

# OSSProj 주간보고서

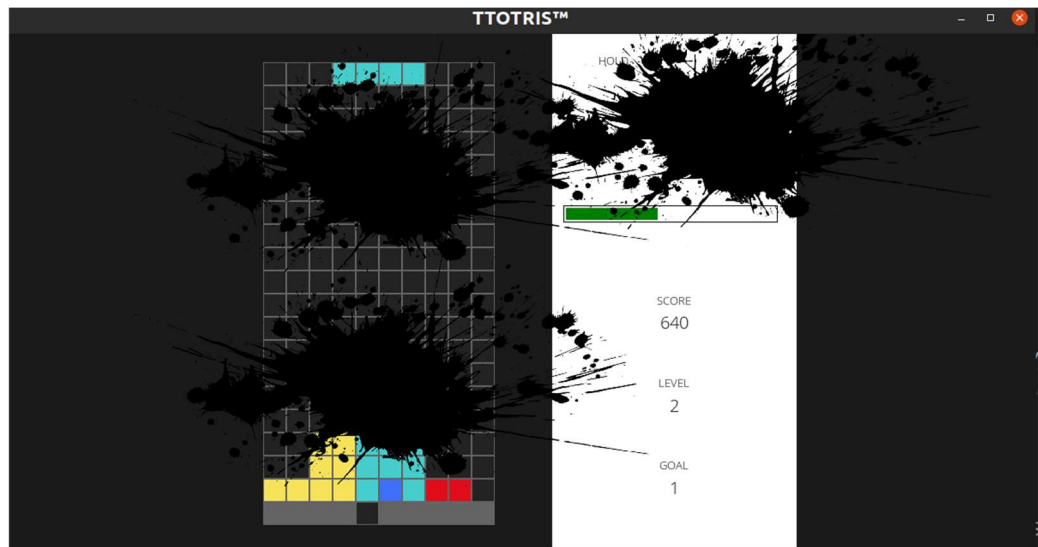
이번 주 주요 내용	<ul style="list-style-type: none"> <li>하드모드 피버 타이머 구현 완료</li> <li>아이템 모드 구현 진행 중</li> </ul>
---------------	---

교과목명	오픈소스 소프트웨어 프로젝트			담당교수	김동호 교수님
과제명	오픈소스를 활용한 테트리스 기능 개발			팀명	DoitDoit
회의일자	11/17 11/19	시간	15:00 - 17:00 19:00 - 21:00	장소	대면(아리수, 카페)

내용

## 1. 하드모드 구현 완료

- 1) 지난주 회의록에서 작성했던 바와 같이 장애물이 나오는 조건 완성
- 2) 500~1000, 2000~2500, 3500~4000, 단위로 장애물 등장
- 3) 난이도를 높이기 위해 다음 블록을 보여주는 화면 또한 방해



