**[5조 CHUPARKIM] OSSP 11주차 주간 보고서**

* + 1. 2021.05.10

**1.프로젝트 개요**

|  |  |
| --- | --- |
| 팀 명 | CHUPARKIM |
| 팀 원 | 김정현, 박상준, 추예진 |
| 주 제 | CPK\_PYTRIS |

**2.팀 활동내용**

04/26(월)

제안서 발표 후 교수님 피드백 받음. ‘무엇을 하려는 지 정확히 모르겠다’, ‘완성본을 실행할 때 프로그램이 버벅거릴 것 같다’, ‘할 일의 양이 적어 보인다’,

04/28(수) -회의

조교님 피드백 받음. ‘게임 모드에 대한 설명이 청자에게 와닿도록 설명되지 않았다’. ‘구체적으로 완성된 모습이 어떻게 될 지 모호하다’. ‘왜 완성되었을 때 버벅거릴 것 같은 지’에 대한 피드백은 시간관계상 받지 못함.

그동안 받은 피드백을 바탕으로 수업 종료 후 바로 회의진행. 프로젝트 기획을 위한 회의 단계에서 새로운 mode를 구현하는데 시간이 상당부분 소요될 것이라고 예상하였다. 그러나 실질적으로는 구현하기 쉬울 것이라는 피드백을 수용하여 일단 주말을 이용해 빠른 시간내에 가장 고민할 요소가 많은 rotate mode를 구현하는 것을 목표로 함.

Git Hub에 있는 wiki와 readme를 활용해서 정보를 채워 넣을 것.

05/02(일)-회의

팀원 모두 주말동안 rotate mode를 구현하는데 실패했음. pygame에서 모듈을 통해 그려진 image를 회전하는 방식으로 하려했음. 그러나 테트리스 block이 떨어지는 board를 함수와 image를 그리는 함수가 연결되어 있어, 단순히 image만 회전하기 힘들고 block이 처리되는 방식을 완전히 바꿔줘야 할 것을 예상됨. 결국 90도 180도 270도 마다 다른 객체를 생성하는 해결책이 가장 직관적이라고 판단함. pygame의 game screen 좌표를 일일이 계산하여 되기 때문에 상당시간(2주정도)이 소요될 것으로 예상됨. 그러나 기존 방향대로 프로젝트를 진행할 시 수업에서 받은 피드백들을 프로젝트에 반영하지 못하고 계속 진행하는 것인 된다고 판단됨. 회의 분위기가 암울했고, 완전히 프로젝트 방향을 다시 수정하는 것에 대해서 거론되었음. 본 팀이 기존 테트리스 오픈소스의 문제점을 게임성에 집중하여 ‘좋은 게임’으로 수정하겠다고 설정한 것이 수업의 목적과 부합하지 않을 수 있다고 결론을 냄. OSSP수업은 좋은 게임을 만드는 것이기 보다는 오픈소스를 가지고 거기에 여러 기능들을 더해 새로운 결과물을 내는 것이라고 판단되기 때문임.

본 팀은 게임성 향상을 위해 mode구현에 상당한 시간을 배분하였고, 나머지 다른 부분들에 대해서는 최소한으로 고려했음. 당일 회의 진행의 결과 ‘두서없이 프로젝트 진행을 하는데 급급하기 보다는 돌아오는 월요일(05/03)에 방향수정에 대한 피드백을 받고 다시 프로젝트 방향을 수정 후 본격적으로 착수하는 것이 옳다’에 도달함. 본 팀의 선택지는 다음과 같음.

