1팀 박수연, 김용기, 박문정, 유성빈

목차

01
게임 주제

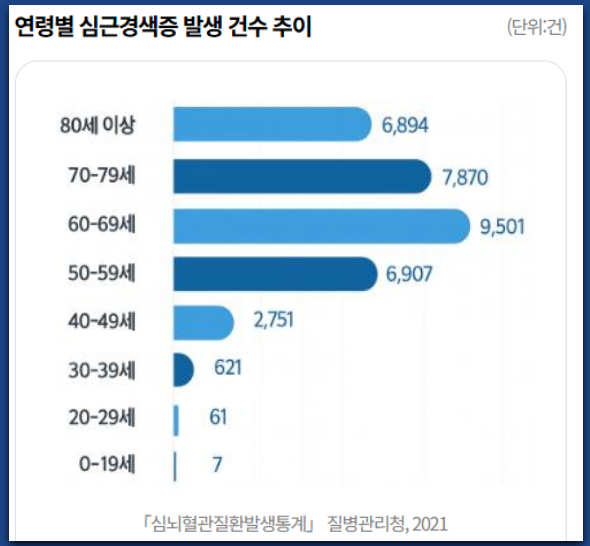
선정 배경과 게임 컨셉

선정 배경

- 4년간 10만 명당 심근경색 발생률은 65~68명 수준을 유지
- 줄어들지 않고 꾸준히 발생하는 주요 심혈관 질환

출처: 질병관리청(2024) [심뇌혈관질환 발생통계] 심근경색증 (I21~I23) 상병코드로 입원한 환자(국민건강보험빅데이터)

선정 배경



- 60대에서 가장 많은 9,501명의 발생
- 20~30대도 60~600명 수준으로 발생

심근경색증 발생률 (단위:명/10만명당)

남자	여자	평균
99.4	35.6	67.4

「심뇌혈관질환발생통계」 질병관리청 2021

- 남성(99.4명)은 여성(35.6명)보다 약 3배 높은 발생률

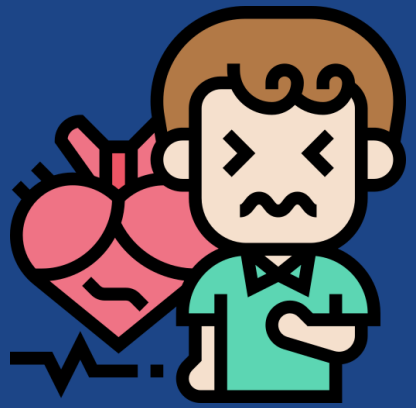
출처: 질병관리청(2024) [심뇌혈관질환 발생통계] 심근경색증 (I21~I23) 상병코드로 입원한 환자(국민건강보험빅데이터)

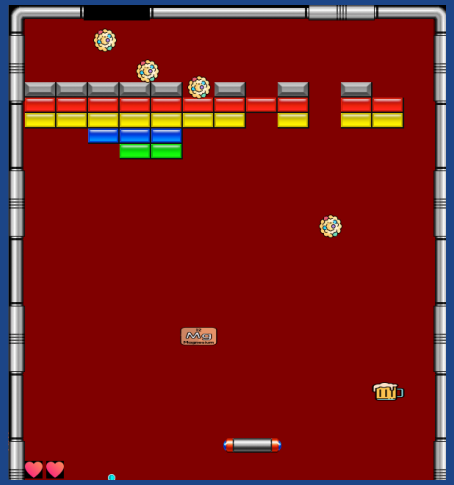
게임 컨셉

왜 심근경색과 알카노이드인가?

