

춘식이와 함께하는



도전! 군고구마 장사



원동진

#Leader #PM #AA



최희철

#DA #TA_assist



손미연

#DA #TA_assist



윤대호

#TA



채규비

#Art_Director #AA

1

기획

- 시장 조사
- 서비스 기획
- 게임 시나리오

2

게임 소개

- 기본 기능
- 디자인

3

코드 구현

- 모듈
- 플로우 차트
- 게임 소스



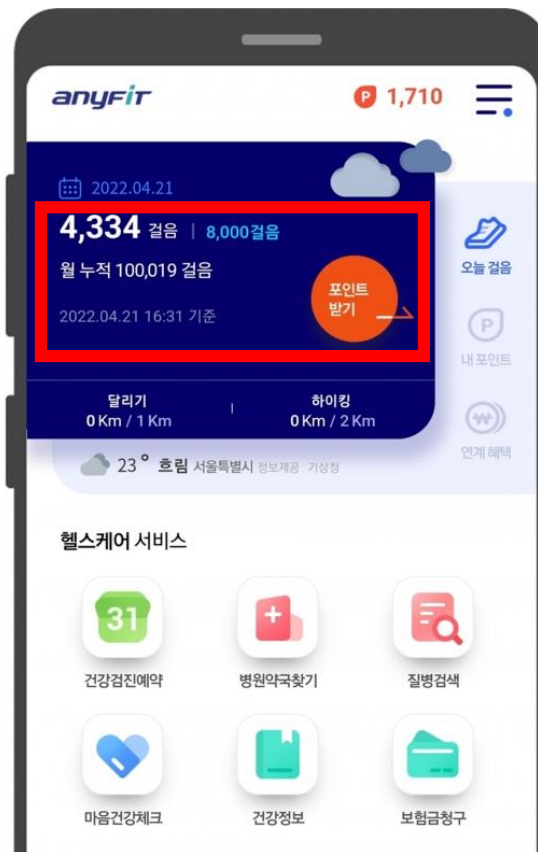
기획

“ 앱테크 ”

디지털 환경에서 **성실하게 이벤트**
등에 참여한 후 **포인트나 쿠폰**을 챙겨
생활비를 번다는 뜻에서 '**디지털**
폐지줍기'라는 이름으로도 불림