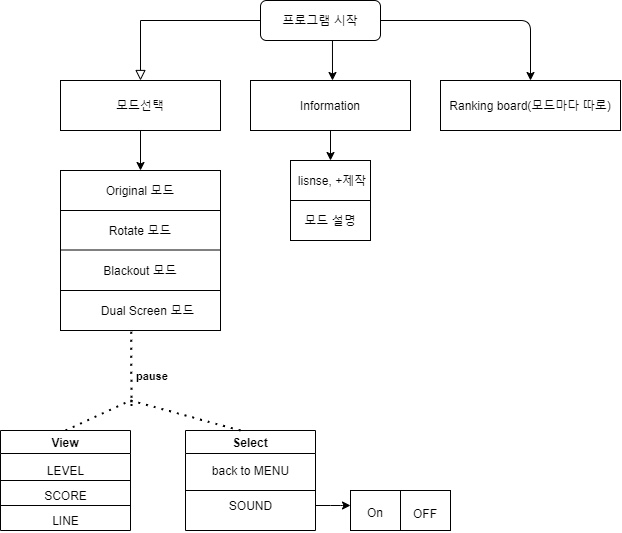
*1. 기존 프로젝트 계획을 그대로 진행*

*2. 모드 구현보다는 서버와 다른 오픈소스를 테트리스에 부착하는 방식으로 프로젝트 계획을 수정.*

05/03(월)-회의

수업시간에 미리 교수 연구실에 가서 05/02 회의 결과에 대해 밝힘. 프로젝트 방향성에 대해 ‘어떻게 해도 좋으나(굳이 데이터베이스나 서버를 활용할 필요는 없음), 다만 팀원들끼리 일정시간(약 90시간)을 소요해 본인들이 납득할 만한 결과물을 내는 게 중요하다’라는 피드백을 받음. 또한 rotate mode 구현이 좌표 값을 일일이 기입하는 방식 보다는 선형대수개념을 통해 계산하는 방식으로 접근하면 효율적일 것이라고 지적 받음.

수업에서 받은 피드백을 바탕으로 당일 19:00에 회의를 진행함. 회의 초반부에는 여전히 방향성에 대해 갈피를 잡지 못하였음. 그러나 교수님의 피드백을 고려한 결과, 본 팀의 결과물의 청사진이 팀원들에게는 이미 회의를 통해, 익숙하게 느껴지지만 프로젝트 외부에서 바라볼 때 모호할 수 있다고 사료되었음. 따라서 프로젝트 방향을 완전히 바꾸기 보다는 좀 더 구체적으로 프로젝트를 디자인하기로 정함. 회의 중반부부터는 완성될 프로그램에 대한 구체적인 flowchart를 만들기 위한 회를 진행함. 이는 아래 그림과 같음



구체적인 형태를 구성하면서 이전 수업에서 받은 ‘무엇을 하려는 지 모르겠다’라는 피드백이 무엇인지 깨닫게 되었음. 프로그램이 어떻게 동작될 것인지에 대한 세부적인 계획이 부족했기 때문임. 구체적으로 정한 부분은 다음과 같음

*-메뉴화면에서 선택은 마우스로 선택하는 방식으로 한다.*

*-기존의 네 개의 테트리스를 동작하는 quarter screen mode는 두 개의 테트리스를 동작하는 dual screen으로 compact down 한다.*

*-기존 프로그램에서 pause시 아무것도 보이지 않았지만, pause시 레벨, 스코어, 라인 값이 보이도록 한다.*

*-pause시 메뉴로 돌아가는 기능과 사운드를 on, off 하는 기능을 추가한다.*

*-mode가 늘어났으므로 구체적으로 어떠한 모드인지 설명해주는 Information menu를 추가한다.*

*-기존프로그램에서 메인 화면의 좌측상단에 3위까지만 보여줬던 스코어 보드를 따로 메뉴에서 스코어 보드를 열람할 수 있는 메뉴를 추가한다. 스코어 보드는 mode별로 따로 표시되게끔 한다. Mode가 4개이기 때문에 스코어 보드도 4개이다.*

*- dual screen 모드의 경우 두개의 테트리스가 별개로 스코어를 갖는다고 표시되지만, 두 테트리스의 점수를 합산하여 하나의 스코어로 계산 되도록한다.*

*-rotate 모드의 구현을 위하여 테트리스를 오른쪽 방향으로만 회전시킬 수 있도록 한다.*

*- original mode와 rotate모드는 배경을 삽입하여, 회전하는 사용자가 방향성을 정확하게 인지할 수 있도록 한다. 이미지는 오래된 TV이미지를 사용하여 TV안에 테트리스가 실행되고 있는 것으로 한다.*

