

2023-2 강원중 상상 플레이어

강원중학교

Python 게임 프로그래밍

강원대학교사범대학부설고등학교
장원재

학습 순서

01

횡스크롤 게임 만들기

캐릭터 애니메이션 및 충돌판정

02

게임 프로그래밍을 위한 물리학

Vpython을 활용한 역학 프로그래밍

01

횡스크롤 게임 만들기

캐릭터 애니메이션 및 충돌 판정 만들기

횡스크롤 게임이란? Github.com/carlosquperman

배경이 횡으로 흘러가고 그런 과정에서 플레이어, 적, 보스 캐릭터를 만들고 적을 무찌르는 게임



❖ 횡스크롤 게임

1. 배경은 왼쪽 방향으로 흘러가는 구조
2. 게임을 실행하면 보스 캐릭터가 생성됨
3. 게임을 실행하면 쥘병 캐릭터가 생성됨
4. 키보드로 조작할 수 있는 플레이어를 배경에 만들고
5. 키보드로 미사일이나 빛의 구를 쏘아 적을 무찌르는 게임!

횡스크롤 게임 만들기

게임을 만들기 위한 구성요소를 순서대로 구현하는 과정으로

❖ 횡스크롤 게임

1. 게임의 배경이 되는 게임배경을 생성하고 흘러가게 하기
2. 게임의 조작 대상인 플레이어를 생성하게 하기
3. 게임의 플레이어 캐릭터를 키보드로 움직이게 하기
4. 게임을 방해하는 적 캐릭터를 생성해 움직이게 하기
5. 게임을 방해하는 보스 캐릭터를 생성해 움직이게 하기
6. 플레이어가 조작하는 캐릭터 스프라이트를 동작에 맞춰 바꾸게 하기
7. 미사일을 2가지 생성하여 움직이게 하고 적 캐릭터와 충돌 시 판정 만들기
8. 기타 기능 추가!!!!

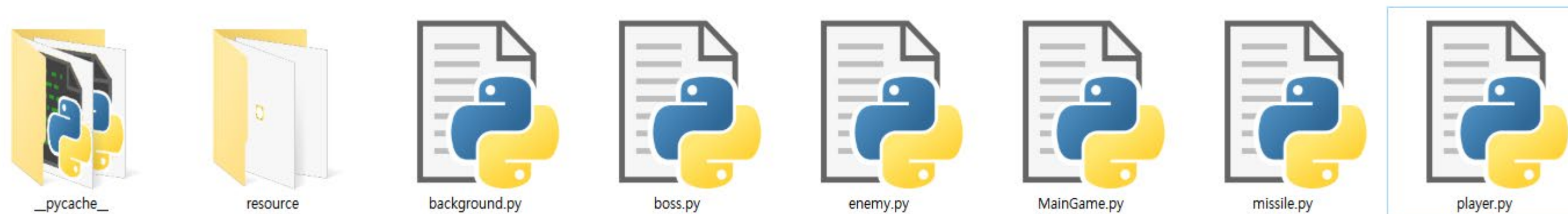
윙스크롤 게임 만들기

게임의 구성 요소를 별도 파일로 저장하기

- ❖ 게임 구성 요소를 클래스로 생성한 후, 생성한 클래스를 별도의 모듈로 저장하기
- 1. 객체지향 구조 프로그래밍(OOP)
- 2. 객체를 중심으로 프로그래밍을 수행하게 하는 것
- 3. 추상화, 상속, 캡슐화, 다형성
- 4. 재사용 및 유지보수가 용이함
- 5. 객체를 중심으로 프로그래밍을 수행하고, 객체를 재정의함으로 다양한 프로그램을 수행함.

윙스크롤 게임 만들기

게임의 구성 요소를 별도 파일로 저장하기



❖ 각각의 모듈 및 리소스

- MainGame.py : 게임의 가장 핵심 파일
- Background, boss, enemy, missile, player.py 프로그래밍에 사용되는 각각의 객체에 대한 정보를 파일로 저장한 것.
- Resource : 게임에 활용되는 리소스 (이미지, 사운드 등을 저장한 폴더)

횡스크롤 게임 만들기

5. 보스 캐릭터 생성하기

❖ In Boss.py

1. 메소드 : 객체지향 클래스에서 특정 기능을 수행하는 코드 집합
2. 변수 : 객체지향 클래스에서 특정한 정보를 저장하는 저장 공간

• Boss 클래스에 있는 메소드

1. `--init--`
2. `draw`
3. `update`
4. `animation`

```
1  import pygame
2  import random
3  import time
4
5  class Boss :
6
7  >   def __init__(self, screen): #보스 정보 초기화...
31
32 >   def draw(self): #보스를 화면에 그리는 기능 ...
37
38 >   def update(self): #보스의 이동에 관련된 요소를 갱신하는 기능 ...
55
56 >   def animation(self) : #보스 스프라이트의 변화를 위한 기능 ...
61
62 >   def bosskill(self) : #보스가 죽었을 시 (사용자가 HP를 0으로 만들었을 시) 화면에 킬 카운트를 출력하고 보스 재생성 ...
74
```