큰 품을 들이지 않고 **자신의 생활 방식**
이나 **흥미에 따라 즐겁게 참여**할
수 있다는 것이 장점

출처: 이데일리



출처: 보험저널

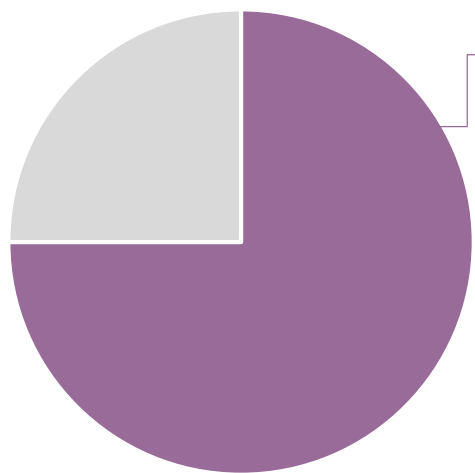
<삼성 화재 애니핏>

하루 **8,000걸음** 당
애니핏 **100포인트**로 전환 가능



전환한 애니핏 포인트로
보험료 결제 가능

<애플테크 현황>

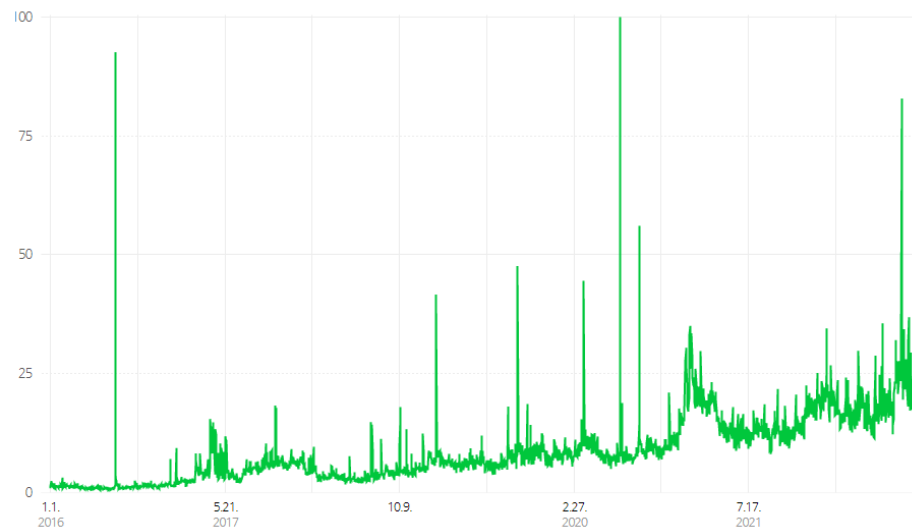


• **75.0%**
“하고 있다”