*- Black out모드를 가볍게 구현한 결과 단순히 배경을 칸을 인식할 수 있는 격자를 없앤다고 해도 공간개념에 혼란을 줘서 게임의 재미를 높이려는 목표를 달성하기 어렵다. 따라서 시점이 매번 떨어지는 block에 고정되도록 한다.*

*-게임의 전체적인 theme는 retro로 과거 DOS 컴퓨터 화면의 느낌을 motive로 한다. 이에 맞는 배경, 폰트, 색조, 음악을 구현한다.*

*-mode 구현은 5/16까지 무슨 일이 있어도 완성하도록 한다.*

05/05(수)-회의

15:00에 진행. 프로젝트진행상황을 각자 짧게 브리핑하고, GitHub 협업rule을 재확인: 각자 구현하려는 모드의 py파일을 pull request를 통해 merge 한다.

05/07(금)-회의

14:00에 진행. 가장 문제가 되었던 rotate mode 구현방식을 공유함. 기존의 코드에서는 draw\_board 함수를 통해 왼쪽상단에서부터 오른쪽 하단까지의 순서로 그렸기 때문에 이 부분을 수정하여, 예상했던 것 보다 짧은 코드로 rotate mode를 구현할 수 있게 되었음. 화면이 회전할 때 마다 방향키도 이에 맞게 수정되어야 하기 때문에 이를 수정할 필요가 있음.

GitHub wiki와 readme 활용에 대해 rule을 정함. Wiki는 팀의 활동 내용을 중심으로 작성하도록 하고, readme는 본 팀의 오픈소스를 사용할 사용자 입장에 고려되어 작성하도록 함.

코드가 버벅거릴 거라는 교수님의 피드백에 대해서 진지하게 고민되어야 함. Pygame의 메모리 사용 문제를 먼저 구글링 해보고, 찾지 못할 시, 버벅거림과 튕김 문제를 겪은 일취월장 팀의 ohmytris의 코드를 분석해 볼 필요가 있음.

05/07(일)-회의

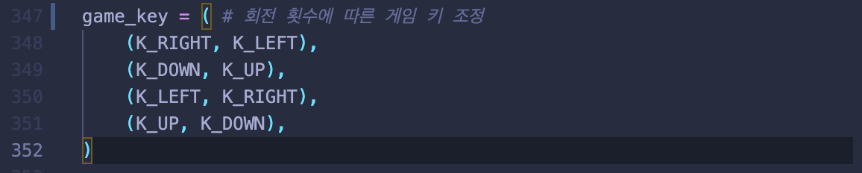
16:00에 진행됨. 프로젝트 진행상황 각자 짧게 브리핑.

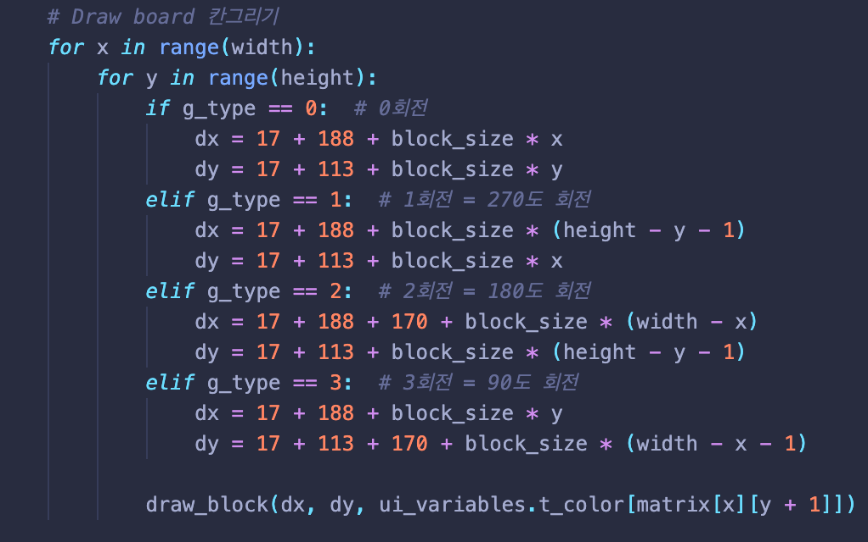
버벅거림과 팅김현상 이슈는 기존에 PYTRIS 코드가 가지고 있던 문제점에 있음. 기능에 대한 부분은 모듈화가 잘 되어있지만, loop이 돌아가는 실질적으로 main 역할을 하는 부분에서 기능별로 모듈화가 되어있지 않음. 무분별하게 남발되는 if문으로 프로그램이 돌아가기 때문에 while문이 계속해서 반복되는 loop이 한번 실행될 때 시간이 오래 걸림. 순차적인 주석 처리 과정을 통해 ohmytris의 코드를 분석한 결과 함수가 콜 되는 부분이 상당부분 엉켜 있는 것 같음. 본 팀의 프로젝트 역시 프로그램을 구성하는데 있어서, main 부분이 길어질 수밖에 없음. 따라서 최대한 기능을 분리해 메인 부분을 module화 하고, 새로 추가할 모드를 구현할 때 최대한 기능들을 모듈화해서 main loop을 간소화해서 개발할 필요가 있음. 인터페이스 구현 단계에서 파이썬 파일을 따로 해서 상수들과, 함수부분, main 부분을 분리할 것도 고려해야 됨.

