

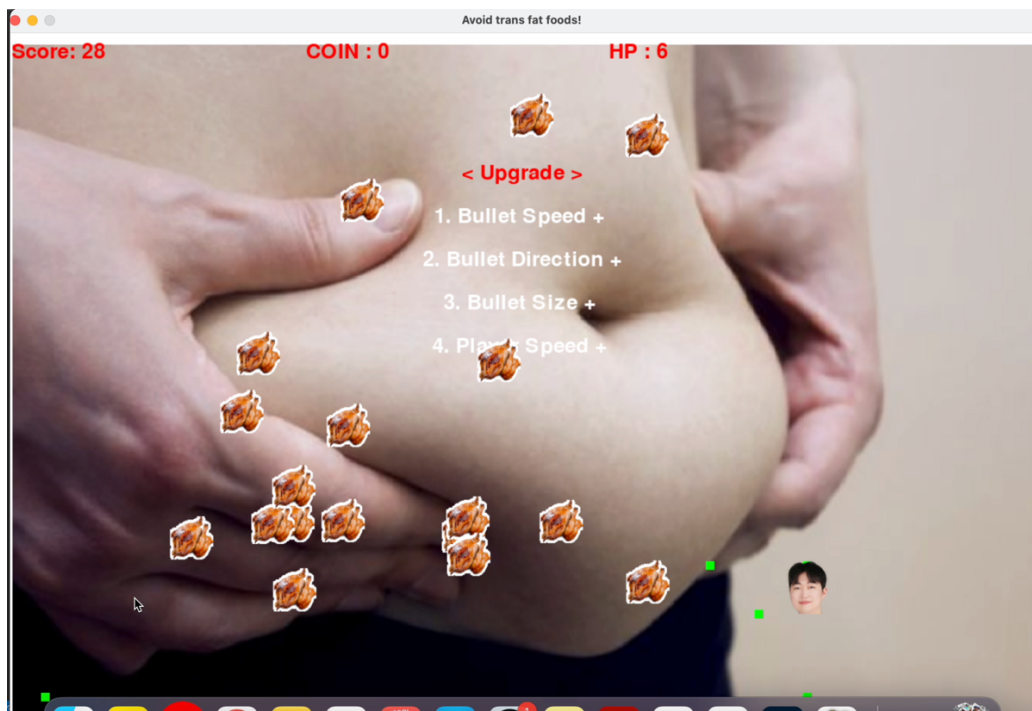
게임프로그래밍입문 00분반

Project#3 Design & Develop Your Own Game

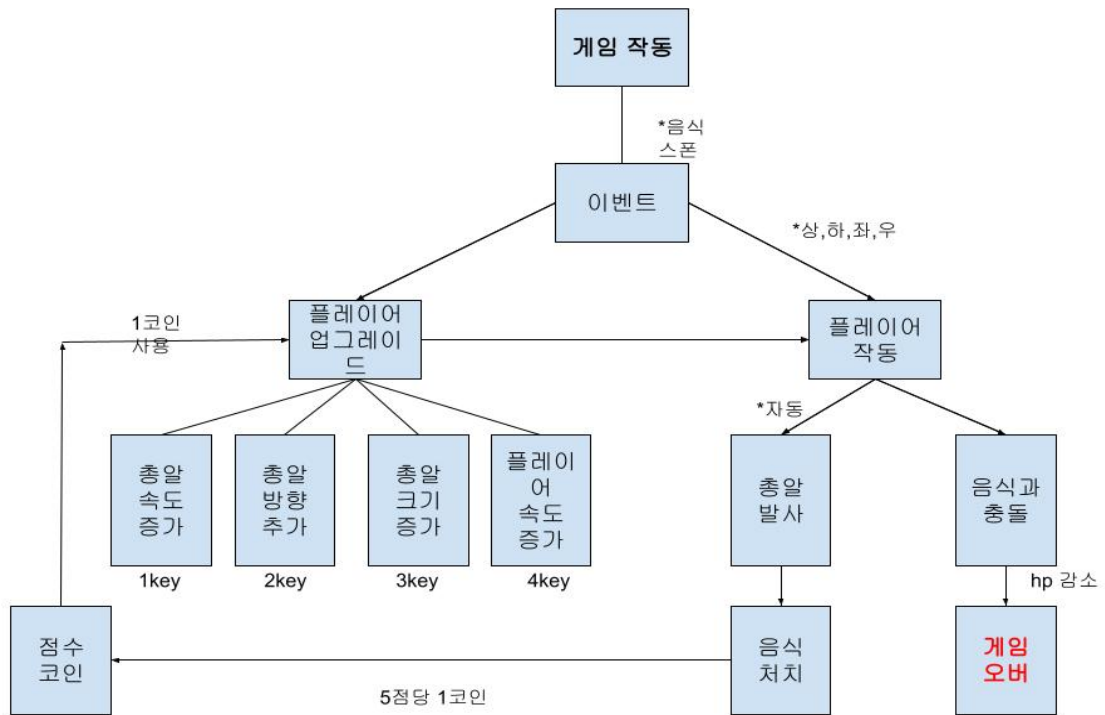
2021105644 컴퓨터공학과 조영호

게임설명

- 1) 게임 종류 : 슈팅게임(missile)
- 2) 게임 제목 : 트랜스지방 푸드 피하기.
- 3) 게임 설명 : 플레이어에게 끊임없이 달려드는 살찌는 음식들을 처치하며 피하는것입니다.
- 4) 게임 진행 :