횡스크롤 게임 만들기

5. 보스 캐릭터 생성

❖ In Boss.py에 있는 2.변수 상황

```
def __init__(self, screen): #보스 정보 초기화
    self.screen= screen
    self.boss_img = [ ]
    self.bsx = 800
    self.bsy = 400
    self.std_count = 0

    self.hp_img = pygame.image.load("resource/hp.png") #초록색 = 풀피
    self.bar_img = pygame.image.load("resource/hp_bar.png") #빨간색 뺄피

    self.moveD = True
    self.moveU = False

    self.fhp = 80
    self.hp = 80
    self.isDead = False
    self.killcount = 0
    self.font = pygame.font.SysFont('arial', 50)
```

횱스크롤 게임 만들기

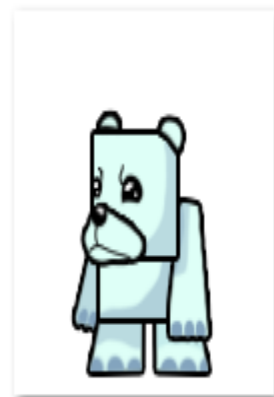
5. 보스 캐릭터 생성

❖ 보스 캐릭터 스프라이트 변화 및 이동 원리

1. 최초 보스 생성 위치는 (800,400)으로 생성 (_ _init _ _)
2. 보스가 죽지 않았다면 보스(움직임은 계속 바뀌게)와 HP 바를 화면에 그려줌 (draw)
3. 보스를 아래로 이동시키다 화면을 넘어가면 위치를 랜덤 X값으로 설정하고, 보스를 위로 움직이게 함. (반대도 함께) (update)
4. 보스 스프라이트의 이미지 변화를 위해 인덱스 값을 바꿔줌 (animation)
5. 보스가 죽었을 시 화면에 킬 카운트를 출력하고 보스를 재생성함(bosskill)

횡스크롤 게임 만들기

5. 보스 캐릭터 생성



play_mombear_run_0.png



play_mombear_run_1.png



play_mombear_run_2.png



play_mombear_run_3.png



play_mombear_run_4.png



play_mombear_run_5.png



play_mombear_run_6.png



횡스크롤 게임 만들기

5. 보스 캐릭터 생성

```
class Boss :  
  
    def __init__(self, screen): #보스 정보 초기화  
        self.screen= screen  
        self.boss_img = [ pygame.image.load("resource/boss_run/play_mombear_run_0.png"),  
                           pygame.image.load("resource/boss_run/play_mombear_run_1.png"),  
                           pygame.image.load("resource/boss_run/play_mombear_run_2.png"),  
                           pygame.image.load("resource/boss_run/play_mombear_run_3.png"),  
                           pygame.image.load("resource/boss_run/play_mombear_run_4.png"),  
                           pygame.image.load("resource/boss_run/play_mombear_run_5.png"),  
                           pygame.image.load("resource/boss_run/play_mombear_run_6.png") ]  
  
        self.bsx = 800  
        self.bsy = 400  
        self.std_count = 0  
  
        self.hp_img = pygame.image.load("resource/hp.png") #초록색 = 풀피  
        self.bar_img = pygame.image.load("resource/hp_bar.png") #빨간색 빨피  
  
        self.moveD = True  
        self.moveU = False  
  
        self.fhp = 80  
        self.hp = 80  
        self.isDead = False
```

- 앞에서 봤던 이미지를 프로그래밍에 활용하기 위해 리스트에 저장
 1. 리스트 : 이름 하나에 N개의 값을 저장하는 자료구조
 2. self.(변수)의 형식 이 인스턴스에서 사용하는 변수(리스트)를 의미함
 3. Pygame.image.load를 활용하여 이미지 위치 정보를 읽어와 각 리스트에 저장

횱스크롤 게임 만들기

5. 보스 캐릭터 생성

```
self.font = pygame.font.SysFont('arial', 30)

def draw(self): #보스를 화면에 그리는 기능
    if not self.isDead:
        self.screen.blit(self.boss_img[self.std_count//60], (self.bsx, self.bsy))
        self.screen.blit(self.bar_img , (self.bsx + 30 , self.bsy), (0,0 ,self.fhp, 20) ) #hp는 보스의 머리위에 그린다. 빨간색 바
        self.screen.blit(self.hp_img , (self.bsx + 30 , self.bsy), (0,0 ,self.hp, 20) ) #hp는 보스의 머리위에 그린다. 초록색 바

def update(self): #보스의 이동에 관련된 요소를 갱신하는 기능

    if self.moveD and self.bsy <= 1000:
        self.bsy += 1

    if self.bsy > 1000:
        self.bsx = random.randrange(0, 1800)
        self.moveD = False
        self.moveU = True

    if self.moveU and self.bsy >= 0:
        self.bsy -= 1

    if self.bsy < 0:
        self.bsx = random.randrange(0, 1800)
        self.moveU = False
        self.moveD = True

def animation(self) : #보스 스프라이트의 변화를 위한 기능
    self.std_count += 1

    if self.std_count >= 419:
        self.std_count =0
```

