

OSSP 주간보고서

이번 주 주요 내용	Base Source 코드 분석 및 버그 픽스 아이디어 제안				
교과목명	오픈소스 소프트웨어 프로그래밍		담당교수	김동호 교수님	
과제명	Base Source 코드 분석 및 버그 픽스 아이디어 제안		팀명	무야호	
일자	2021.05.05	시간	15:00~19:00	장소	충무로 카페

## 회의 주요 내용

1. Base Source 코드 숫자 처리
2. LEVEL과 무관한 비정상적 낙하 속도 코드 분석 및 FIX
3. 그림자 버그 FIX

### 1. Base Source 코드 숫자 처리

기존 Base Source 코드 분석 도중 무분별하게 변수에 저장되는 값 들이 해당 파일에 숫자로 처리되어 있는 문제 발견

프로젝트 개발 진행에 있어 새로운 변수 관련 파일(Variable.py)을 통해 한곳에 모아 관리하는 것이 효율적일 것으로 판단

#### 변수 파일 따로 생성

#### 내용

##### Board.py

```
self.width = 10  
self.height = 20  
self.block_size = 25
```

```
self.score = 0  
self.level = 1  
self.goal = 5  
self.skill = 0
```

##### Variable.py

```
class Size:  
    field_width = 10 # 맵의 좌에서 우로 사이즈  
    field_height = 20 # 맵 위에서 아래로 사이즈  
    block_size = 25
```

```
class Set:  
    init_score = 0 # 초기 점수 세팅  
    init_level = 1 # 초기 레벨  
    init_goal = 5 # 레벨업을 하기 위한 조건  
    init_skill = 0 # q스킬 게이지  
    empty_board = 0 # 블록이 없는 빈 상태  
    create_location_x = 3 # 블록 생성 초기 x위치  
    create_location_y = 0 # 블록 생성 초기 y위치
```

##### Board.py

```
+ from Variable import *
```

```
self.width = Size.field_width  
self.height = Size.field_height  
self.block_size = Size.block_size
```

```
self.score = Set.init_score  
self.level = Set.init_level  
self.goal = Set.init_goal  
self.skill = Set.init_skill
```

위와 같이 기존 **Board.py**의 변수에 사용되었던 숫자들을 Variable.py에서 각 변수의 특성에 맞게 **Size 클래스** 또는 **Set 클래스** 등으로 묶어서 변수 명 설정

Board.py를 시작으로 Tetris.py, Piece.py에 있는 숫자들도 처리할 예정

## 2. LEVEL과 무관한 비정상적 낙하속도 코드 분석 및 FIX

기존 **Base Source** 코드에서는 해당 **LEVEL**에서 요구하는 목표(지운 줄 수)를 달성할 시 다음 **LEVEL**로 올라가고 레벨이 올라가면 블록이 내려오는 속도가 빨라지는 시스템

레벨이 올라감에 따라 `pygame.time.set_timer()`에 의해 속도 증가

```
def delete_lines(self):
    remove = [y for y, row in enumerate(self.board) if all(row)]
    for y in remove:
        line_sound = pygame.mixer.Sound("assets/sounds/Line_Clear.wav")
        line_sound.play()
        self.delete_line(y)
        self.score += 10 * self.level
        self.goal -= 1
    if self.goal == 0:
        if self.level < 10:
            self.level += 1
            self.goal = 5 * self.level
        else:
            self.goal = '-'
    if self.level <= 9:
        pygame.time.set_timer(pygame.USEREVENT, (500 - 50 * (self.level-1)))
    else:
        pygame.time.set_timer(pygame.USEREVENT, 100)
```

하지만 게임 진행시 LEVEL과 상관없이 버벅거리며 블록이 빨리 내려오는 현상 발생

이 조건문이 `delete_lines()`함수에 포함되어 있기 때문에, 한 줄을 지워야만 조건문이 실행되어 문제가 발생

**따라서 새로운 블록마다 조건문을 실행하는 `absorb_piece()` 함수에 해당 조건문을 추가하였고 문제 해결**

추가 버그 발견

게임을 restart 할 시 실행파일 `run()`의 초기 값을 받아서 실행하는 것이 아닌 마지막 속도로 실행 후, 다음 속도부터 정상 속도로 실행되는 문제 발견

추후에 menu를 생성 후 restart 코드를 수정해야 할 것으로 예상됨

### 3. 그림자 버그 FIX

기존 **draw\_blocks()** 함수 내에 그림자 기능이 같이 구현 되어있는 것에서 문제 발생

```
def draw_shadow(self, array2d, dx, dy): #그림자 기능 함수 추가
    for y, row in enumerate(array2d):
        y += dy
        if y >= 2 and y < self.height:
            for x, block in enumerate(row):
                x += dx
                if block:
                    tmp = 1
                    while self.can_move_piece(0, tmp):
                        tmp += 1
                    x_s, y_s = self.pos_to_pixel(x, y + tmp - 1)

                    pygame.draw.rect(self.screen, self.piece.T_COLOR[7],
                                     (x_s, y_s, self.block_size, self.block_size))
                    pygame.draw.rect(self.screen, BLACK,
                                     (x_s, y_s, self.block_size, self.block_size), 1)
```

따라서 해당 **draw\_shadow()** 함수를 새로 정의하여 그림자 오류 해결

하지만 블록 생성하는 시점에서 블록이 보이지 않는 부분은 그림자로 표시되지 않는 문제 발생

이는 그림자를 draw 하는 조건에 문제가 있음을 파악

그림자 생성 조건에 **y >= 2 and y < self.height**에서 y>=2인 시점부터 그림자를 그리는 것이 원인

**따라서 해당 블록 생성 시점도 그림자가 그려지도록 조건을 아래와 같이 변경**

```
def draw_shadow(self, array2d, dx, dy): #그림자 기능 함수 추가
    for y, row in enumerate(array2d):
        y += dy
        if y >= 0 and y < self.height:
            for x, block in enumerate(row):
```