 1. 음식들이 랜덤으로 플레이어에게 달려온다.
 2. 플레이어를 상,하,좌,우로 움직이며 총알을 쏘서 처치한다.
 3. 처치한 음식들을 스코어에 반영 / 5점당 코인1개.
 4. 코인으로 플레이어의 무기를 업그레이드 한다. (총알 속도, 총발 방향, 총알 크기, 플레이어 속도)
 5. 음식과 닿아서 체력이 0이 되면 게임오버
- 5) 게임 화면

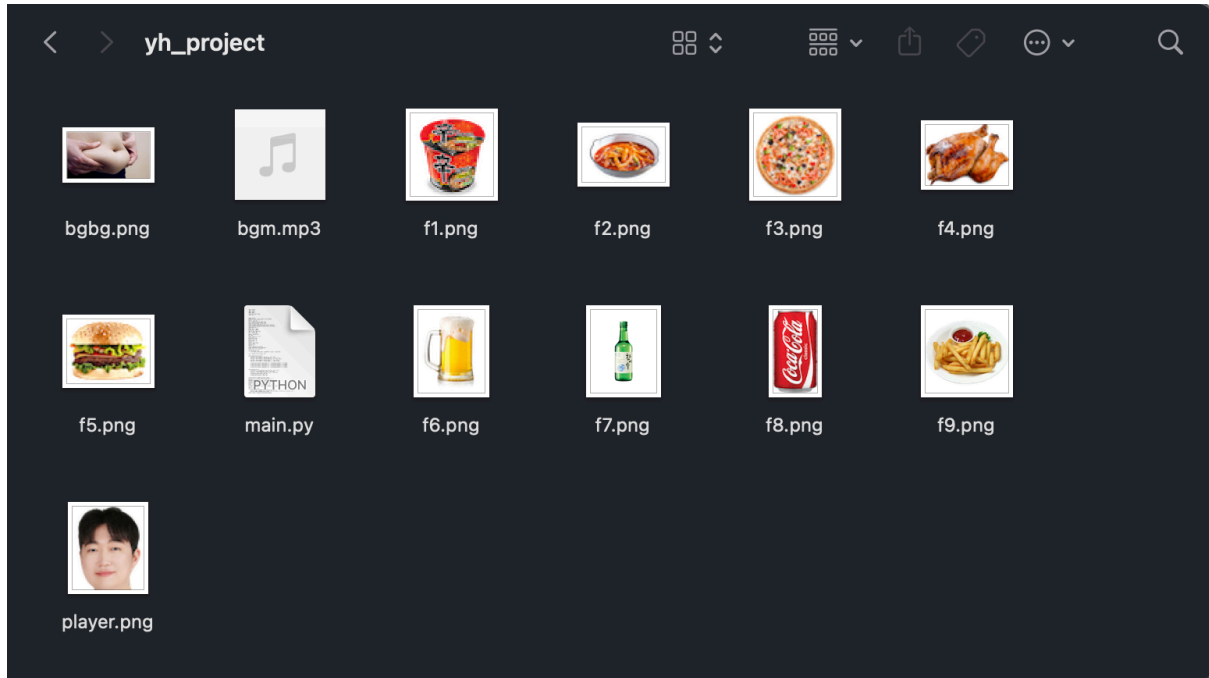


프로세스 플로우 차트



게임 전체적인 진행방향을 다이어그램으로 나타냈습니다.

