

1 조 오픈마인드

프로젝트 제안서

- Open Source Tetris Upgrade -

〈 오픈마인드 팀원 〉

2017110710 법학과 곽명준
2016111283 경찰행정학과 김병철
2013110344 중어중문학과 이원우

1 조 오픈마인드 제안서

OM Tetris

목차

1. [Base Source](#)
2. [선정 이유](#)
3. [SWOT 분석](#)
4. [주제](#)
5. [목표](#)
6. [세부목표](#)
7. [개발 환경](#)
8. [역할 분담](#)
9. [기대효과](#)
10. [프로젝트 타임라인](#)
- 부록 1. [자료관리 / 코딩 / 커밋 규칙](#)
- 부록 2. [기능별 Flow Chart 설계](#)
- 부록 3. [Reference](#)

BASE SOURCE

i PYTRIS

✓ 구성

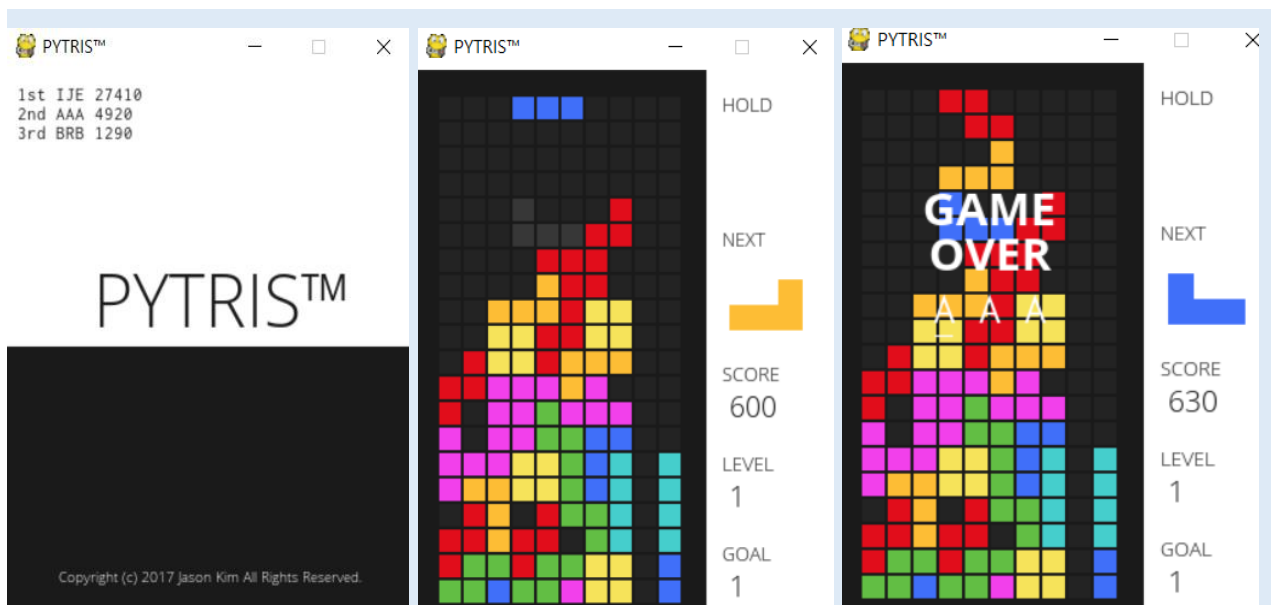
pytris.py : main 함수 및 각종 기능

mino.py : 블록 모델

assets : 폰트, 이미지, 사운드

✓ MIT License

✓ URL : <https://github.com/k0626089/PYTRIS>



선택 이유

i REASON

- ✓ 첫 프로젝트를 수행함에 있어 방대한 오픈소스를 보유한 테트리스가 적합하다고 판단
- ✓ 팀원이 모두 익숙한 Python 언어로 되어 있음
- ✓ Pygame 를 활용하여 간결한 소스 코드로 제작되어 있음
- ✓ 모듈화가 잘 되어 있음
- ✓ 게임 흥미 요소의 부족
- ✓ 게임 플레이시 여러가지 불편한 점 개선 여지
- ✓ MIT License 가 사용되어 수정, 배포의 권한이 허용됨

SWOT 분석

i SWOT

- ✓ Strength
 - 테트리스의 기본적인 기능들이 정상 작동함
 - 수정/변경이 용이하게 구성
 - 최소한의 모듈로 구성 / 직관적인 소스코드
- ✓ Weakness
 - 기존 테트리스 게임들과의 차별화 부족
 - 게임 플레이 시 불편한 부분 다수 존재 (인터페이스, 조작, 창 크기, 등)
 - 조작법 설명 부족
- ✓ Opportunity
 - 단순하게 구성된 게임이라 개선 가능한 부분 많음
 - 대중적인 게임으로 많은 사람들이 즐기는 게임임
 - 다양한 버전의 변형 / 개조판이 존재함
- ✓ Threat
 - 기능 추가 시 생각하지 못한 오류 발견 가능성
 - 적절한 시간/역할의 분배가 필요할 것
 - 더 재미있고 빠른 게임이 존재함

주제

i Subject

- ✓ Github 을 사용하여 팀 프로젝트 협동성과 역량 증진
- ✓ Git, Linux 활용 능력 증진
- ✓ 오픈소스 기반 테트리스의 게임성, 재미 향상 및 편의성 개선

목표

i Goals

- ✓ 인터페이스 업그레이드
- ✓ 그래픽 업그레이드
- ✓ 사운드 추가
- ✓ 난이도 추가
- ✓ 새로운 게임 모드 추가
- ✓ 게임 가독성 업그레이드

세부 목표

i Detailed Targets

- ✓ 인터페이스 업그레이드
시작 화면 개선 / 점수판 개선 / 종료 화면 개선 / 다음 블록 출력화면 개선 / 창 크기 조정
- ✓ 그래픽 업그레이드
블록 그래픽 업그레이드 / Combo 그래픽 추가
- ✓ 사운드 추가
BGM 추가 / 효과음 추가 / 사운드 속도 조절 / Combo 사운드 추가
- ✓ 난이도 추가
기존 점수 방식 개선 / 난이도 조절 기능 추가
- ