OSSProj 주간보고서

이번 주 주요 내용	• 지적사항 수정 • AWS 완성단계 돌입

교과목명	오픈소스 소프트웨어 프로젝트			**담당교수**	김동호 교수님
과제명	오픈소스를 활용한 테트리스 기능 개발			**팀명**	DoitDoit
회의일자	11/4	**시간**	20:00 - 21:00	**장소**	Webex

내용

1. 수정요구사항

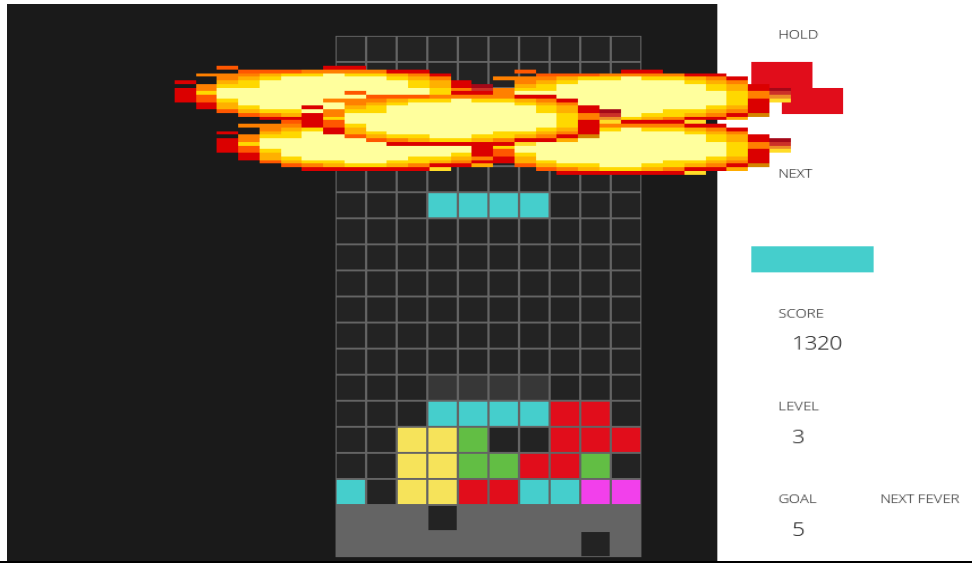
1) 9주차 발표에서 수정사항으로 요구하셨던 사항 반영

2) 최대한 변수를 활용하여 게임루프상에서 수정하는 것이 아닌, 코드 최상단 변수에서 수정하는 방식으로 변경

```
FEVERTIMER = [0,1,2,3,4]
FEVERSCOREBOARD = [0,1500,5000,15000,30000]
FEVERGOAL = 4
```

```
if FEVERSCOREBOARD[1] > score >=FEVERSCOREBOARD[0]:
    ADD = FEVERTIMER[0]
if FEVERSCOREBOARD[2] > score >= FEVERSCOREBOARD[1]:
    ADD = FEVERTIMER[1]
if  FEVERSCOREBOARD[3] > score >= FEVERSCOREBOARD[2]:
    ADD = FEVERTIMER[2]
if  FEVERSCOREBOARD[4] > score >= FEVERSCOREBOARD[3]:
    ADD = FEVERTIMER[3]
if  score >= FEVERSCOREBOARD[4]:
```

* 피버타임 시작 시의 게임 화면



2. AWS 데이터 삽입 함수

1) 모드별 랭킹을 기록하기 위한 함수 설계
2) isthereID란 함수를 이용하여 데이터베이스의 id 조회 가능

```
def isthereID(ID, table):
    curs = tetris.cursor()
    sql = "SELECT * FROM {} WHERE id=%s".format(table)
    curs.execute(sql, ID)
    data = curs.fetchone()
    curs.close()
    if data:
        return False
    else:
        return True
```

3) 이후 isthereID함수가 true값을 리턴할 시 Insert문을, false일 시 Update문을 실행

.

```
if difficulty == 0: ## normal
    name2 = chr(name[0]) + chr(name[1]) + chr(name[2])
    if isthereID(name2,"NORMAL"):
        cursor = tetris.cursor()
        sql = 'INSERT INTO NORMAL (id, score) VALUES (%s,%s)'
        cursor.execute(sql, (name2, score))
        tetris.commit()
        cursor.close() ## tetris db insert
    else :
        cursor = tetris.cursor()
        sql = "select score from NORMAL where id =%s"
        cursor.execute(sql, name2)
        result = cursor.fetchone()
        if result[0] < score:
            sql = "Update NORMAL set score = %s where id =%s"
            cursor.execute(sql, (score,name2))
            tetris.commit()
            cursor.close() ## tetris db insert
        else: pass
```