조사 대상: 성인남녀 1,707명
자료: 인쿠르트

→ 2030 성인 남녀 10명 중 7명은 애플테크 참여

<애플테크 언급량>



→ '애플테크' 관련 네이버 검색량 증가 추세

재미 요소를 활용한 앱테크 서비스로 카카오페이 앱 MZ 세대 유입 증가 기대

<앱테크 수익 현황>



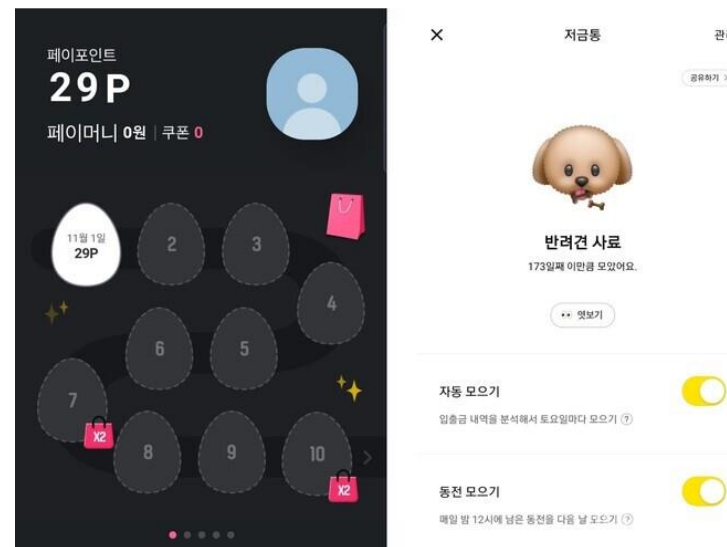
하루 평균
312원



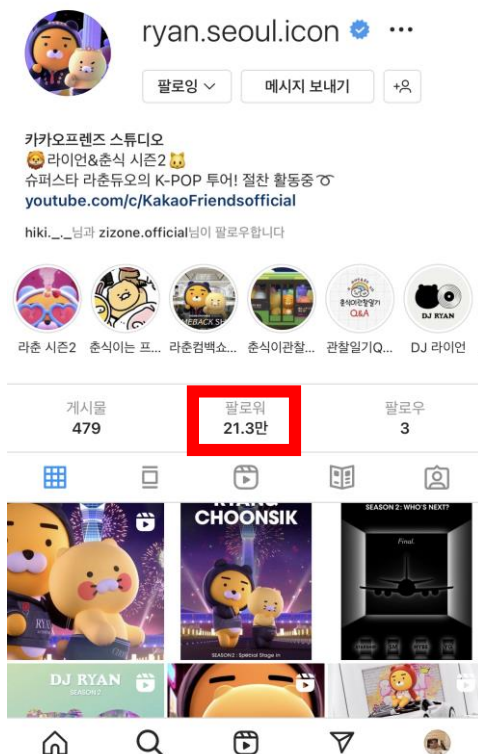
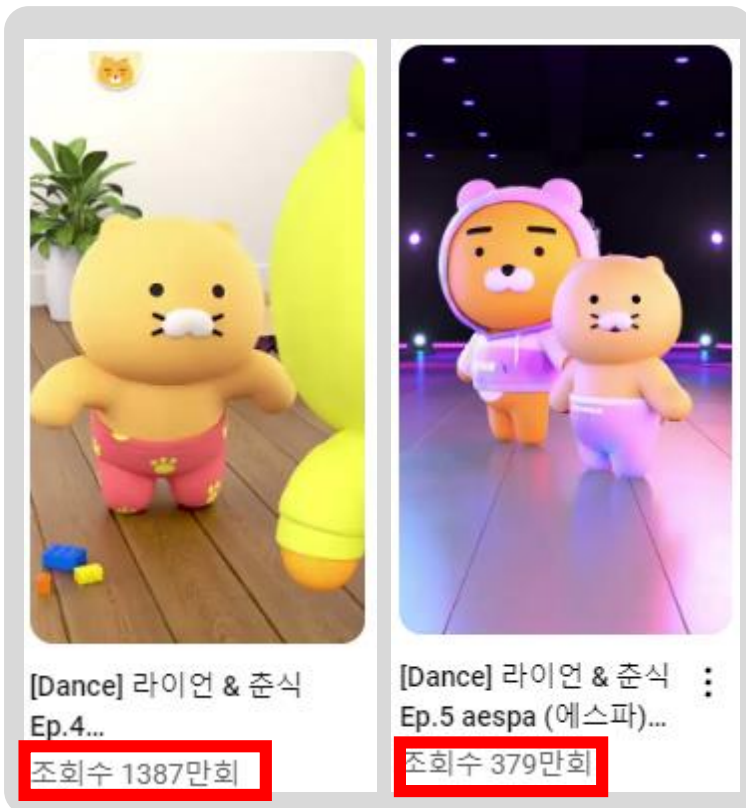
한 달 기준
~3천원 미만(37.2%)
가장 많음

→ 수익이 **소액**이라도 많이 참여

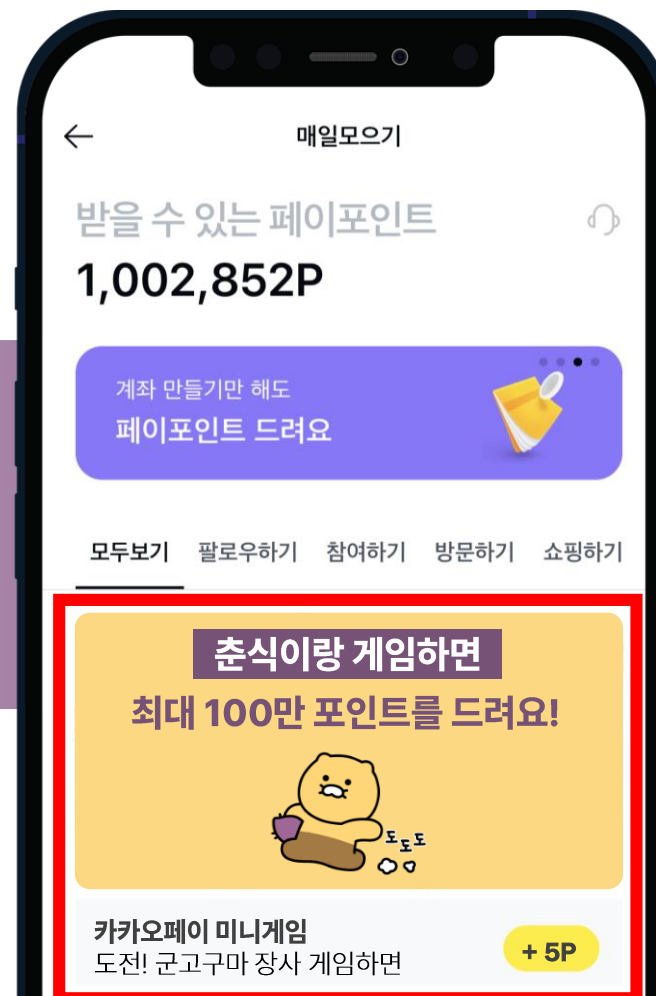
<카카오페이>



→ 카카오페이는 '알모으기', 저금통' 등
재미 요소를 추가하여 높은 유저 반응 획득



춘식을
활용한
앱테크
게임 기획



→ 카카오 프렌즈 캐릭터 중 MZ세대 대상 '춘식' 인기 급부상



시나리오

MZ 친구들이 붕어빵 장사를 시작하는 것을 보고
군고구마 장사를 시작한 춘식이!
하지만 날라오는 눈덩이에 장사 위기에 처하는데...
눈덩이를 피해 코인을 먹고 성공적으로 장사를 마쳐보자!