심근경색의 골든타임은 초 단위 싸움

알카노이드도 초 단위 판단이 생존을 결정





알카노이드 게임 화면

"아이템 하나를 놓치면 판이 뒤집히는 것처럼, 심근경색도 한순간 대응이 생사를 바꾼다"

기대효과

게임을 통해 플레이어에게
심근경색에 도움이 되는 영양소와 해로운 요소를 쉽게 기억할 수 있도록 한다

02
활용 방안

02 활용 방안

타겟 사용자

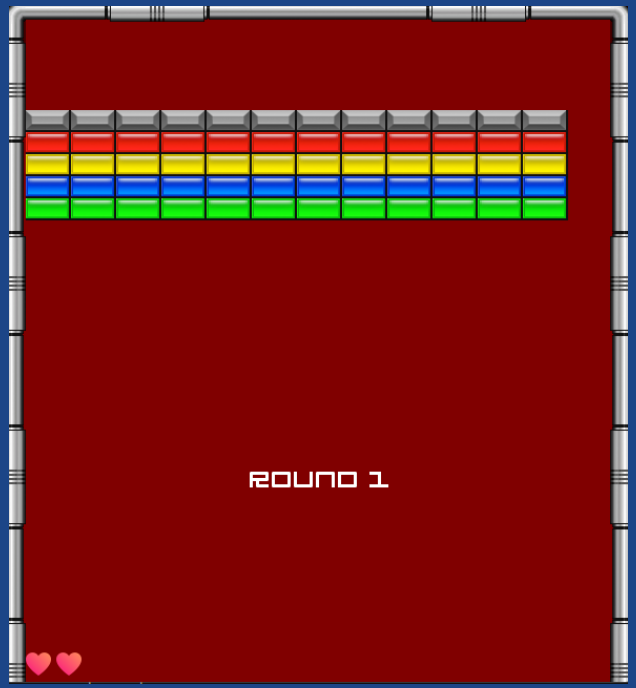
현장에서의 활용 방식

현장활용을 위한 협업방안

보건복지부

03
게임 설정

게임 리소스



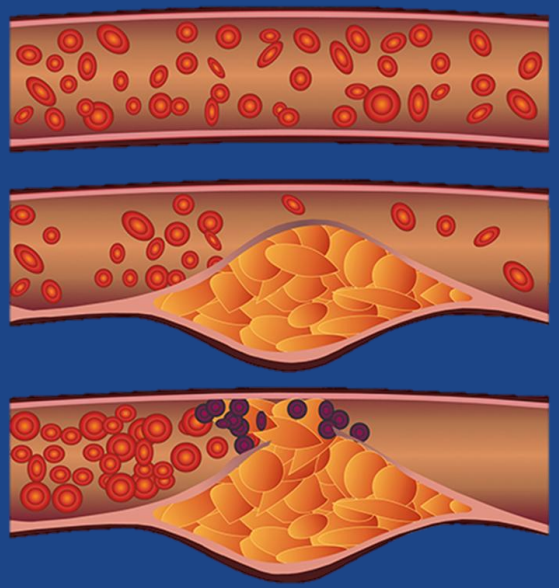
배경

혈관의 느낌을 주기 위해 배경을
붉은색으로 설정

방해 시스템

방해꾼

혈류의 방해꾼, 지질



이미지 출처: [서울 아산병원 뉴스룸] 심뇌혈관질환의 씨앗 '고지혈증' 제대로 알고 관리하자

아이템 시스템

비타민 D (공 복제)

혈압 조절, 염증 감소

오메가3 (슬로우)

혈액 순환을 개선

음주 (스피드업)

혈압 상승,
심박 리듬 불안정

마그네슘 (패들 확장)

혈관 이완,
심장 리듬 조절 도움

심폐소생술 (생명 추가)

혈류 개선 (골든타임 연장)

수면 부족 (패들 축소)