4) 위 함수를 게임루프상의 코드로 넣지 않고, 메소드화 시켜 전체적인 코드의 수 감소

* 변경전 코드

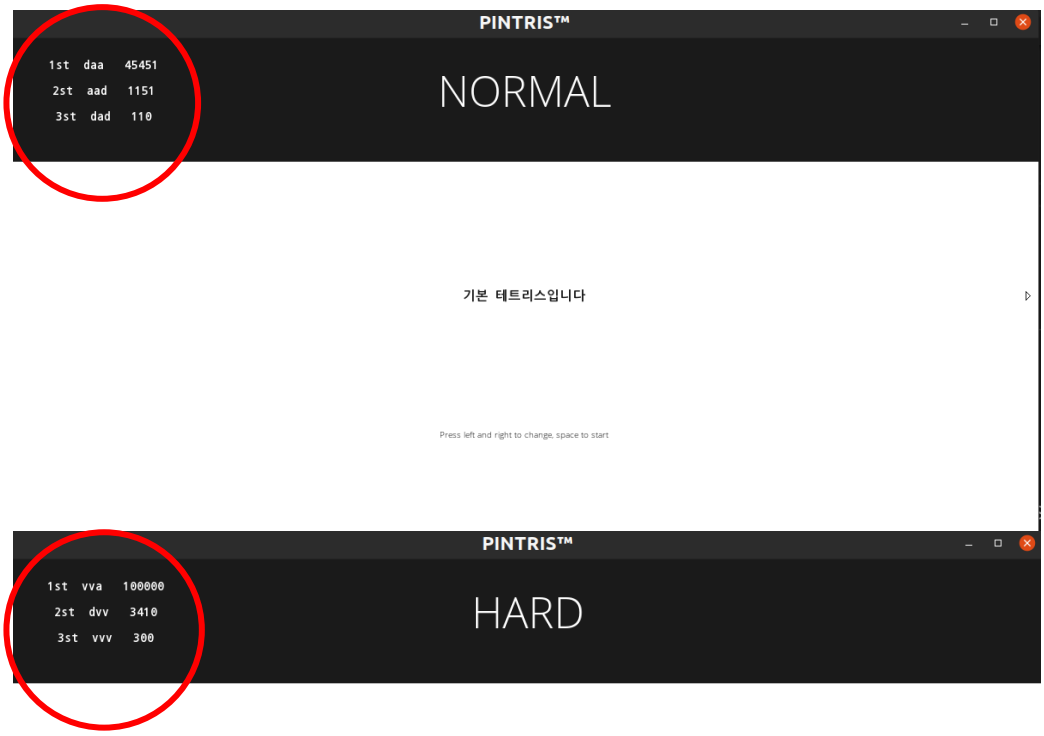
```
if difficulty == 1: ## hard
    cursor = tetris.cursor()
    name2 = chr(name[0]) + chr(name[1]) + chr(name[2])
    sql = 'INSERT INTO Hard (id, score) VALUES (%s,%s)'
    cursor.execute(sql, (name2, score))
    tetris.commit()
    cursor.close()
if difficulty == 2: ## ITem
    cursor = tetris.cursor()
    name2 = chr(name[0]) + chr(name[1]) + chr(name[2])
    sql = 'INSERT INTO Item (id, score) VALUES (%s,%s)'
    cursor.execute(sql, (name2, score))
    tetris.commit()
    cursor.close()
if difficulty == 4: ## reverse
    cursor = tetris.cursor()
    name2 = chr(name[0]) + chr(name[1]) + chr(name[2])
    sql = 'INSERT INTO Reverse (id, score) VALUES (%s,%s)'
    cursor.execute(sql, (name2, score))
    tetris.commit()
    cursor.close()
```