게임 방법

1. 날라다니는 눈덩이를 피해 군고구마를 지켜라!
2. 눈덩이에 맞아 **HP를 모두 잃으면 장사 실패**
3. 제한 시간 60초 안에 코인을 모아 장사를 성공시키자!

코인 리워드

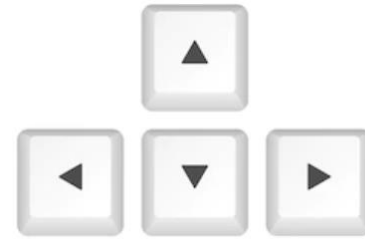
1. 10코인 → 5 카카오페이 포인트
2. 이벤트 기간 동안 **코인 누적 랭킹 1위 : 100만 카카오페이 포인트** 증정



게임 소개



메뉴얼



상, 하, 좌, 우로 이동



춘식이가 눈에 맞으면 HP 10점 감소



춘식이가 코인을 먹으면 Score 10점 증가



눈을 파괴할 수 있는 미사일 발사

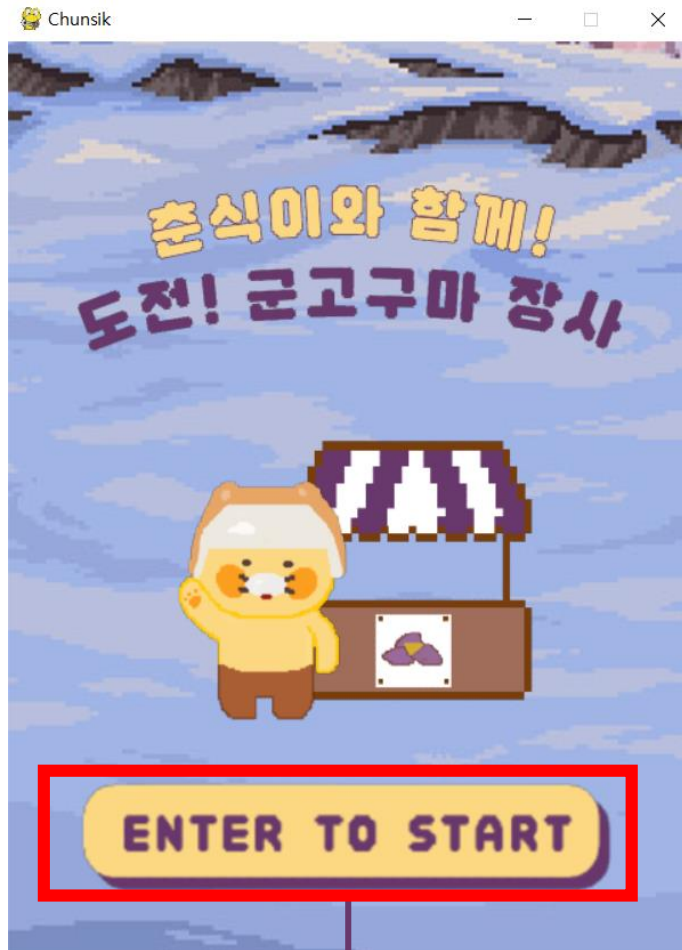


60초 시간 제한

Score : 점수

HP(Health Point): 체력

Time: 60초 제한

체력이 바닥나면
Game over

Enter키 누르면 게임시작



코드 구현

pygame

python으로 작성 가능한 게임 등의 멀티미디어 표현을 위한 모듈



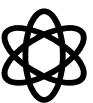
?