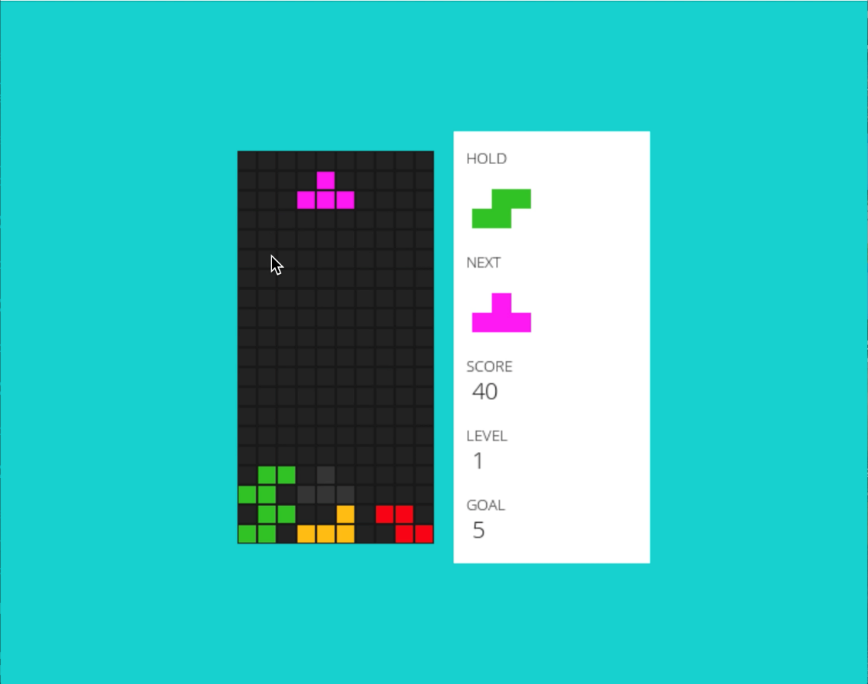
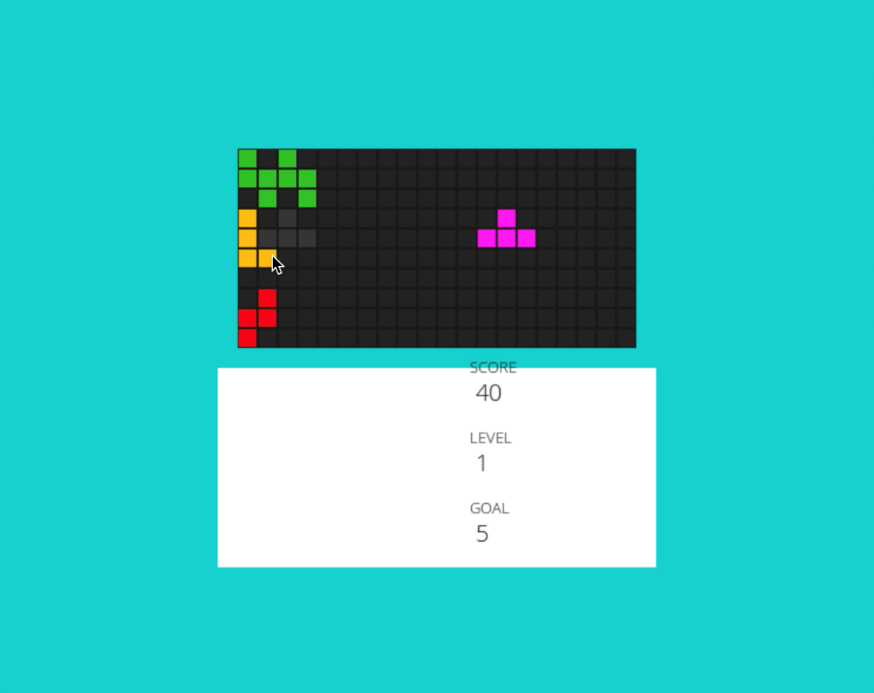
Rotate mode와 blackout mode를 병합하여 새로운 mode 개발도 고려해 볼만함. Lab 형식으로 구현 시도해 볼 것.