게임리소스



플레이어, 9종 배드푸드, 뱃살 배경, bgm 등

코드설명

```
import pygame
import sys
import random
import math
from time import sleep
import os

pygame.init() #초기화
bullet_dir_add = ["UP","DOWN","LEFT","RIGHT"]
options = ["1","2","3"]
black = (0, 0, 0)
font = pygame.font.Font(None, 36)
map_width, map_height = 1200, 800
width, height = 1200, 800
screen = pygame.display.set_mode((width, height))
pygame.display.set_caption("Avoid trans fat foods!")
selected_option = ""
coin = 5
bullet_fast = 1000
bullet_dir = ["UP"]
white = (255, 255, 255)
```

```

red = (255, 0, 0)
blue = (0, 0, 255)
green = (0, 255, 0)
hp = 100
monster_response = 0.005
player_speed = 5
give_coin = True
bullet_size = 10
bullet_speed = 8
bullets = []
monster_size = 50
monster_speed = 1.7
monsters = []
score = 0
clock = pygame.time.Clock()
pygame.mixer.music.load('bgm.mp3')
pygame.mixer.music.play(-1)

def distance(p1, p2):
    return math.sqrt((p1[0] - p2[0])**2 + (p1[1] - p2[1])**2)

font = pygame.font.Font(None, 36) #폰트

def display_choices(color): #업그레이드 기능 화면에 띄우기
    choice0 = font.render("< Upgrade >", True, red)
    choice1 = font.render("1. Bullet Speed + ", True, color)
    choice2 = font.render("2. Bullet Direction +", True, color)
    choice3 = font.render("3. Bullet Size + ", True, color)
    choice4 = font.render("4. Player Speed + ", True, color)

    screen.blit(choice0, (map_width // 2 - choice0.get_width() // 2, 150))
    screen.blit(choice1, (map_width // 2 - choice1.get_width() // 2, 200))
    screen.blit(choice2, (map_width // 2 - choice2.get_width() // 2, 250))
    screen.blit(choice3, (map_width // 2 - choice3.get_width() // 2, 300))
    screen.blit(choice4, (map_width // 2 - choice4.get_width() // 2, 350))

def display_score(): #스코어 코인 체력 띄우기
    score_text = font.render(f"Score: {score}", True, red)
    hp_text = font.render(f"HP : {hp}", True, red)
    coin_text = font.render(f"COIN : {coin}", True, red)
    screen.blit(score_text, (10, 10))
    screen.blit(hp_text, (700, 10))
    screen.blit(coin_text, (350, 10))

last_shot_time = pygame.time.get_ticks()
player_image = pygame.image.load('./player.png') #플레이어 이미지 로드
bg_image = pygame.image.load('./bgbg.png') #배경 가져옴
playerSize = player_image.get_rect().size # 이미지의 크기를 구해옴
playerWidth = playerSize[0] # 플레이어의 가로 크기
playerHeight = playerSize[1] # 플레이어의 세로 크기

```

```

player_x = (map_width - playerWidth) // 2
player_y = (map_height - playerHeight) // 2
bg_image = pygame.transform.scale(bg_image, (1200, 800))

monster_images = [] #몬스터 이미지 로드
for i in range(1,10):
    monster_image = pygame.image.load(f'./f{i}.png')
    monster_image = pygame.transform.scale(monster_image, (monster_size,
monster_size))
    monster_images.append(monster_image)
monster_image = random.choice(monster_images)
while True:
    screen.blit(bg_image,(0,0))
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K_1:
                if coin > 0 :
                    coin -= 1
                    bullet_fast -= 50

            elif event.key == pygame.K_2:
                if coin > 0 and len(bullet_dir) < 4:
                    coin -= 1
                    bullet_dir.append(bullet_dir_add[len(bullet_dir)%4])

            elif event.key == pygame.K_3:
                if coin > 0 :
                    coin -= 1
                    bullet_size += 2

            elif event.key == pygame.K_4:
                if coin > 0 :
                    coin -= 1
                    player_speed += 1

    keys = pygame.key.get_pressed()
    player_x += (keys[pygame.K_RIGHT] - keys[pygame.K_LEFT]) * player_speed
    player_y += (keys[pygame.K_DOWN] - keys[pygame.K_UP]) * player_speed

    # 플레이어가 맵 경계에서 바운스
    if player_x < 0:
        player_x = 0
    elif player_x > map_width - playerWidth:
        player_x = map_width - playerWidth

