퍼와 전체화면 사용

font = pygame.font.Font("koverwatch.ttf", 15)//폰트와 그 사이즈를 지정

clock = pygame.time.Clock()//시간을 재주는 변수

effect = pygame.mixer.Sound("atk.wav")//벽돌이 부서졌을때 나는 소리파일을 저장

intro()//인트로 매서드 호출

1. **개발 내용[[1]](#footnote-1)**
   1. 작품의 기능

이 작품은 말 그대로 벽돌을 부수는 게임입니다. 이 게임에서는 단지 공을 막대로 쳐내어 위에 존재하는 벽돌들을 깨부수는 것인데, 이때 공에 막대가 맞는 부분에 따라서 튕겨 나가는 각도를 설정되어 있습니다. 또 벽에 맞아도 튕기게 설정되어 있으며, 게임 시작시에는 시간을 재기 시작하여 게임이 플레이어의 실수로 인하여, 또는 플레이어가 게임을 클리어하여 종료될 시에는 그때까지 걸린 시간을 출력해줄 예정입니다. 또 게임이 시작되기전 화면과 게임이 종료된 후의 화면을 각각 띄울 것이며, 시작화면에서는 게임의 시작만을, 종료화면에서는 게임을 다시 시작하게 하거나 아니면 종료할 수 있도록 설정이 되어 있습니다.

* 1. 구현 내용

위의 기능을 어떻게 만들었는지 설명한다. 설명할 때는 만든 소프트웨어의 순서를 주요하게 설명하면 된다. 코드가 어떤 원리로 동작하는지를 상세히 적는다.

먼저 블록들에 색깔을 적용하기 위해서 색깔들의 RGB를 각각 설정해 주었고, 해상도 또한 설정했습니다.

다음에는 intro()라는 게임 시작 전 게임 메인 화면에서 처리해야 될 부분을 처리해 주었습니다. 노래를 실행하고, 메시지를 띄워 어느 키를 누르면 실행이 되는지 아니면 종료가 되는지 표시해 주었고, 이벤트 타입을 통해서 받아온 키의 종류에 따라서 각각 함수 처리를 해주었습니다. 예를 들어, 스페이스라는 키가 눌렸다고 하면 ready()라는 함수를 호출하도록 하였고, 그 외의 esc키가 눌리면 파이게임을 종료하고 프로그램도 종료하도록 설정했습니다. 마지막으로 파이게임에 있는 함수인 display.flip()을 이용하여 이 매세지들을 화면에 보이게 했습니다.

다음은 intro()에서 space가 눌렸을 경우 발생하는 ready()함수에 대해 말하자면 게임 시작전에 막대와 공 그리고 벽돌들과 벽돌들의 색깔 그리고 공의 움직임을 나타내는 상태들을 global변수로 나타내어 함수 밖에서도 사용이 가능하도록 설정했습니다. 그리고 적절한 위치에 벽돌들을 만들었고, 색깔을 random모듈을 이용해 가져온 난수들을 이용해 색깔들을 설정했습니다. 그리고 노래를 종료하고 다시 시작하도록 하였고 마지막으로 play()함수를 호출했습니다.

거기서 호출된 play()함수는 이 게임에 가장 중요한 부분인데 공이 움직이는가 그렇지 않은가에 대해서 상태를 나타내기 위하여 moving이라는 변수를 두었습니다. 그리고 벽돌들에 색깔을 칠했습니다. 다음에는 공이 움직이고 있을 경우 시간을 재기 시작하고 그 시간을 문자열로 저장합니다. 그래서 게임 중에 계속 흐르고 있는 시간을 볼 수 있습니다. 그리고 이번에는 공의 처리 부분에서 공의 y좌표를 이용하여 그 상태에 따라 게임종료나 공을 반대방향으로 튕기도록 설정했습니다. 다음은 colliderect를 이용하여 공과 막대 그리고 벽돌과 벽에 충돌 판정을 이용하여 플레이어와 맞았을 경우에는 다시 공이 튕겨 나가도록 했고 벽돌에 맞았을 경우에는 벽돌을 사라지도록 처리했습니다. 또한 이벤트처리를 통해 키보드의 움직이는 부분을 이용하여 막대를 조종하여 공을 튕기게 할 수 있게 하였고, 아직 스페이스를 누르지 않아 공이 출발하지 않았을 경우에는 막대와 공을 같은 속도로 이동시켜 같이 움직이게 하였습니다. 결국 벽돌을 다 깨거나 게임 클리어에 실패할 경우 end()함수를 호출하도록 합니다.

End()함수에서는 노래를 종료하고 벽돌을 모두 깨거나 깨지 못했을 경우에 메시지 출력을 다르게 합니다. 그리고 키를 입력 받아서 그 종류에 따라서 ready()함수를 이용하여 다시 게임을 시작하거나 혹은 종료하게 처리하였습니다.