random

랜덤 관련한 함수들을 모아놓은 모듈

**os**

운영체제에서 제공되는 여러 기능을 파이썬에서 수행할 수 있게 해 줌

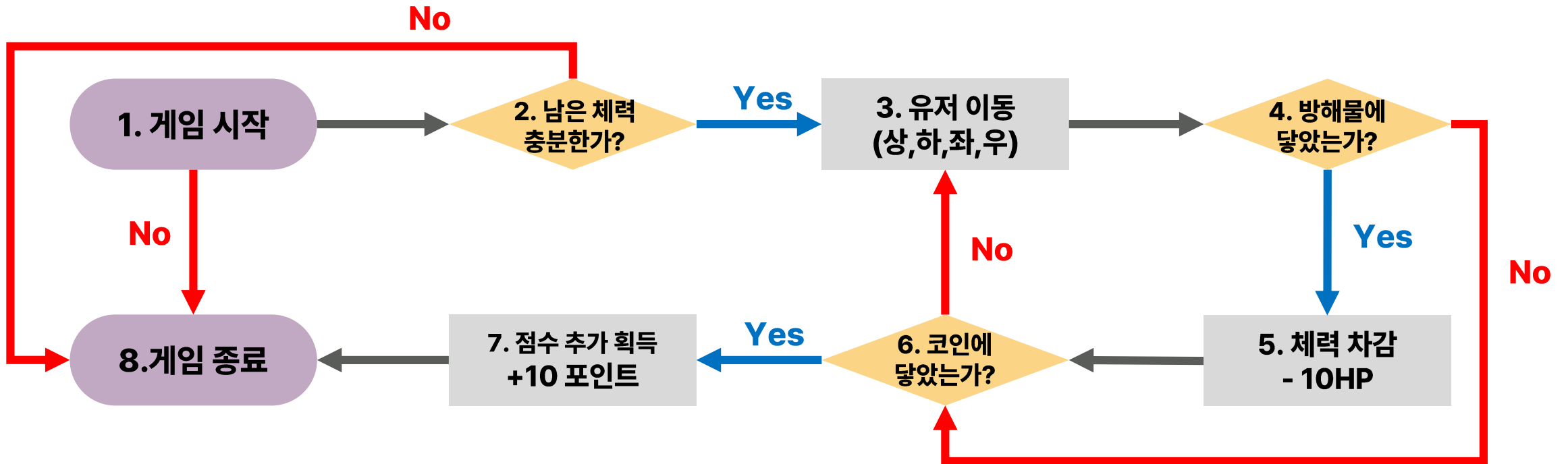
**sys**

파이썬이 제공하는 변수나 함수를 제어할 수 있는 방법 제공

**time**

시간 표현 및 계산을 다루는 모듈

Moduel



***2~7번 플로우 중 남은 시간 = 0이 될 경우 즉시 8번으로 이동**

SUMMARY

- 1 Import 모듈
- 2 def initialize_game(게임창 설정)
- 3 def game_loof(게임 로직)**
- 4 def score_update(상단 점수바)
- 5 def gameover(게임 중지)
- 6 def close_game(게임창 닫기)
- 7 def restart(게임 재시작)
- 8 class PlayerShip(캐릭터 클래스)
- 9 class Coin(코인 클래스)
- 10 class Mob(장애물 클래스)
- 11 class Bullet(미사일 클래스)
- 12 모듈 실행



게임창/게임로직 함수

```
import pygame
import random
import os
import sys
import time
```

----- 게임창 설정 -----

```
def initialize_game(width, height):
    pygame.init()
    surface = pygame.display.set_mode((width, height)) # 화면 표면반환
    pygame.display.set_caption("Chunsik") # 캡션 이름 'Chunsik'
    return surface
```

----- 게임 로직 -----

```
def game_loop(surface):
    background = pygame.image.load('background.png') # 게임진행 배경화면
    background2=pygame.image.load('enter_to_start.png') # 게임실행 배경화면
    bg_rect = background.get_rect()
    clock = pygame.time.Clock()
    sprite_group = pygame.sprite.Group()
    mobs = pygame.sprite.Group() # 장애물
    bullets = pygame.sprite.Group() # 미사일
    player = PlayerShip() # 캐릭터
    coins = pygame.sprite.Group() # 코인
```

```
global play_time, total_time
play_time=60 # 전역함수, 플레이 시간
total_time=60 # 제한시간 60초
```

```
global player_health
player_health= 100 # 전역함수, 캐릭터 생명력
# 제한 100
```

```
global score
score = 0 # 전역함수 점수
sprite_group.add(player)
for i in range(7): # 한번에 생성되는 눈, 코인 수
    enemy = Mob()
    item = Coin()
    sprite_group.add(item)
    sprite_group.add(enemy)
    mobs.add(enemy)
    coins.add(item)
```