심장 기능 부담

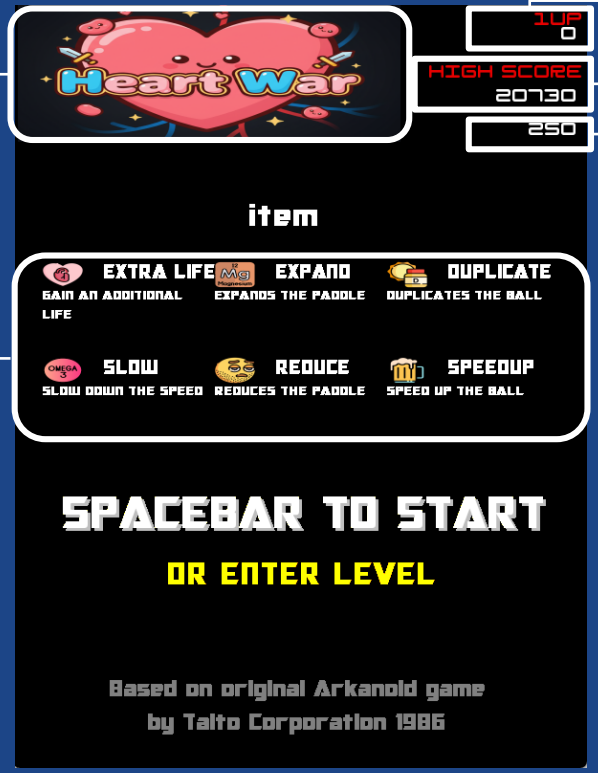
필살기

- 랜덤으로 출현
  - 각 라운드 하나
- 사용 시 화면의 지질(방해꾼) 제거
  -  키 사용

04
게임 구성

튜토리얼 & 워크플로우

튜토리얼



로고

아이템

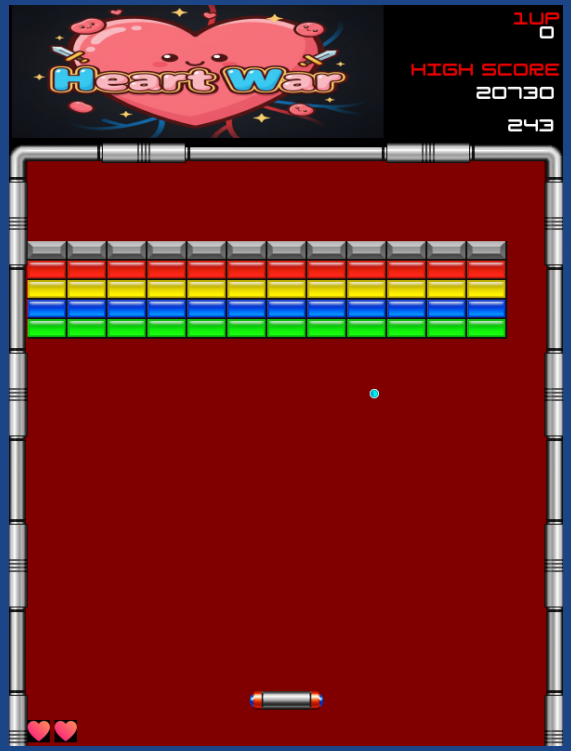
현재 점수

최고 점수

제한 시간 타이머

스페이스바 눌러 시작!

튜토리얼



패들 조작

필살기

S 키 눌러 사용

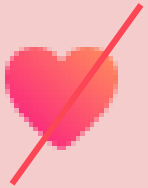
방해꾼

게임 방식

게임 목표

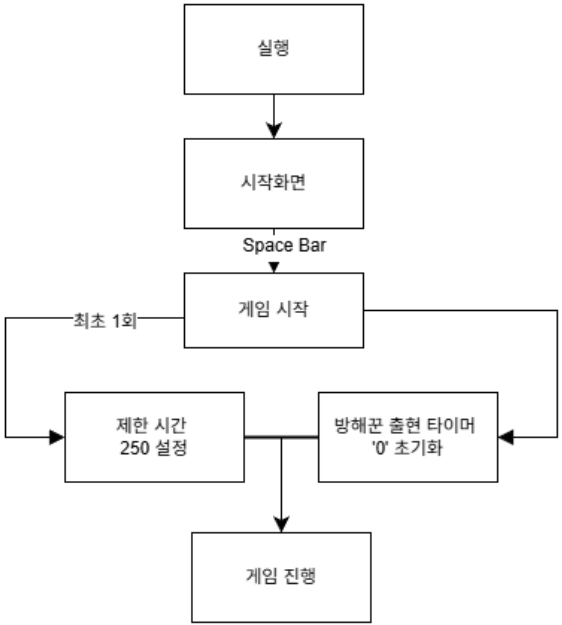
클리어 조건

게임 오버

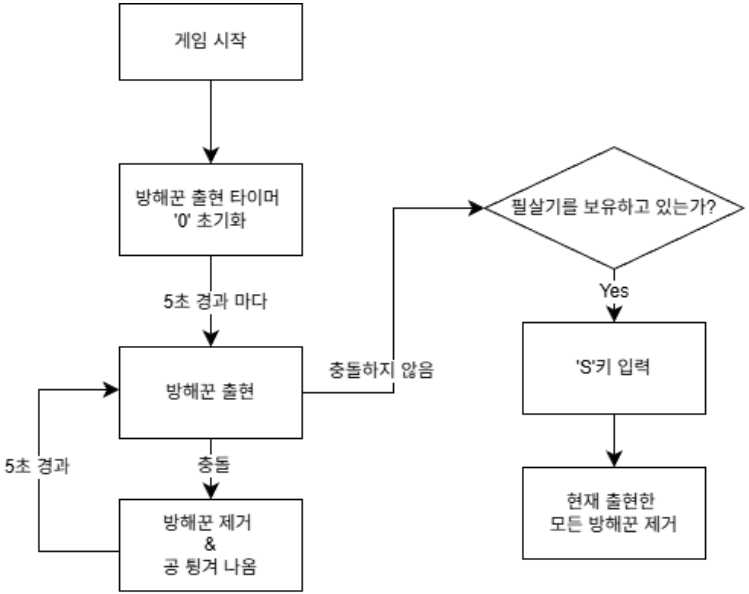


워크플로우

게임 시작 및 진행

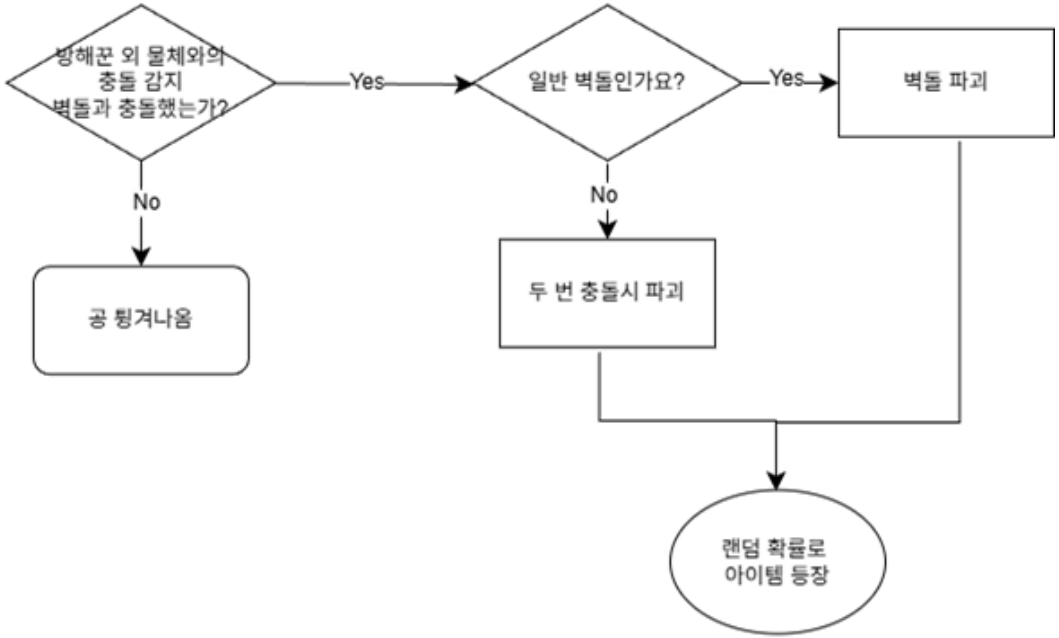


방해꾼 시스템



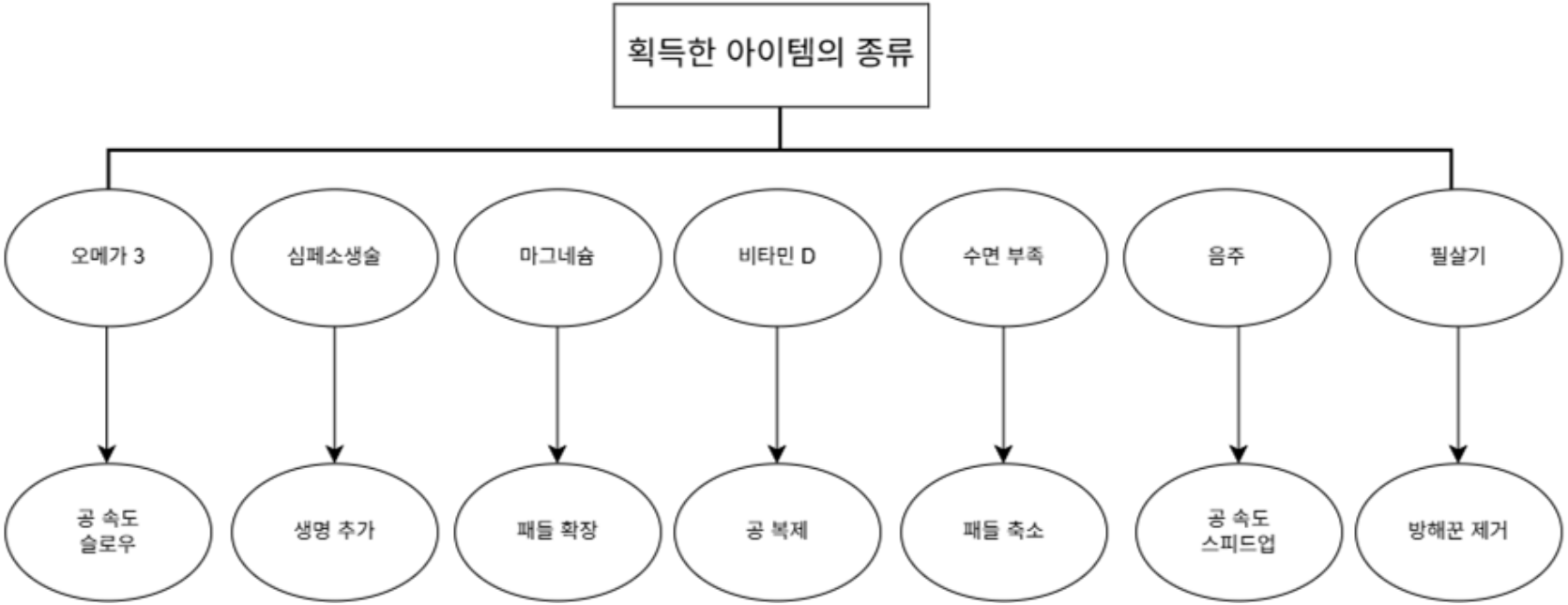
워크플로우

충돌 감지


방해꾼 외 물체와의
충돌 감지
벽돌과 충돌했는가?
Yes
일반 벽돌인가요?
Yes
벽돌 파괴
No
No
공 튕겨나옴
두 번 충돌시 파괴
랜덤 확률로
아이템 등장

