

기획서 및 프로젝트(미니) 심사평가표

과정명 : 의학연구 및 AI 의료서비스 모델 개발 과정(4회차)

발표일일시	2025.11.19	훈련과정	프로그래밍 (Python)	발표자	발표: 1팀 전체 시연: 김용기
프로젝트명	파이게임	팀명 팀장,팀원	팀명: 1팀 팀장: 박수연 팀원: 김용기, 박윤정, 유성빈	평가점수	(인)
프로젝트 핵심내용	1. 개요: 심근경색 인식 개선을 위한 간단한 게임 2. 주요기능: 아이템, 필살기, 제한시간, 방해 시스템 3. 사용툴: Visual Studio, Python 3.12, (pygame)				
프로젝트 주요내용 (대표 이미지)	1. 스토리텔링 - 게임 주제 및 보건의료 분야 연관성 심근경색을 주제로 하여 혈관 내에 쌓여있는 방해 요소들을 제거하는 게임 - 게임 목적 및 교육적 의미 게임을 플레이하는 사람들에게 심근경색에 좋은 영양소를 기억에 남게 하기 위함 - 스토리 개요 심근경색이 오기 전에, 지질의 방해를 피해 혈관 내 벽돌들을 제거하자! 2. 주요 기능 체크리스트 <input type="checkbox"/> 아이템 시스템 <input type="checkbox"/> 필살기 <input type="checkbox"/> 제한시간 <input type="checkbox"/> 방해 시스템 3. 각 기능별 상세 설명 <ul style="list-style-type: none">아이템 시스템: 심근경색에 좋은 영양소를 게임 클리어에 도움이 되는 아이템으로 설정필살기: 방해 요소를 한 번에 처리할 수 있는 스킬제한 시간: 제한 시간 250초 내에 플레이를 하도록 설정.방해 시스템: 지질 축적은 심근경색 및 심혈관 질환의 주요 원인 중 하나로 알려져 있기 때문에, 게임 속 방해꾼으로 설정해 “지질 = 위험 요소” 라는 개념의 직관적 이해를 도움				