시작화면 삽입/장애물과 코인 충돌 시 이벤트 처리/플레이 시간 출력

```
running = False
while 1:
    # 시작 화면 삽입, enter키 눌렀을 때 게임 시작
    if not running:
        surface.blit(background2, (0, 0))
        pygame.display.flip()
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                exit(0)
            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_RETURN:
                    running = True
                    start_ticks = pygame.time.get_ticks()

    else:
        # 마우스클릭 하거나 스페이스 버튼 클릭시 슈팅
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                running = False
            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_q:
                    running = False
                if event.key == pygame.K_SPACE:
                    player.shoot(sprite_group, bullets)
            if event.type == pygame.MOUSEBUTTONDOWN:
                player.shoot(sprite_group, bullets)

    sprite_group.update()
```

```
# 춘식이와 눈 충돌시 HP 10씩 감소, 0이 되면 gameover 화면 5초 유지 후 게임중지, 리스타트
hits = pygame.sprite.spritecollide(player, mobs, True)
if hits:
    print('a mob hits player!')
    player_health -= 10
    if player_health < 0:
        gameover(surface)
        close_game()
        restart()

# 춘식이가 코인 획득시 score 10씩 증가
gets = pygame.sprite.spritecollide(player, coins, True)
if gets:
    print('player gets a coin!')
    score += 10

# 경과 시간 계산
elapsed_time = (pygame.time.get_ticks() - start_ticks) / 1000 # ms -> s
textfont=pygame.font.Font('PixelGameFont.ttf', 60)
play_time=int(total_time - elapsed_time)

# 제한시간 60초 초과했다면 Time over 출력 및 게임 중지
if total_time - elapsed_time < 0:
    game_result = "Time Over"
    textfont=pygame.font.Font('PixelGameFont.ttf', 60)
    text=textfont.render("Time Over",True,BLACK)
    textpos=text.get_rect()
    textpos.center=(SCREEN_WIDTH/2,SCREEN_HEIGHT/2)
    screen.blit(text,textpos)
    time.sleep(5)
    running = False

surface.fill(LIGHT_PINK1)
surface.blit(background, bg_rect)
sprite_group.draw(surface)
score_update(surface)
pygame.display.flip()

pygame.quit()
print('game played: ',playtime)
```

상단 점수바/GAME OVER 화면 함수

```
# 상단 점수바 폰트, 위치 설정
def score_update(surface):
    font = pygame.font.Font('PixelGameFont.ttf', 30)
    image = font.render(f' Score : {score} HP: {player_health} Time: {play_time}', True, BLUE2)
    pos = image.get_rect()
    pos.move_ip(10, 10)
    pygame.draw.rect(image, BLACK, (pos.x-10, pos.y-10, pos.width, pos.height), 2)
    surface.blit(image, pos)

# gameover 화면 띄우기
def gameover(surface):
    gameover_bg = pygame.image.load('GAME_OVER_FULL.png')
    surface.blit(gameover_bg, (0, 0))
    pygame.display.update()
    time.sleep(5)
```

게임창 닫기/재시작 함수 정의

```
# 게임창 닫기
def close_game():
    pygame.quit()
    print('Game closed')

# 게임창 리스타트
def restart():
    screen = initialize_game(SCREEN_WIDTH, SCREEN_HEIGHT)
    game_loop(screen)
    close_game()
```

