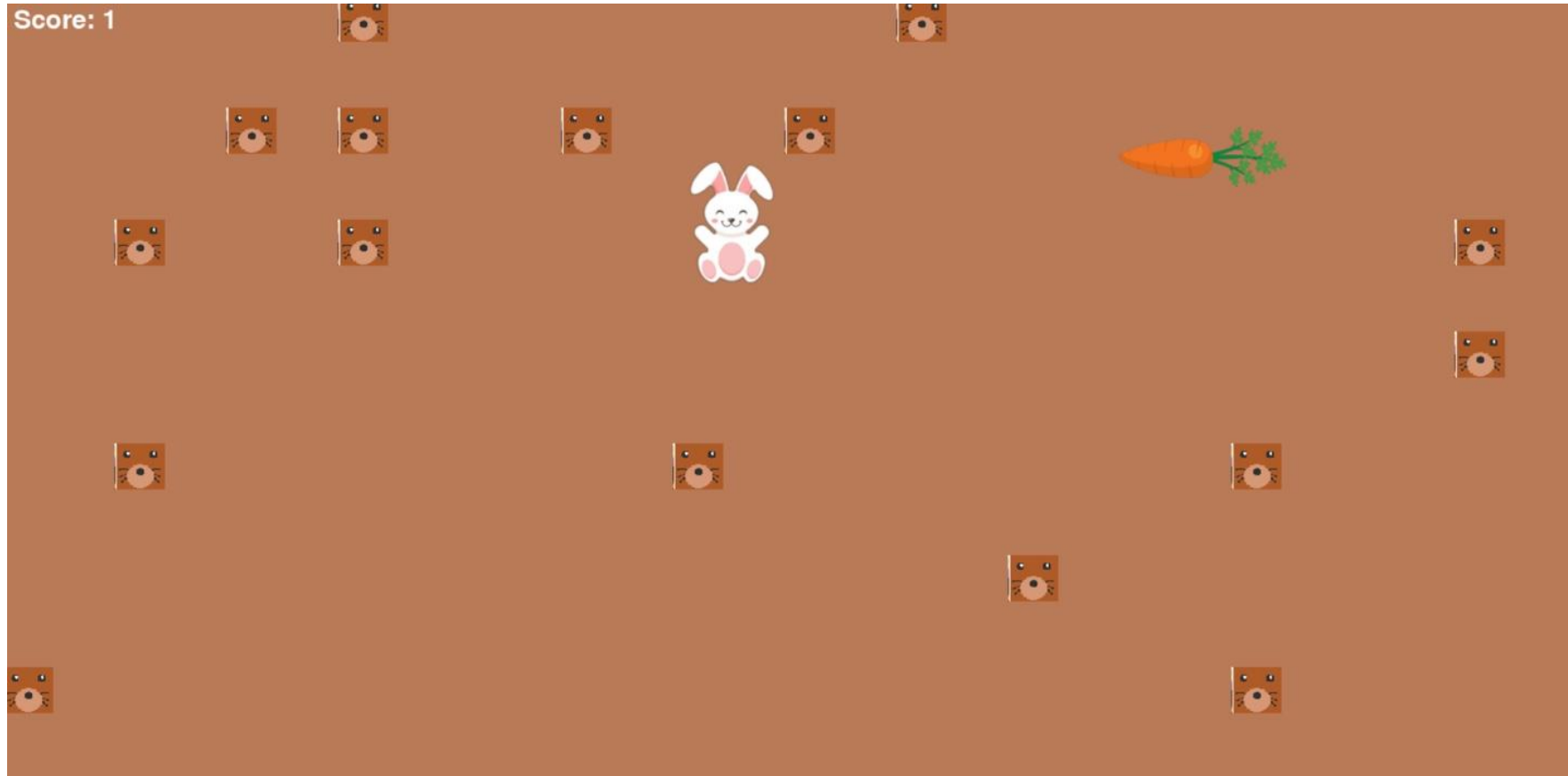




23년 1학기 게임 소개 – 토끼 게임

게임 소개 – 토끼 게임



토끼가 당근 10개를 먹으면 성공하는 게임

게임 소개 – 토끼 게임



두더지에 닿으면 GAME OVER

게임 소개 - 토끼 게임

```
# 적캐릭터 = 두더지
class Mole:
    def __init__(self, x_pos, y_pos):
        self.mole = pygame.image.load("C:/Users/Jihun/Desktop/2023CPU 활동")
        self.mole_size = self.mole.get_rect().size
        self.mole_width = self.mole_size[0]
        self.mole_height = self.mole_size[1]
        self.mole_x_pos = x_pos
        self.mole_y_pos = y_pos
        self.appear = False
        self.AppearTime = 0
```

```
mole_list= [] #두더지 리스트 생성 : 여러개의 객체를 리스트에 차곡차곡 넣을예정

num_x_moles = 14 # 생성할 두더지 개수 (두더지생성 반복회수) : 행 (가로)
num_y_moles = 7 # 열 (세로)
```

```
# 반복문으로 두더지 객체 생성
mole_x_pos = 0
mole_y_pos = 0 # 두더지 생성반복문의 초기 x, y 좌표
```

```
# 배치할 간격 계산
x_interval = screen_width // num_x_moles
y_interval = screen_height // num_y_moles
```

```
# 반복문으로 두더지 객체 생성
for i in range(num_x_moles):
    for j in range(num_y_moles):
        mole = Mole(mole_x_pos, mole_y_pos)
        mole_list.append(mole)

        mole_y_pos += y_interval # y좌표를 간격만큼 증가
        mole_y_pos = 0 # 해당 열이 끝났으므로 y 좌표를 0으로 초기화

        mole_x_pos += x_interval # x 좌표를 간격만큼 증가
```

```
# 시간 경과에 따른 두더지 등장 여부 결정
time_elapsed += dt
if time_elapsed >= 1000: # 1초가 지났을 때