```

```

if player_y < 0:
    player_y = 0
elif player_y > map_height - playerHeight:
    player_y = map_height - playerHeight

if random.random() < monster_response:
    monster_x = random.randint(0, map_width - monster_size)
    monster_y = random.randint(0, map_height - monster_size)
    monsters.append((monster_x, monster_y))

for i in range(len(monsters)):
    monster_x, monster_y = monsters[i]
    if monster_x < player_x:
        monster_x += monster_speed
    elif monster_x > player_x:
        monster_x -= monster_speed

    if monster_y < player_y:
        monster_y += monster_speed
    elif monster_y > player_y:
        monster_y -= monster_speed

    monsters[i] = (monster_x, monster_y)

for i in range(len(bullets)):
    bullet_x, bullet_y, direction = bullets[i]
    if direction == 'UP':
        bullet_y -= bullet_speed
    elif direction == 'DOWN':
        bullet_y += bullet_speed
    elif direction == 'LEFT':
        bullet_x -= bullet_speed
    elif direction == 'RIGHT':
        bullet_x += bullet_speed

    bullets[i] = (bullet_x, bullet_y, direction)

bullets = [
    (bx, by, direction) for bx, by, direction in bullets
    if 0 <= bx <= map_width and 0 <= by <= map_height
]

current_time = pygame.time.get_ticks()
if current_time - last_shot_time >= bullet_fast:
    for direction in bullet_dir:
        bullet_x = player_x + playerWidth // 2 - bullet_size // 2
        bullet_y = player_y + playerHeight // 2 - bullet_size // 2
        bullets.append((bullet_x, bullet_y, direction))
    last_shot_time = current_time

```

```

for monster_x, monster_y in monsters:
    if distance((player_x, player_y), (monster_x, monster_y)) < playerWidth:
        hp -= 1

for bullet_x, bullet_y, _ in bullets:
    for monster_x, monster_y in monsters:
        if (
            monster_x < bullet_x < monster_x + monster_size
            and monster_y < bullet_y < monster_y + monster_size
        ):
            bullets.remove((bullet_x, bullet_y, _))
            monsters.remove((monster_x, monster_y))
            score += 1
            break

if hp < 0 :
    pygame.quit()
    sys.exit()

if score % 5 == 0 and not give_coin:
    coin += 1
    give_coin = True
    monster_response += 0.01
    monster_image = random.choice(monster_images)

elif score % 5 != 0 :
    give_coin = False

if coin > 0 :
    display_choices(blue)
else :
    display_choices(white)

screen.blit(player_image, (player_x, player_y))

for monster_x, monster_y in monsters:
    screen.blit(monster_image, (monster_x, monster_y))

for monster_x, monster_y in monsters:
    screen.blit(monster_image, (monster_x, monster_y))

for bullet_x, bullet_y, _ in bullets:
    pygame.draw.rect(screen, green, (bullet_x, bullet_y, bullet_size,
bullet_size))
    screen.blit(player_image, (player_x, player_y))

```

```
display_score()

pygame.display.flip()

clock.tick(60)
```

코드 옆에 주석으로 코드에 대한 설명을 달아두었습니다.

실행환경

Python: 시스템에 Python이 설치되어 있어야 합니다. Python을 공식 웹사이트 (<https://www.python.org/>)에서 다운로드하고 해당 플랫폼에 맞는 설치 지침을 따르면 됩니다.

Pygame: Pygame은 2D 게임 및 멀티미디어 애플리케이션을 만들기 위한 Python 라이브러리입니다. Python 환경에 Pygame이 설치되어 있어야 합니다.

게임 리소스: 코드는 게임 자산에 대한 여러 이미지 파일을 필요로 합니다. 배경 이미지, 캐릭터 이미지 (누렁이), 미사일 이미지, 폭발 이미지 및 다양한 몬스터 이미지 (monster1.png, monster2.png 등)가 필요합니다. Python 스크립트와 동일한 디렉토리에 이러한 이미지가 있어야 합니다. 이러한 이미지는 PNG 형식이어야 합니다.

글꼴 파일: 코드는 게임 화면에 텍스트를 표시하기 위해 'NanumGothic.ttf' 글꼴 파일을 사용합니다. 실행 환경에 이 글꼴 파일을 사용할 수 있어야 합니다. 스크립트와 동일한 디렉토리에 있거나, 다른 위치에 있다면 글꼴 파일의 올바른 경로를 지정해야 합니다.

운영 체제: 코드는 Python과 Pygame이 설치된다면 여러 운영 체제(Windows, macOS, Linux)에서 호환될 것입니다.

하드웨어 요구 사항: 게임은 비교적 간단하며 특별한 하드웨어 요구 사항은 없을 것입니다.

Python 환경을 설정하고 Pygame을 설치하고 필요한 게임 자산을 Python 스크립트와 동일한 디렉토리에 배치한 후 스크립트를 실행할 수 있습니다. 게임 창이 열리고 키보드 입력(화살표 키와 스페이스바)을 사용하여 게임을 플레이할 수 있습니다.