- 세 개의 메소드 draw, update, animation을 통해서 보스의 움직임과 스프라이트 변화를 프로그래밍함

횡스크롤 게임 만들기

5. 보스 캐릭터 생성



횱스크롤 게임 만들기

6. 키를 누를 때마다 캐릭터 애니메이션이 바뀌게 하기

❖ In player.py

1. 메소드 : 객체지향 클래스에서 특정 기능을 수행하는 코드 집합
2. 변수 : 객체지향 클래스에서 특정한 정보를 저장하는 저장 공간

```
2
3 ~class Player:
4 > def __init__(self, screen): #초기화 ...
41
42 > def draw(self): #화면에 그리기 ...
51
52 > def update(self): #키보드 동작에 따른 업데이트 ...
61
62 > def animation(self): #플레이어 캐릭터 애니메이션 설정 ...
78
```

- Player 클래스에 있는 메소드

1. __init__
2. draw
3. update
4. animation

횡스크롤 게임 만들기

6. 키를 누를 때마다 캐릭터 애니메이션이 바뀌게 하기

❖ In player.py에 있는 2.번수 상황

```
class Player:
    def __init__(self, screen): #초기화
        self.screen = screen
        self.player_img =
        self.playerR_img =
        self.attack_img =
        self.px = 250
        self.py = 300
        self.moveR= False
        self.moveL= False
        self.moveU=False
        self.moveD=False

        self.std_count = 0 #리스트 인덱스 따라가며 애니메이션 보여주기 위한 변수
        self.run_count = 0 #리스트 인덱스 따라가며 애니메이션 보여주기 위한 변수2

        self.attack = False
        self.att_count = 0
        self.speed= 2
```

횡스크롤 게임 만들기

6. 키를 누를 때마다 캐릭터 애니메이션이 바뀌게 하기

❖ 플레이어 캐릭터 스프라이트 변화 원리

1. 플레이어 캐릭터 스프라이트를 표현하는 세 가지 방법

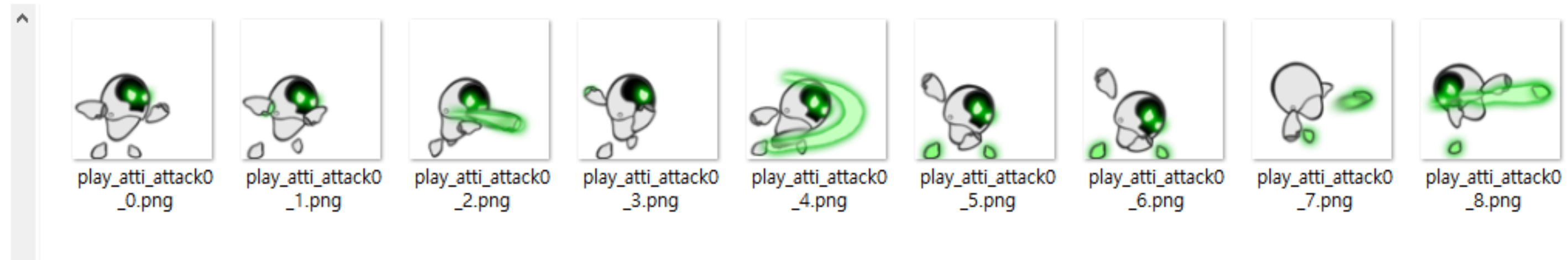
- 1) 사용자가 오른쪽 버튼을 눌렀을 때
- 2) 사용자가 공격버튼을 눌렀을 때
- 3) 사용자가 오른쪽과, 공격버튼 이외의 버튼을 눌렀을 때

2. 각 3가지의 경우에 따른 캐릭터 스프라이트를 형성하는 여러 이미지를 순서대로 바꿔 캐릭터의 움직임을 역동적으로 만들게 한다.