워크플로우

아이템 시스템



05
게임 코드 구성

코드 구성 -1 패널

```python
def update(self):

    self._state.update() #상태 업데이트 및 이동 명령

    if self._move:

        newpos = self.rect.move(self._move, 0) #새 위치 계산 및 경계 확인

        if self._area_contains(newpos):#경계 내 정상 이동 처리

            self.rect = newpos
        else: #경계 도달 시 정지 및 위치 보정

            while self._move != 0:
                if self._move < 0:
                    self._move += 1
                else:
                    self._move -= 1

                newpos = self.rect.move(self._move, 0)
                if self._area_contains(newpos):
                    self.rect = newpos
                    break
```

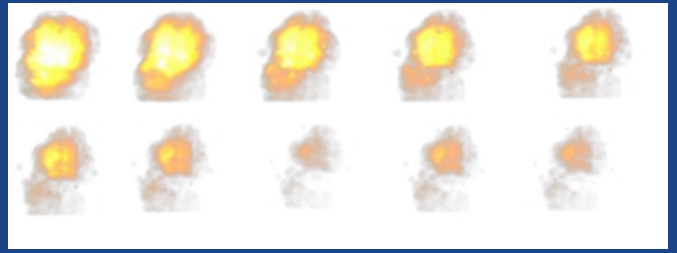
```python
# 게임 루프 내에서 이벤트 처리
for event in pygame.event.get():
    if event.type == pygame.KEYDOWN:
        # ⬅️ 왼쪽 화살표 키가 눌렸을 때
        if event.key == pygame.K_LEFT:
            my_paddle.move_left()
        # ➡️ 오른쪽 화살표 키가 눌렸을 때
        elif event.key == pygame.K_RIGHT:
            my_paddle.move_right()

    if event.type == pygame.KEYUP:
        # 키에서 손을 떼었을 때
        if event.key == pygame.K_LEFT or event.key == pygame.K_RIGHT:
            my_paddle.stop()
```

코드 구성 -2 적

```
class Enemy(pygame.sprite.Sprite):
    def __init__(self, enemy_type, paddle, on_paddle_collide,
                 collidable_sprites, on_destroyed):

        super().__init__()
        self._enemies.add(self)
        self._paddle = paddle
        self._on_paddle_collide = on_paddle_collide
        self._on_destroyed = on_destroyed # 적이 파괴되었을 때 호출되는 콜백 함수
        self._on_destroyed_called = False

        screen = pygame.display.get_surface()
        self._area = screen.get_rect()

        self._animation, width, height = self._load_animation_sequence(
            enemy_type.value)
s.
        self.rect = pygame.Rect(self._area.center, (width, height))
        self.image = None
        self._explode_animation = None #  폭발 애니메이션 변수 초기화
        self._direction = START_DIRECTION # 초기 이동 방향 설정
        self._duration = START_DURATION
```

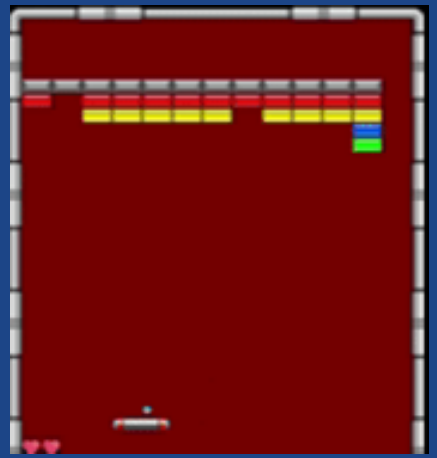
코드 구성 -2.1 적 파괴

```
# 적이 공이나 패들에 맞아 파괴될 때 폭발 애니메이션을 처리
def _explode(self):
    try: # 폭발 애니메이션 시퀀스(_explode_animation)의 프레임을 모두 사용하면 발생
        if self._update_count % 2 == 0:
            rect = self.rect
            self.image, self.rect = next(self._explode_animation)
            self.rect.center = rect.center

    except StopIteration:
        self._explode_animation = None
        #콜백이 None이 아닐 때만 호출하도록 변경
        if self._on_destroyed:
            self._on_destroyed(self) # 콜백을 호출하여 적을 게임에서 최종적으로 제거
```