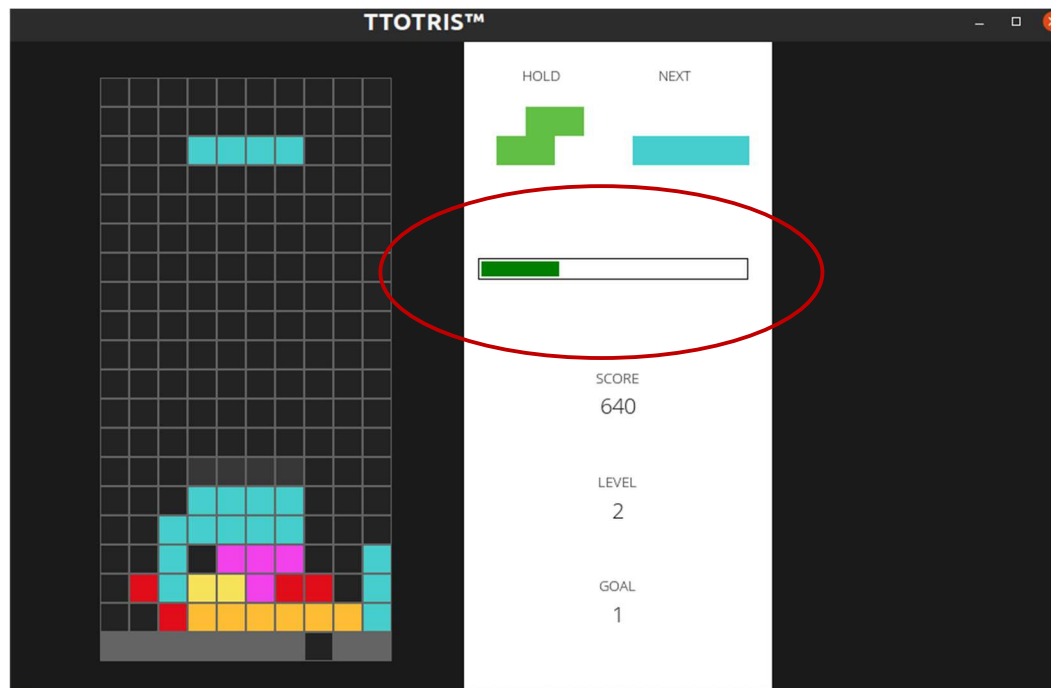
: 블록이 내려오는 부분과 다음 블록을 보여주는 부분에 장애물이 등장하는 화면

- 4) 하드모드인 만큼 난이도를 높일 수 있는 추가 기능 고려

## 2. 피버 타이머 위치 수정

- 1) 피버 타임이 얼마나 남았는지 유저가 직관적으로 알 수 있게끔 타이머를 구현하고 적절한 곳에 타이머 바를 배치
- 2) 처음 구현했던 피버모드는 게임이 시작하는 순간 시간을 측정하기 때문에 블록을 깨면 너무 짧은 피버 시간이 주어지거나 아예 피버 시간이 주어지지 않는 오류 발견  
→ 스위치를 만들어 피버 조건이 갖춰질 때부터 시간을 측정하도록 재설계하여 해결
- 3) score가 올라갈수록 많은 피버 타임을 주던 것을 score가 올라갈수록 적은 피버 타임을 주

도록 변경



: 남은 피버 시간을 알려주는 타이머 바를 추가한 모습

### 3. 구조화 작업

```
class ui_variables:
    pygame.init()
    ## 게임 보드 관련
    SCREEN_WIDTH = 1200
    SCREEN_HEIGHT = 600
    DEFAULT_WIDTH = 10
    DEFAULT_HEIGHT = 20
    block_size = 17 # Height, width of single block
    width = DEFAULT_WIDTH # Board width
    height = DEFAULT_HEIGHT # Board height
    min_width = 700
    min_height = 350
    c = 0
    mino_size = 4
    mino_turn = 4
    # Fonts
    font_path = "./assets/fonts/OpenSans-Light.ttf"
    font_path_b = "./assets/fonts/OpenSans-Bold.ttf"
    font_path_i = "./assets/fonts/Inconsolata/Inconsolata.otf"
```

\* 기존 모든 변수와 상수가 Pintris.py에 있던 것을 기능에 따라 알맞은 클래스로 분류 작업 진행 중

### 4. 아이템 모드 구현 진행

1) 아이템 모드의 아이템 보드 구현 완료

- 아이템 인벤토리를 표시하기 위해 사이드바의 전체적인 레이아웃을 변경(아래 그림 참고)

2) 구체적인 아이템 사용 방법 고안 및 개략적인 아이템 함수 구현 완료

- 아이템 모드: 500점 단위로 점수에 도달하면 아이템을 랜덤으로 부여

→ 키를 눌러 인벤토리에 저장된 아이템 사용 가능(제안서에서의 아이템 = 맨 밑줄 삭제, 전체 블록 리셋, 현재 블록 중심으로 3x3 크기 블록 삭제, 10초간 점수 2배)