횡스크롤 게임 만들기

6. 키를 누를 때마다 캐릭터 애니메이션이 바뀌게 하기

> 내 PC > 다운로드 > 횡스크롤 게임 자료(강원중_2) > resource > attack



클립보드

구성

새로 만들기

> 내 PC > 다운로드 > 횡스크롤 게임 자료(강원중_2) > resource > run_right

otlib

e

|

d

ia

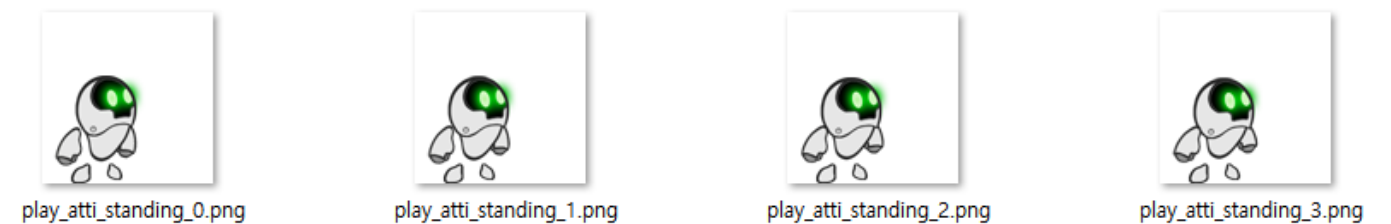
lBox



> 내 PC > 다운로드 > 횡스크롤 게임 자료(강원중_2) > resource > standing

lib

ox



횡스크롤 게임 만들기

6. 키를 누를 때마다 캐릭터 애니메이션이 바뀌게 하기

```
self.screen = screen
self.player_img = [ pygame.image.load("resource/standing/play_atti_standing_0.png"),
                    pygame.image.load("resource/standing/play_atti_standing_1.png"),
                    pygame.image.load("resource/standing/play_atti_standing_2.png"),
                    pygame.image.load("resource/standing/play_atti_standing_3.png")
self.playerR_img = [ pygame.image.load("resource/run_right/play_atti_run_0.png"),
                    pygame.image.load("resource/run_right/play_atti_run_1.png"),
                    pygame.image.load("resource/run_right/play_atti_run_2.png"),
                    pygame.image.load("resource/run_right/play_atti_run_3.png") ]
self.attack_img = [pygame.image.load("resource/attack/play_atti_attack0_0.png"),
                   pygame.image.load("resource/attack/play_atti_attack0_1.png"),
                   pygame.image.load("resource/attack/play_atti_attack0_2.png"),
                   pygame.image.load("resource/attack/play_atti_attack0_3.png"),
                   pygame.image.load("resource/attack/play_atti_attack0_4.png"),
                   pygame.image.load("resource/attack/play_atti_attack0_5.png"),
                   pygame.image.load("resource/attack/play_atti_attack0_6.png"),
                   pygame.image.load("resource/attack/play_atti_attack0_7.png"),
                   pygame.image.load("resource/attack/play_atti_attack0_8.png")
```

- 앞에서 봤던 이미지를 프로그래밍에 활용하기 위해 리스트에 저장
 1. 리스트 : 이름 하나에 N개의 값을 저장하는 자료구조
 2. self.(변수)의 형식 이 인스턴스에서 사용하는 변수(리스트)를 의미함
 3. Pygame.image.load를 활용하여 이미지 위치 정보를 읽어와 각 리스트에 저장

횡스크롤 게임 만들기

6. 키를 누를 때마다 캐릭터 애니메이션이 바뀌게 하기

```
self.std_count = 0 #리스트 인덱스 따라가며 애니메이션 보여주기 위한 변수  
self.run_count = 0 #리스트 인덱스 따라가며 애니메이션 보여주기 위한 변수2  
self.att_count = 0 #리스트 인덱스 따라가며 애니메이션 보여주기 위한 변수3
```

```
def draw(self): #화면에 그리기  
    if self.moveR :  
        self.screen.blit(self.playerR_img[self.run_count // 60] , (self.px, self.py)) #프레임 단위  
  
    elif self.attack:  
        self.screen.blit(self.attack_img[self.att_count // 60] , (self.px, self.py)) #프레임 단위  
  
    else:  
        self.screen.blit(self.player_img[self.std_count // 60] , (self.px, self.py)) #프레임 단위  
  
def animation(self): #플레이어 캐릭터 애니메이션 설정  
  
    self.std_count += 1 #리스트 인덱스를 증가시킨다.  
    self.run_count += 1  
    self.att_count += 1  
  
    if self.std_count >= 239 : #프레임 초기화를 위한 값  
        self.std_count = 0  
  
    if self.run_count >= 239 :  
        self.run_count = 0  
  
    if self.att_count >= 539 :  
        self.att_count = 0
```

- 이미지의 위치를 지정하게
할 3개의 count 변수 생성
후 초기화
- Draw 메소드와 animation
메소드를 활용해 캐릭터를
다양하게 그려줌