마지막으로 메인에서는 각각 라이브러리를 초기화하고 폰트를 설정하고 시간을 재주는 변수를 만들고 효과음도 처리했습니다.

* 1. 성능 실험 결과

처음에는 파이게임이라는 것도 처음 써보고, 조원들 모두 처음 써보고 게임이라는 것을 처음 만들어 봐서 꽤 힘들었습니다. 하지만 다들 열심히 게임을 만들기 위해 공부했고, 결국 만들어 냈지만 그 동안의 시행착오가 많이 있었다.

먼저, 벽돌들의 위치를 선정하는 것이 힘들었습니다. 이 좌표공간 안에서 어떻게 좌표를 잡아야 벽돌들이 그리고 공과 막대가 잘 화면에 나타날 것인가에 대해 하나하나 좌표를 바꿔가면서 설정하는 것이 힘들었습니다.

그 다음 힘들었던 부분은 공과 충돌 체크 부분이었습니다. 처음에는 단지 좌표만을 이용해 충돌 처리를 하려고 하다가 보니 공이 끼는 경우도 생기고 게임에 오류가 많이 보이기 시작했습니다. 그래서 다른 사람들은 어떻게 처리를 했나 찾아보니colliderect라는 함수를 이용하는 것을 보았습니다. 그래서 그것을 이용해 충돌 처리를 해보니 공이 끼는 경우도 드물게 나타나게 되었고 게임의 질이 향상된 것을 느꼈습니다.

다음으로는 공의 튕기는 방향 부분입니다. 이 부분은 아직도 제대로 고쳤다는 느낌이 들지 않습니다. 왜냐하면 가끔 공이 튕기는 것을 보면 막대가 오른쪽으로 움직여서 공을 치면 공이 좀더 강하게 오른쪽으로 날라가야 되는데 그런 물리적인 방법까지 계산하여 처리하기에는 시간적, 기술적 문제가 부족하다고 느껴 단지 가만히 있을 때 공이 튕기는 것과 막대가 움직여서 튕기게 하는 것 모두 반대 방향으로만 튕기게 처리하였습니다.

마지막으로 공과 벽돌 들과의 상호작용 부분입니다. 이게 힘들었던 이유가 공과 벽돌의 위쪽 부분, 아래쪽 부분, 오른쪽 부분, 왼쪽 부분에 따라서 튕기는 것을 따로 처리해야 되었기 때문에 힘이 들었습니다. 결국 벽돌의 가로 부분에 맞았을 경우 공의 x좌표를 -로 바꾸어 날라가도록 처리했고 세로 부분에 맞았을 경우에는 공의 y좌표를 -로 바꾸어서 튕겨 나가도록 처리했습니다. 나중에는 좀더 물리적인 법칙들이나 그런 것들을 적용하고 아이템이나 나만의 독특한 벽돌 부수기 게임이 될 수 있도록 코드를 수정, 보완하고 싶습니다.

1. **남은 과제**

처음 만든 게임인 만큼 부족한 부분이 많이 보이고 힘들었지만, 가장 힘들었던 부분은 내가 넣고 싶던 많은 기능들을 넣지 못한 것입니다. 대략 만들려고 하다 삭제한 기능들이 몇 가지 있는데, 먼저 게임의 난이도 부분입니다. 원래는 벽돌이 지금과는 다르게 많이 존재하며 그 벽돌을 일정이상 깨면 깰수록 공의 개수가 증가하도록 설정하고, 그에 따른 난이도의 증가로 플레이어는 목숨을 여러 개(대략3개) 가지고 있도록 하려 했습니다. 그리고 공을 놓칠 때마다 목숨이 하나씩 감소하도록 하려 했지만 아쉽게 시간적 문제로 나중을 기약하게 되었습니다. 그리고 두번째로는 랭킹 시스템을 만들고 싶었습니다. 게임을 클리어 할 시 그때까지 걸린 시간을 파일에 저장하고 예전기록은 불러오는 식으로 순위를 계산하여 플레이어의 순위를 나타내 주고 싶었습니다. 그리고 또 벽돌들 중에도 여러 특징들을 넣어서 공략 순서를 만들고 싶었습니다. 벽돌들에 색깔이 존재하기 때문에 그 색깔을 이용하여 색깔 순서대로 벽돌을 깰 수 있도록 설정하여 빨간색의 벽돌들을 모두 깼을 경우, 주황색의 벽돌들을 깰 수 있는 권한을 가지게 되는 그런 식으로 만들고 싶었습니다. 이럴 경우에는 벽돌들 간의 간격을 넓게 둬서 특정 벽돌들을 노리기 쉽게 하려했습니다.

1. <http://toyfab.tistory.com/592>

   2. <https://nabilera1.blog.me/221079122603>

   -출처- [↑](#footnote-ref-1)