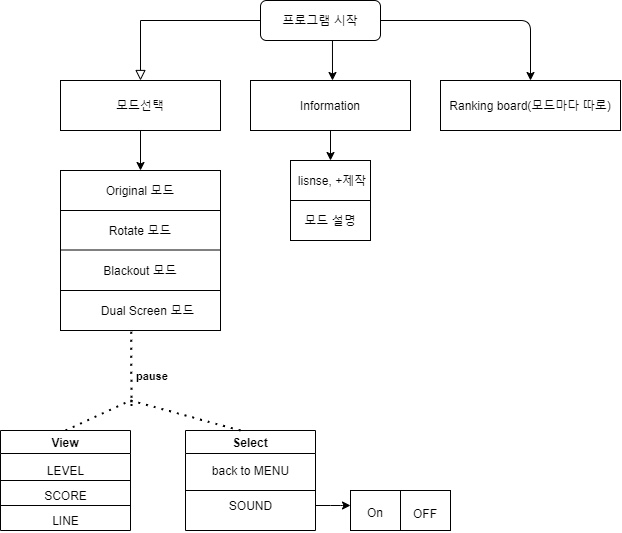
인터페이스 소스에 대한 자료수집이 필요함.

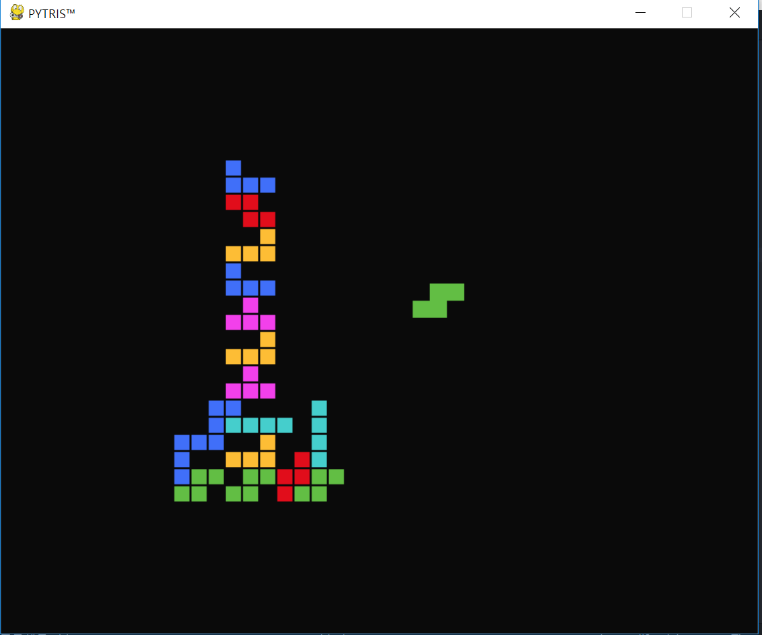
**3.프로젝트 진행상황**

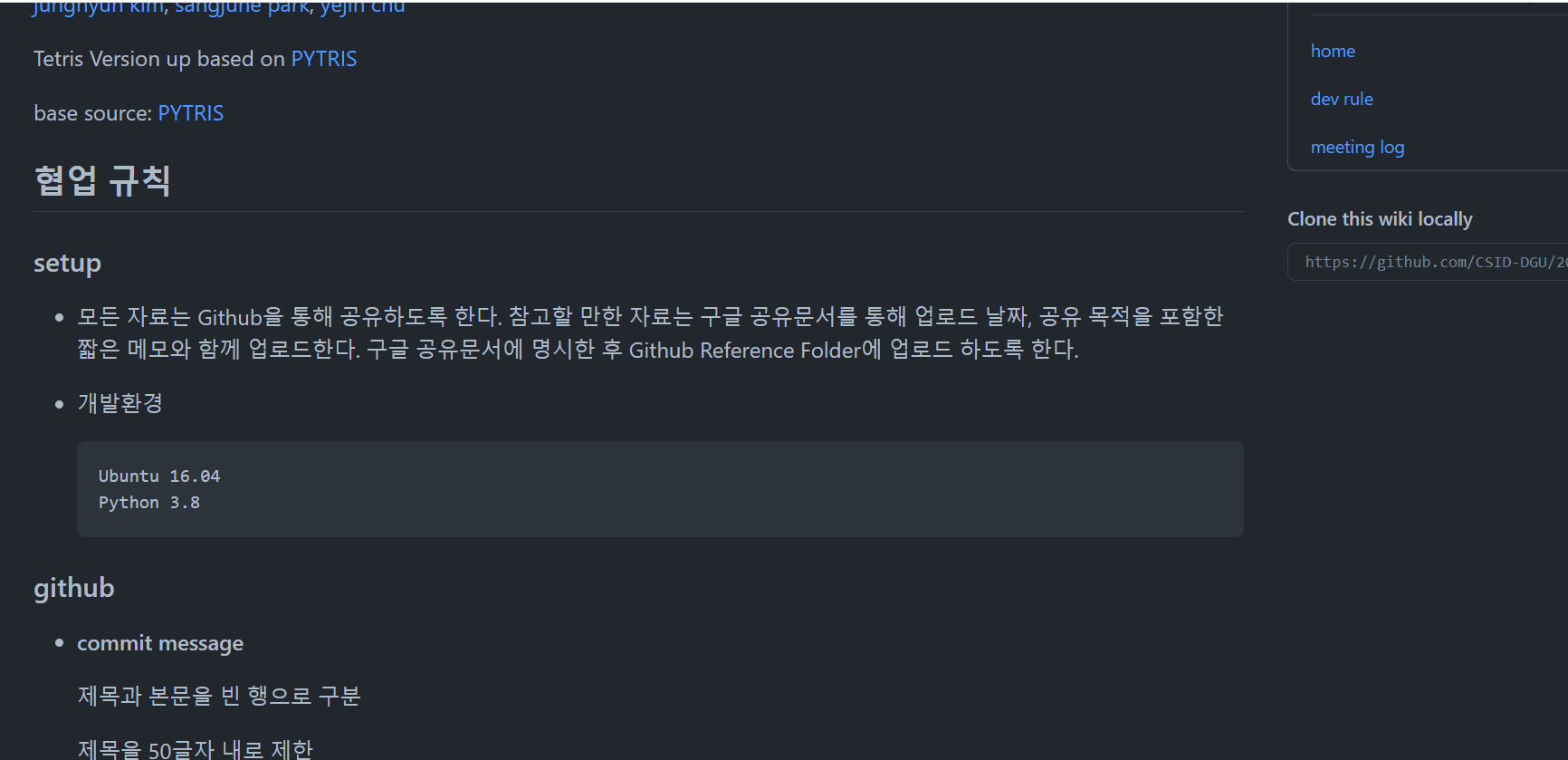


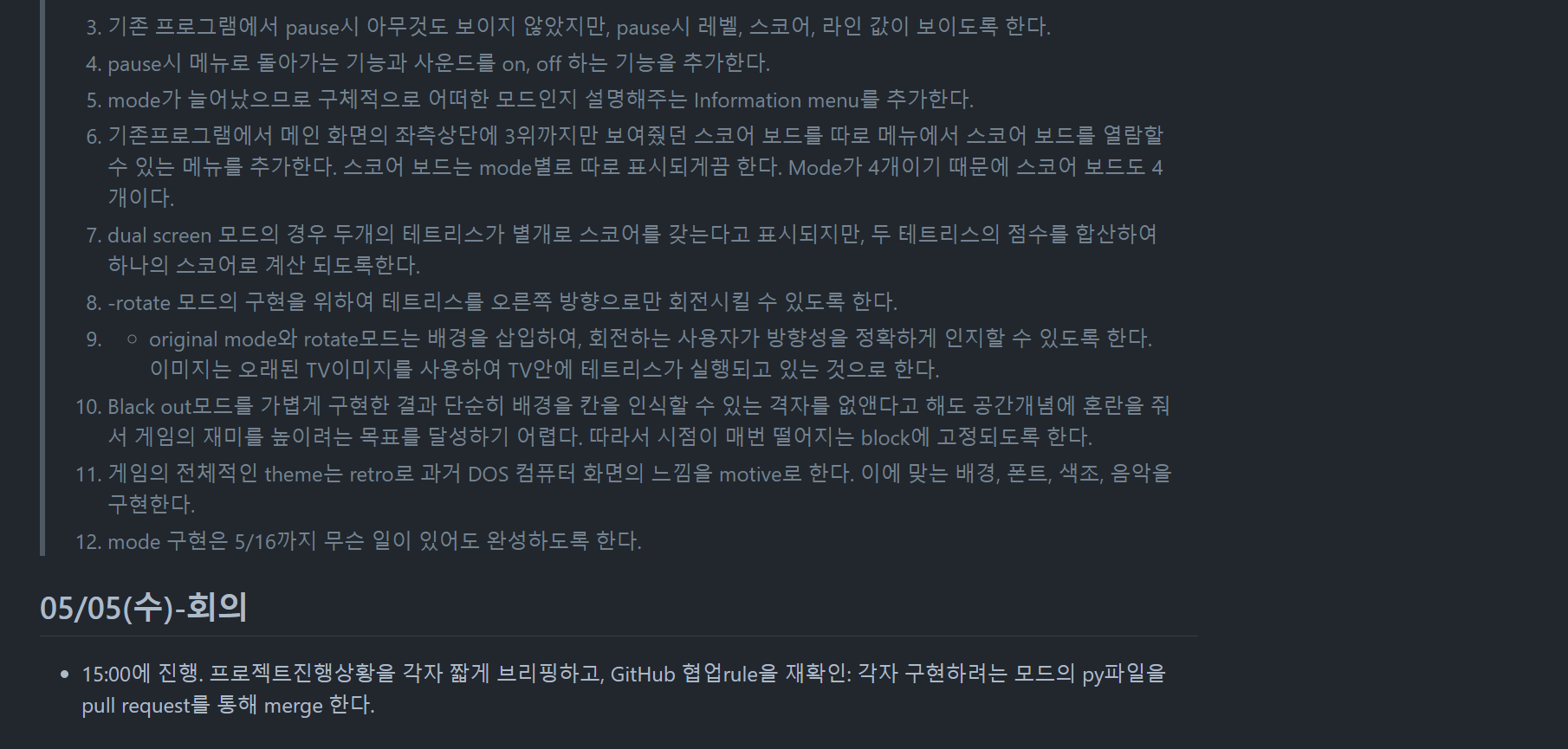




****





-rotate mode 구현중 (사이드 바는 사이즈 때문에, 고칠 예정, 화면 회전 후 키설정까지)

-blackout mode 구현중(화면시점 고정까지)

-dual Screen mode 구현중(테트리스 두개 생성까지)

-Git Hub wiki, readme 관리(회의록, Dev rule)

-work flow 및 세부사항 점검

-issue 적극적으로 활요

-11주차 주간 보고서 작성

**4.다음주 계획**

-상준: rotate mode 구현완성 + main 모듈화 + 인터페이스 source 자료수집

-예진: blackout mode 구현완성 + main 모듈화 + 인터페이스 source 자료수집

-정현: dual Screen mode 구현완성 + main 모듈화 + 인터페이스 source 자료수집