횡스크롤 게임 만들기

6. 키를 누를 때마다 캐릭터 애니메이션이 바뀌게 하기



횡스크롤 게임 만들기

7. 미사일을 2가지 생성하여 움직이게 하고 적 캐릭터와 충돌 시 판정 만들기

❖ In Missile.py

- 메소드 : `init(self,screen,boss,enemy)`, `draw`, `update`

```
1 import pygame
2
3 class Missile:
4     def __init__(self, screen, boss, enemy): #미사일을 생성하게 하는 기능 (매개변수 4개 주목) ...
17
18     def draw(self): #미사일 2개를 화면에 그려주는 기능 ...
26
27     def update(self): #미사일이 다른 인스턴스와 상호작용하게 하는 기능 ...
75
```


횡스크롤 게임 만들기

7. 미사일을 2가지 생성하여 움직이게 하고 적 캐릭터와 충돌 시 판정 만들기

❖ In Missile.py에 있는 2.변수 상황

```
def __init__(self, screen, boss, enemy): #미사일을 생성하게 하는 기능 (매개변수 4개 주목)  
    self.screen = screen  
  
    self.boss = boss  
    self.enemy = enemy  
  
    self.missile_img = pygame.image.load('resource/bullet.png')  
    self.missile_img2 = pygame.image.load('resource/bullet2.png')  
  
    self.missile_pos = []  
    self.missile_posU = []  
  
    self.isShot = False
```

윙스크롤 게임 만들기

7. 미사일을 2가지 생성하여 움직이게 하고 적 캐릭터와 충돌 시 판정 만들기

❖ 미사일 작동 원리

1. 사용자가 키보드(a, s키)를 누를 때마다 미사일 객체에 있는 리스트(self.missile_pos, self.missile_posU)에 집어넣어주고 (사용자의 위치 함께 전달)
2. draw() 메소드를 통해서 화면에 미사일을 그려줌
3. update() 메소드를 통해서 미사일 종류에 따라서 미사일의 x축 값, 또는 y축 값을 바꿔주고 화면 범위를 벗어나면 해당 미사일을 삭제함
4. update() 메소드를 통해서 미사일과 적(보스, 일반 적)과의 충돌 여부를 판정하고, 일반적의 경우에는 적을 바로 없애주고, 보스의 경우에는 보스의 hp를 10 감소시킴

횡스크롤 게임 만들기

7. 미사일을 2가지 생성하여 움직이게 하고 적 캐릭터와 충돌 시 판정 만들기

충돌 체크

```
import pygame
```

```
class Missile:
```

```
def __init__(self, screen, bo):
```

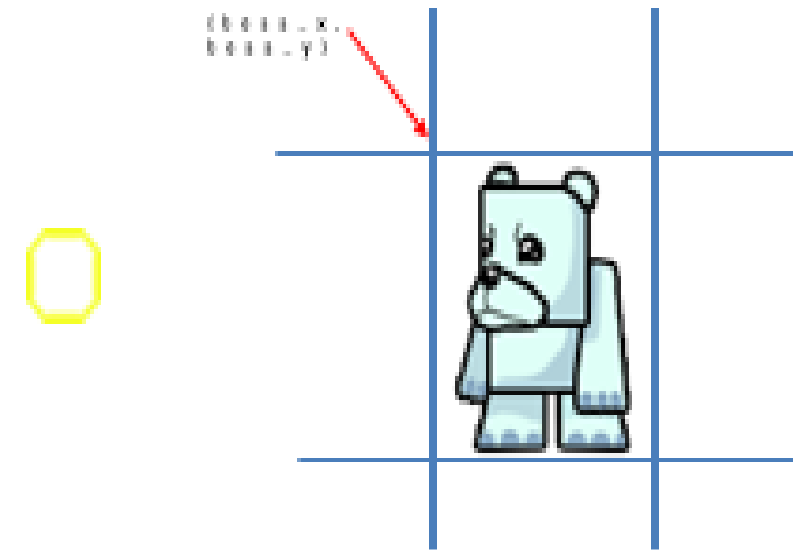
```
    self.screen = screen
```

```
    self.missile_img = pygame.image.load("bullet.png")
```

```
    self.missile = []
```

```
    self.isShot = False
```

```
    self.boss = bo
```



횡스크롤 게임 만들기

7. 미사일을 2가지 생성하여 움직이게 하고 적 캐릭터와 충돌 시 판정 만들기

```
if self.boss.bsx <= self.mxy[0] < self.boss.bsx+130:  
    if self.boss.bsy <= self.mxy[1] <= self.boss.bsy+130:  
        self.isShot = True  
        del self.missile_pos[self.i]  
  
if self.enemy.ex <= self.mxy[0] < self.enemy.ex + 90:  
    if self.enemy.ey <= self.mxy[1] <= self.enemy.ey+90:  
        self.isShot = True  
        del self.missile_pos[self.i]
```

횡스크롤 게임 만들기

7. 미사일을 2가지 생성하여 움직이게 하고 적 캐릭터와 충돌 시 판정 만들기



윙스크롤 게임 만들기

7. 미사일을 2가지 생성하여 움직이게 하고 적 캐릭터와 충돌 시 판정 만들기

❖ 미사일 판정 점검의 문제점

1. 일반적인 경우에는 미사일을 맞으면 바로 사라지지만
2. 일반적이 미사일을 맞을 때 보스 캐릭터의 HP가 같이 깎이는 문제점
3. 이 문제를 어떻게 해결할 수 있을까?