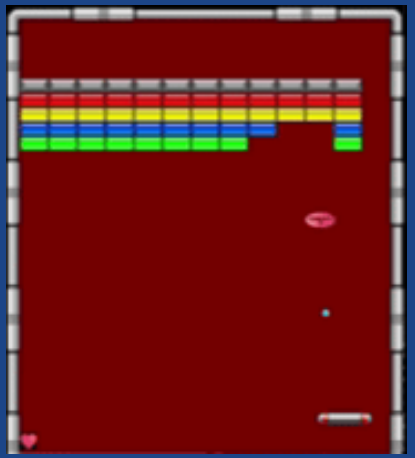


코드 구성 -3 아이템: 생명 추가, 공 속도 감소

```python
class ExtraLifePowerUp(PowerUp):

    def __init__(self, game, brick):
        super().__init__(game, brick, 'powerup_life')

    def _activate(self):

    def deactivate(self):
        pass

class SlowBallPowerUp(PowerUp):

    _SLOW_BALL_SPEED = 6  # Pixels per frame.

    def __init__(self, game, brick):
        super().__init__(game, brick, 'powerup_slow')
        self._orig_speed = None

    def _activate(self):
        self._orig_speed = self.game.ball.base_speed
        for ball in self.game.balls:
            ball.speed = self._SLOW_BALL_SPEED
            ball.base_speed = self._SLOW_BALL_SPEED

    def deactivate(self):
        for ball in self.game.balls:
            ball.speed = self._orig_speed
            ball.base_speed = self._orig_speed
```



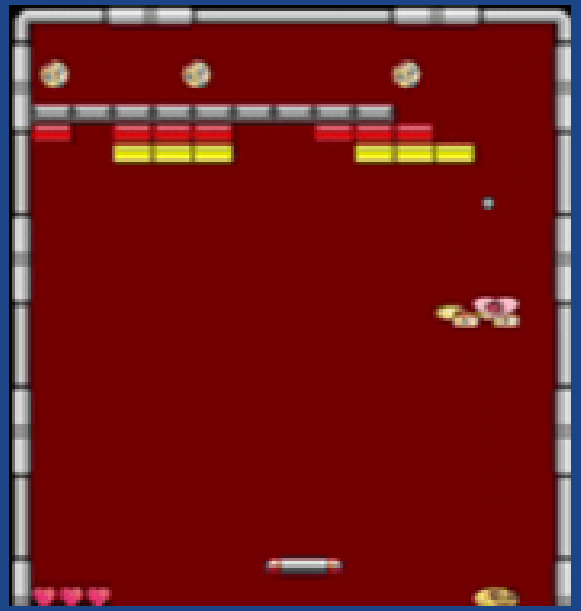
생명 추가



공 속도 감소

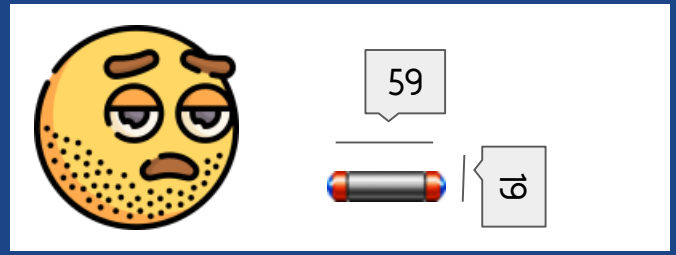
코드 구성 -3 아이템:공 복제

```python
class DuplicatePowerUp(PowerUp):      # 복제 맞음

    def __init__(self, game, brick):
        super().__init__(game, brick, 'powerup_duplicate')

    def _activate(self):

        split_angle = 0.4 #복제가 일어나는곳

        for ball in list(self.game.balls):

            start_pos = ball.rect.center

            start_angle = ball.angle + split_angle #첫 번째 복제공 생성
            if start_angle > 2 * math.pi:
                start_angle -= 2 * math.pi

            ball1 = ball.clone(start_pos=start_pos,
                               start_angle=start_angle)

            start_angle = abs(ball.angle - split_angle) #2번

            ball2 = ball.clone(start_pos=start_pos,
                               start_angle=start_angle)

            self.game.balls.append(ball1) #게임에 공추가
            self.game.balls.append(ball2)

            self.game.sprites.append(ball1)
            self.game.sprites.append(ball2)
```