    mole_list[randrange(0, num_x_moles*num_y_moles+1 )].appear = True
    time_elapsed = 0 # 경과한 시간 초기화

for mole in mole_list :
    if mole.appear == True:
        mole.AppearTime += dt
        if mole.AppearTime >= 3000:
            mole.appear = False
```

두더지객체를 생성하는 클래스 선언 후
각 좌표에 두더지객체 생성,
객체리스트 만들어서 집어넣음

1초마다 객체리스트에서 랜덤으로 두더지 소환
그리고 소환된 객체두더지는 각각 3초 후 사라
짐

게임 소개 - 토끼 게임

[충돌처리]

#두더지

```
for mole in mole_list:      # 리스트의 두더지 appear값이 True라면 두더지를 그린다
    if mole.appear:
        screen.blit(mole.mole, (mole.mole_x_pos, mole.mole_y_pos))
```

객체리스트의 각 두더지.appear=True일때에만 그림

```
# 충돌처리를위해 mole의 rect 정보업데이트 + 충돌 체크
for mole in mole_list:
    if mole.appear:
        mole_rect = mole.mole.get_rect()
        mole_rect.left = mole.mole_x_pos
        mole_rect.top = mole.mole_y_pos
        if character_rect.colliderect(mole_rect):
            print("Crash!")
            running = False
```

if 두더지.appear=True 일 때에만 충돌처리 되도록 함

#당근

```
carrot_rect = carrot.get_rect()
carrot_rect.left = carrot_x_pos
carrot_rect.top = carrot_y_pos

if character_rect.colliderect(carrot_rect):
    score +=1
    carrot_x_pos = randrange(0, screen_width-carrot_width) #
    carrot_y_pos = randrange(0, screen_height-carrot_height) #
```

두더지객체의 appear값이 True일때에만
화면에 두더지를 그리고, 충돌하도록 함

캐릭터가 당근에 닿으면
랜덤위치에서
당근이 재생성 되도록 함



THANK YOU