횡스크롤 게임 만들기

7. 미사일을 2가지 생성하여 움직이게 하고 적 캐릭터와 충돌 시 판정 만들기

❖ 이 프로그램을 어떻게 개선할 수 있을까?

1. 점수 기능 추가
2. 스테이지 기능 추가
3. 음악 집어넣어서 효과성 만들기
4. 배경 이미지를 하나로 고정하는 것이 아니라 좀 더 다양하게 만들기
5. 기타 등등

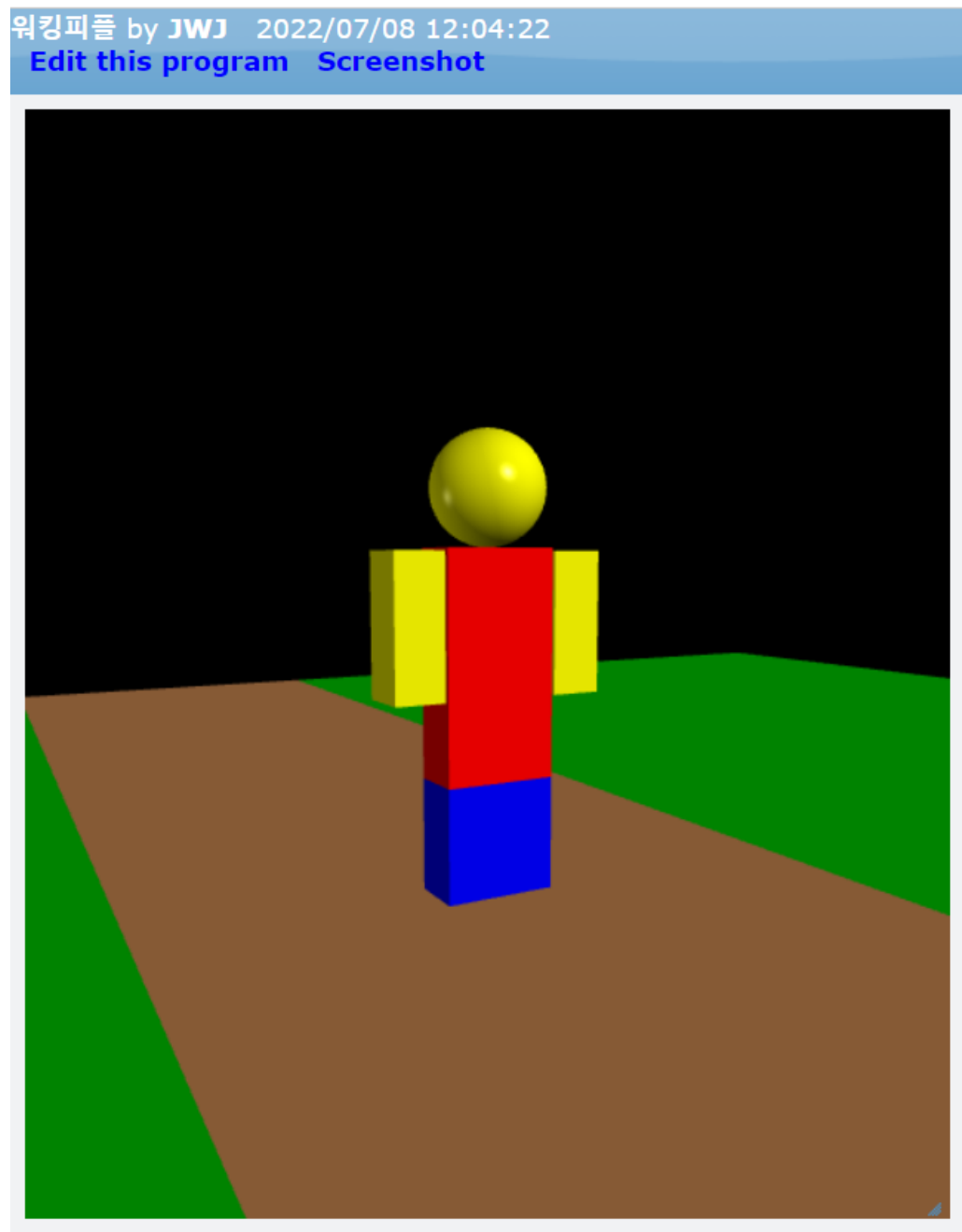
02

게임 프로그래밍을 위한 물리학

Vpython을 활용한 역학 프로그래밍

Vpython 이란?