* 변경 후 코드

```
ef istheresaved(gamemode):
  name2 = chr(name[0]) + chr(name[1]) + chr(name[2])
  if isthereID(name2,gamemode):
      cursor = tetris.cursor()
      sql = "INSERT INTO {} (id, score) VALUES (%s,%s)".format(gamemode)
      cursor.execute(sql, (name2, score))
      tetris.commit()
      cursor.close() ## tetris db insert
  else :
      cursor = tetris.cursor()
      sql = "select score from {} where id =%s".format(gamemode)
      cursor.execute(sql, name2)
      result = cursor.fetchone()
      if result[0] < score:
          sql = "Update {} set score = %s where id =%s".format(gamemode)
          cursor.execute(sql, (score,name2))
          tetris.commit()
          cursor.close() ## tetris db insert
      else: pass
```

```
## 여기서부터 기록 저장
if DIFFICULTY_NAMES[current_selected] == "NORMAL": ## normal
    istheresaved(DIFFICULTY_NAMES[current_selected])
if DIFFICULTY_NAMES[current_selected] == "Item": ## normal
    istheresaved(DIFFICULTY_NAMES[current_selected])
if DIFFICULTY_NAMES[current_selected] == "Hard": ## normal
    istheresaved(DIFFICULTY_NAMES[current_selected])
if DIFFICULTY_NAMES[current_selected] == "Reverse": ## norma
    istheresaved(DIFFICULTY_NAMES[current_selected])
```

* 게임 시작 페이지 좌측 상단에 랭킹 표시

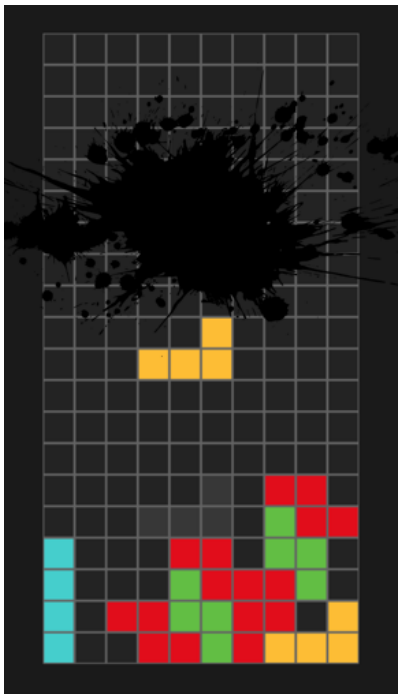


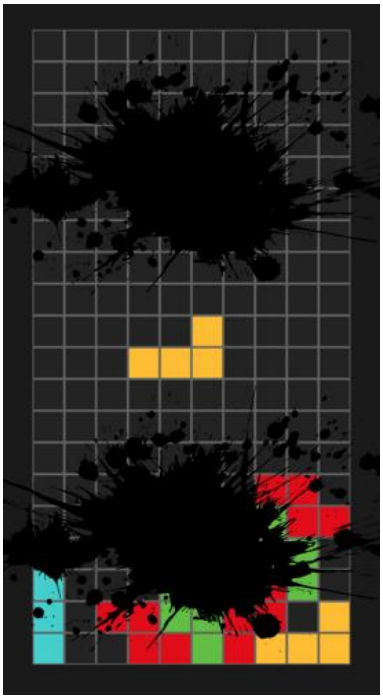
5) 발견된 이슈: 데이터베이스의 인스턴스가 3개가 없을 경우 에러가 뜨는 문제 수정 필요성

3. Hard mode 시작

<구체적 구현 틀 형성>

1) 게임 진행 도중 게임 화면을 가릴 png를 이용한 장애물 등장 예상 화면.





: 하나에서 두 개 정도의 장애물로 화면을 가려 게임을 방해합니다.

2) 장애물 등장하면 이미지가 깜빡거리도록 구현할 계획

```
# fever time시 이미지 깜빡거리게
if blink:
    screen.blit(pygame.transform.scale(ui_variables.fever_image,
                                       (int(SCREEN_WIDTH * 0.5), int(SCREEN_HEIGHT * 0.2))),
                (SCREEN_WIDTH * 0.1, SCREEN_HEIGHT * 0.1))
    blink = False
else:
    blink = True

dt = t1 - t0

if dt >= 27:
    mino = next_mino
    next_mino = randint(1, 7)
    t0 = t1
    comboCounter =0
```

: 이 기능은 앞서 완성한 fever time에서 이미지 파일을 깜빡이도록 구현한 코드를 사용하여 hard mode 함수 내에 작성합니다. 기존 fever time에서의 이미지는 짧은 시간 내에 사라지도록 되어 있는데, 코드의 숫자를 조정하여 게임에 방해가 될 정도의 시간 동안 이미지가 나타나도록 수정할 것입니다.

4. Git 사용법

Github에서 merge전 open request를 각 로컬로 가져오는 방법에 대해 토의하였습니다.
<참고한 링크>
github PR(pull request) 을 merge 하지 않고 local 로 받기 (lesstif.com)

5. 다음주 진행 예정 사항

1. AWS 인스턴스 관련 오류 개선
2. Hard Mode 장애물 구현
3. 속도 조절 구현 시작 – 전체적인 페이지 구조 조정