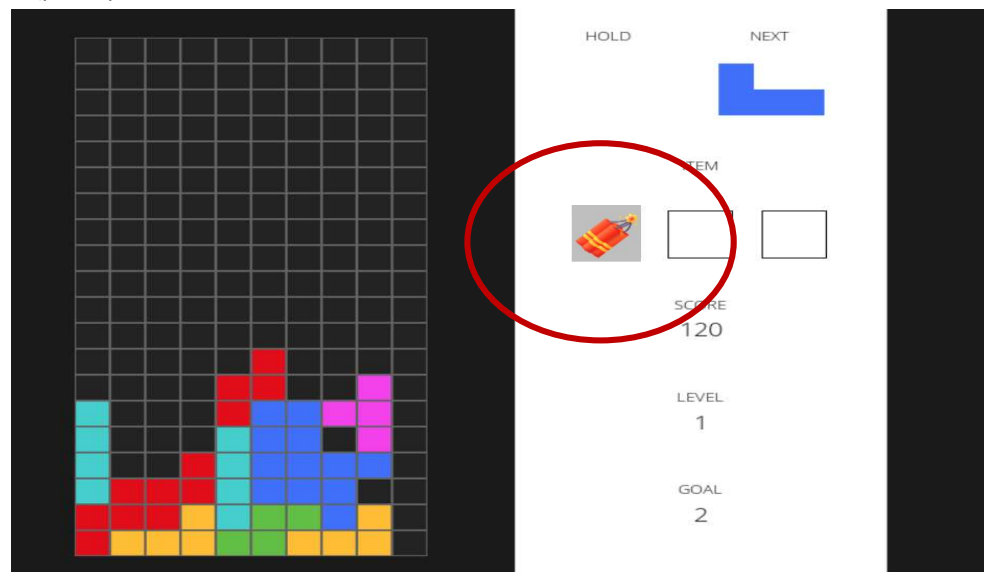
- 아이템 중 '현재 블록 중심으로 3x3크기의 블록 삭제'에서, 현재 블록의 기준이 명확하지 않다는 문제 발생

→ 아이템 사용시, 1x1 크기의 아이템 블록이 등장하고 사용자가 원하는 위치에 이동시켜 삭제가 가능하도록 구현하기로 결정

- 같은 방식으로 가로줄 삭제, 세로줄 삭제가 가능한 아이템을 추가하기로 결정

- 각 아이템의 효과 함수는 구현 완료, 게임 루프 상에서 실행 가능하도록 시도 중

<예상 화면>



: 인벤토리에 아이템이 저장된 모습



: 키를 눌러 아이템을 사용할 경우

## 5. soft drop 키 조작 활성화

<기존 코드>

- 키 조작 부분이 아닌 게임 초반, 아래쪽 화살표가 길게 눌리면 블록 하강 속도가 빨라지도록 선언 → single 모드들에서는 문제가 없었으나, pvp 모드에서 각 플레이어가 별로 하강 속도 조절이 불가능하다는 문제점 발견

```
elif event.type == USEREVENT:
    # Set speed
    if not game_over:
        keys_pressed = pygame.key.get_pressed()
        if keys_pressed[K_DOWN]:
            pygame.time.set_timer(pygame.USEREVENT, framerate * 1)
        else:
            pygame.time.set_timer(pygame.USEREVENT, framerate * 5)
```

<해결 후>

- 키에 따라 화면 출력 속도를 조절하는 것이 아닌, 아래 화살표 또는 S키를 눌렀을 때 블록의 높이에 해당하는 값을 하나씩 증가시켜 하강 속도가 빨라진 효과를 냄(soft drop 기능)



: 2p(화면 상 오른쪽)만 아래 화살표로 soft drop 했을 때의 게임 화면(2p의 블록만 더 빨리 내려와 있음)