Python 기반 그래픽 라이브러리



❖ VPython

1. Python에서 작동하는 3차원 그래픽 관련 라이브러리
2. Python 프로그래밍을 통해 3차원 공간에 물체를 만들고 움직임을 부여하며, 키보드-마우스 상호작용을 구현할 수 있음

Vpython 이란?

게임 프로그래밍에 왜 물리학적 지식이 필요할까?

❖ 게임을 작동하게 하는 원리는 무엇일까?

1. 게임에는 규칙이 존재함.
2. 게임 속 세계는 현실 세계와 유사하지만 다른 세계임
3. 그렇지만 현실의 요소를 얼마나 실감 있게 구현하느냐가 게임에 있어서 중요한 요소
4. 게임을 접하다 보면 많이 듣는 단어 물리 엔진
5. 실감 있는 게임 요소를 만드려면 물리적 요소를 배우고 다루는 경험이 필요하다!

Vpython 이란?

게임 프로그래밍에 왜 물리학적 지식이 필요할까?

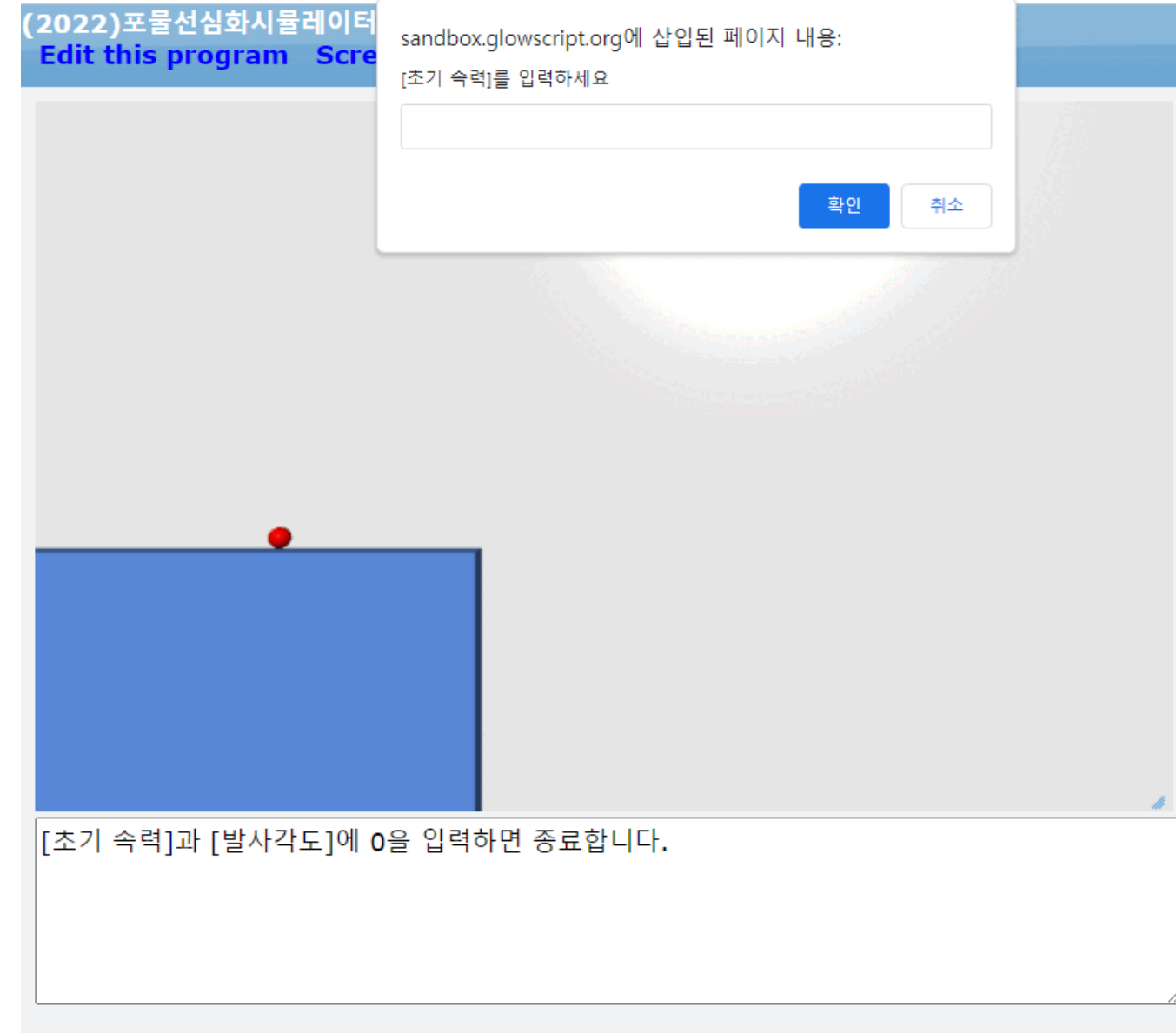
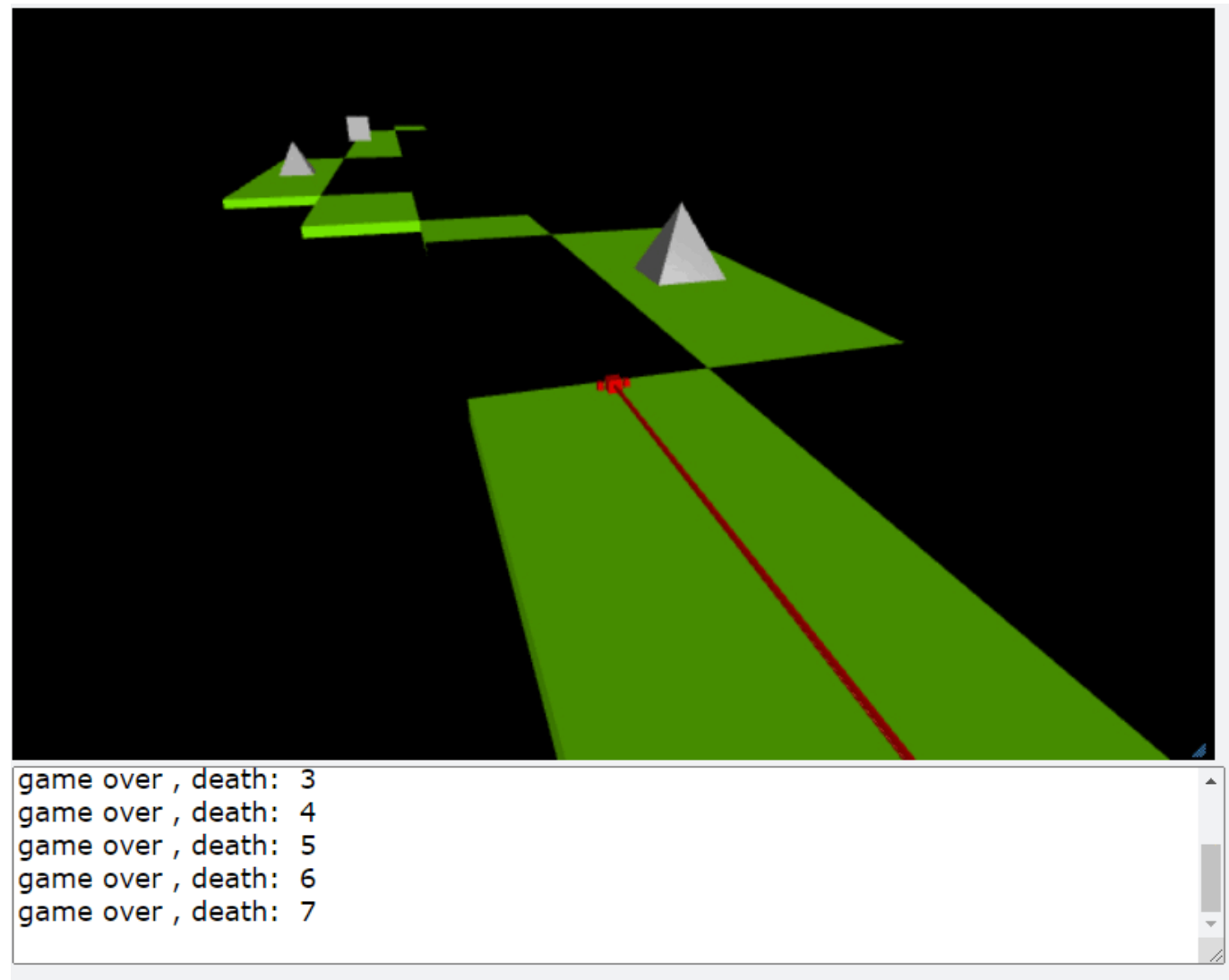
❖ 게임 구현 시 알면 좋은 물리학적 내용?

1. 등속운동 (속도가 동일한 운동)
2. 가속도운동 (속도 변화가 있는 운동)
3. 포물선 운동 (각도와 중력가속도에 따른 운동)
4. 회전운동 (회전축과 회전각도에 따른 운동)

Vpython 이란?

게임 프로그래밍에 왜 물리학적 지식이 필요할까?

❖ 이 내용 등을 구체적 코드를 통해 알아봅시다!



2023 -2 강원중학교

Python 게임 만들기

감사합니다