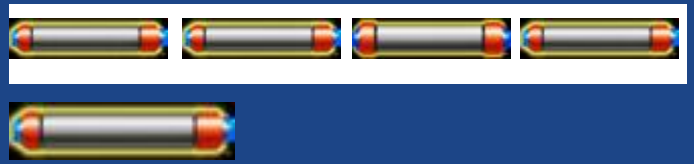
코드 구성 -4 방해 효과

```python
class ReducePowerUp(PowerUp):        # reduce
    def __init__(self, game, brick):
        super().__init__(game, brick, 'powerup_reduce')  # powerup_reduce_1.png 등과 연결

    def _activate(self):
        self.game.paddle.transition(NarrowState(self.game.paddle))
        # ⚠ 공 속도는 그대로 두기 (요청 반영)

    def deactivate(self):
        self.game.paddle.transition(NormalState(self.game.paddle))
        # ⚠ 공 속도 변화 없음

    def _can_activate(self):
        can_activate = super()._can_activate()
        if can_activate:
            # 이미 줄어든 상태면 또 줄이지 않기
            can_activate = not isinstance(self.game.active_powerup,
                                          self.__class__)
```



코드 구성 -5 필살기

```
from pygame.sprite import Sprite # ♦ 필살기 아이템 생성을 위해 pygame.Sprite 임포트
        # ♦ 필살기 관련 변수
        self.special_ready = False  # 필살기 사용 가능 상태
        self.special_used = False   # 현재 라운드에서 필살기 사용 여부
        self.special_item = None    # 화면에 존재하는 필살기 아이템 Sprite
        self.special_brick = None   # 💡 [추가] 필살기를 가지고 있는 블록
        # ♦ 필살기 아이템 낙하 및 획득 처리
        if self.special_item and self.special_item.visible:
                # 📢 [추가] 필살기 아이템 획득 시 패들 이미지 변경
                self.paddle.activate_special_image()
    # ♦ 필살기 발동 메서드
    def activate_special(self):
        """필살기를 발동하고 적들을 폭발시킵니다."""
        # 획득 상태이고, 아직 사용하지 않았다면 발동
        if not self.special_ready or self.special_used:
            return
        # 📢 [추가] 필살기 사용 시 패들 이미지 원래대로 복구
        self.paddle.deactivate_special_image()

        # 현재 화면에 보이는 모든 적을 폭파
        for enemy in self.enemies:
            if enemy.visible:
                enemy.explode()
                self.score += 500 # 점수 추가
```



06
게임 시연

Arkanoid

1UP
0

HIGH SCORE
29240

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

Based on original Arkanoid game
by Taito Corporation 1986

07

Q & A

Thank you

출처

- [심근경색증 바로알기]보건복지부 지정 인하대병원 인천권역심뇌혈관질환센터
- 질병관리청(2024) [심뇌혈관질환 발생통계] 심근경색증 (I21~I23) 상병코드로 입원한 환자(국민건강보험빅데이터)
- [서울 아산병원 뉴스룸] 심뇌혈관질환의 씨앗 '고지혈증' 제대로 알고 관리하자
- 아이콘 출처: flaticon.com
- 워크플로우 그린 사이트: https://www.drawio.com/
- 활용 방식 이미지 출처: https://pixabay.com/
- 폰트 : PF스타더스트