<시작화면>



1UP
0

HIGH SCORE

20730

250

item



EXTRA LIFE

GAIN AN ADDITIONAL
LIFE



EXPAND

EXPANDS THE PADDLE



DUPLICATE

DUPLICATES THE BALL



SLOW

SLOW DOWN THE SPEED



REDUCE

REDUCES THE PADDLE



SPEEDUP

SPEED UP THE BALL

SPACEBAR TO START

OR ENTER LEVEL

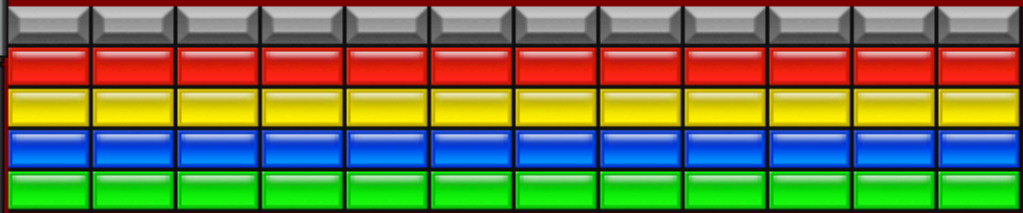
ARKANOID

1UP
0

HIGH SCORE

20730

244



ROUND 1

READY



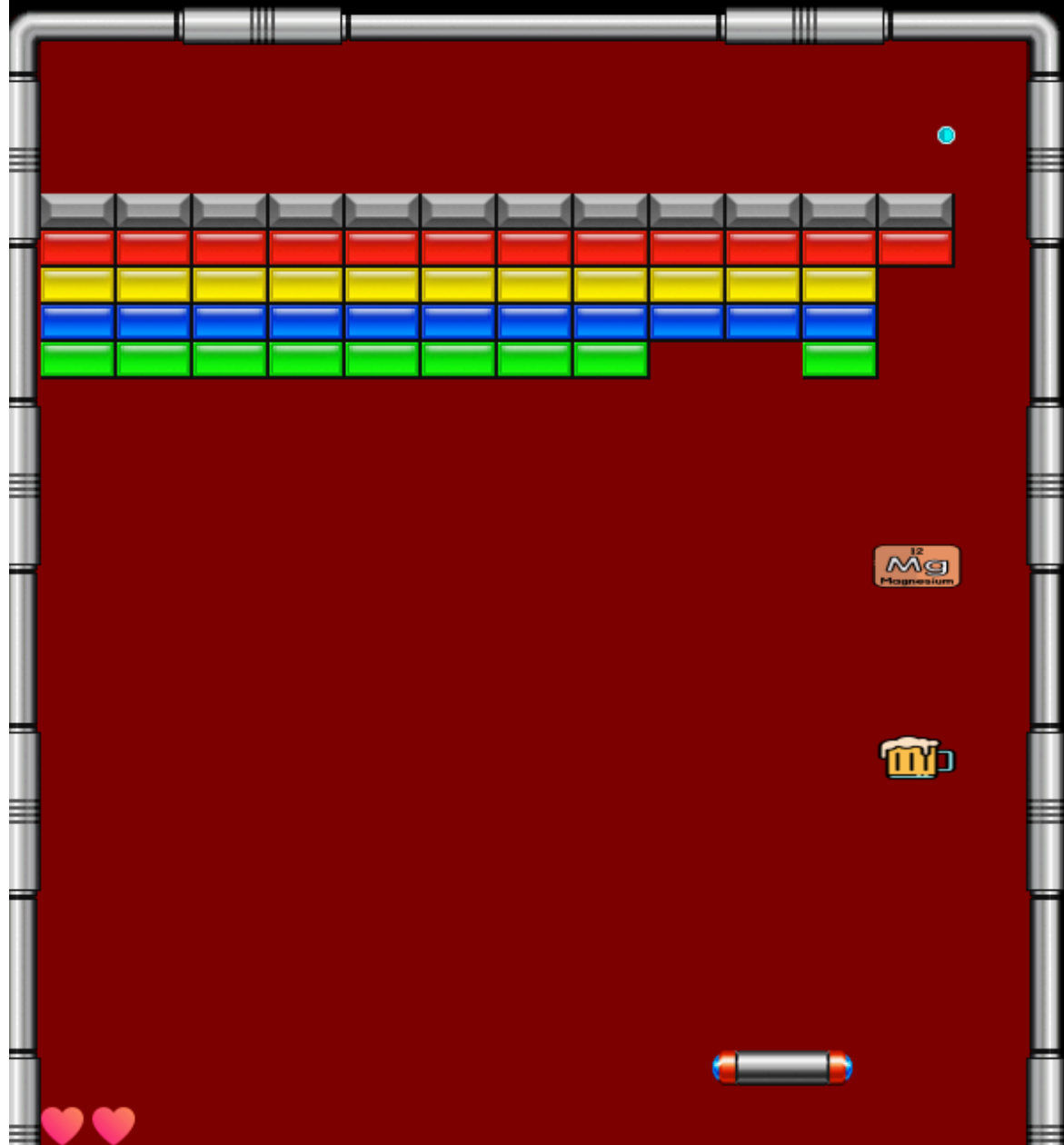
□ 기본화면 (게임 시작)

ARKANOID

1UP
460

HIGH SCORE
20730

240



□ 기본화면 (아이템)

ARKANOID

1UP
380

HIGH SCORE

20730

224



GAME OVER

□ 기본화면 (게임 오버)

□ 필살기

개발중입니다

- 제한 시간 타이머, 위험요소 출현을 위한 타이머
- 게임 진행: 공 상하좌우, 패들 좌우 움직임
- 충돌 감지: 벽돌인 경우와 아닌 경우로 분류
- 공이 바닥에 떨어졌는지 여부로 생명 차감

- 코드

```
class Game:
    """Represents a running Arkanoid game.

    An instance of a Game comes into being when a player starts a new
    game.
    """

    def __init__(self, round_class=Round1, lives=3):
        """Initialise a new Game.

        Args:
            round_class:
                The class of the round to start, default Round1.
            lives:
                Optional number of lives for the player, default 3.
        """

        # Keep track of the score and lives throughout the game.
        self.lives = lives
        self.score = 0

        # Reference to the main screen.
        self._screen = pygame.display.get_surface()

        # The life graphic.
        self._life_img, _ = load_png('paddle_life.png')
        # The life graphic positions.
        self._life_rects = []
```