캐릭터 동작/미사일 발사 클래스 생성

```
# 캐릭터 동작
class PlayerShip(pygame.sprite.Sprite):
    def __init__(self):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.image.load('chunsik.png') # 캐릭터 이미지 삽입
        self.rect = self.image.get_rect()
        self.rect.centerx = int(SCREEN_WIDTH / 2) # 캐릭터 시작 위치 정의 (화면 가운데)
        self.rect.centery = SCREEN_HEIGHT - 20
        self.speedx = 0
        self.speedy = 0

# 춘식이 움직임, 속도, 화면에서 벗어나지 않도록 경계값 설정
def update(self):
    self.speedx = 0
    self.speedy = 0
    keystate = pygame.key.get_pressed()
    if keystate[pygame.K_LEFT]:
        self.speedx = -5
    if keystate[pygame.K_RIGHT]:
        self.speedx = 5
    if keystate[pygame.K_UP]:
        self.speedy = -5
    if keystate[pygame.K_DOWN]:
        self.speedy = 5
    self.rect.x += self.speedx
    self.rect.y += self.speedy

# 캐릭터 화면에서 벗어나지 않게 함수 정의
if self.rect.right > SCREEN_WIDTH:
    self.rect.right = SCREEN_WIDTH
if self.rect.left < 0:
    self.rect.left = 0
if self.rect.bottom > SCREEN_HEIGHT:
    self.rect.bottom = SCREEN_HEIGHT
if self.rect.top < 0:
    self.rect.top = 0

# 미사일 발사
def shoot(self, all_sprites, bullets):
    bullet = Bullet(self.rect.centerx, self.rect.top)
    all_sprites.add(bullet)
    bullets.add(bullet)
```

코인 생성 및 동작 클래스 생성

```
class Coin(pygame.sprite.Sprite):
    def __init__(self): # 클래스 초기화
        pygame.sprite.Sprite.__init__(self) # 스프라이트 초기화
        self.image = pygame.image.load('coin.png')
        self.image = pygame.transform.scale(self.image, (30,30))
        self.rect = self.image.get_rect() # 스프라이트로 그려준다
        # 코인의 생성 위치와 속도 랜덤하게 설정
        self.rect.x = random.randrange(SCREEN_WIDTH - self.rect.width)
        self.rect.y = random.randrange(-100, -40)
        self.speedy = random.randrange(1, 8)
        self.speedx = random.randrange(-3, 3)
        self.direction_change = False

    # 코인 움직임
    def update(self):
        self.rect.x += self.speedx
        self.rect.y += self.speedy

        if self.rect.top > SCREEN_HEIGHT + 10 or self.rect.left < -25 or self.rect.right > SCREEN_WIDTH + 20:
            self.rect.x = random.randrange(SCREEN_WIDTH - self.rect.width)
            self.rect.y = random.randrange(-100, -40)
            self.speedy = random.randrange(3, 8)
```

장애물 생성 및 동작 클래스 생성

```
class Mob(pygame.sprite.Sprite):
    def __init__(self):
        pygame.sprite.Sprite.__init__(self)
        # 눈 크기 랜덤하게 선택하여 생성
        snowImage=['snow_S.png','snow_M.png','snow_L.png']
        self.image = pygame.image.load(random.choice(snowImage))
        self.rect = self.image.get_rect()
        # 코인의 생성 위치와 속도 랜덤하게 설정
        self.rect.x = random.randrange(SCREEN_WIDTH - self.rect.width)
        self.rect.y = random.randrange(-100, -40)
        self.speedy = random.randrange(1, 8)
        self.speedx = random.randrange(-3, 3)
        self.direction_change = False

    # 눈 움직임
    def update(self):
        self.rect.x += self.speedx
        self.rect.y += self.speedy

        if self.rect.top > SCREEN_HEIGHT + 10 or self.rect.left < -25 or self.rect.right > SCREEN_WIDTH + 20:
            self.rect.x = random.randrange(SCREEN_WIDTH - self.rect.width)
            self.rect.y = random.randrange(-100, -40)
            self.speedy = random.randrange(3, 8)
```

총알 생성 및 동작 클래스 생성

```
class Bullet(pygame.sprite.Sprite):
    def __init__(self, player_x, player_y):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.Surface((10,20))
        self.image.fill(GREEN1)
        self.rect = self.image.get_rect()
        self.rect.bottom = player_y
        self.rect.centerx = player_x
        self.speedy = - 10

    def update(self):
        self.rect.y += self.speedy
        if self.rect.bottom < 0:
            self.kill()
```

게임 시작

```
if __name__ == '__main__':
    screen = initialize_game(SCREEN_WIDTH,SCREEN_HEIGHT)
    game_loop(screen)
    sys.exit()
```

Q&A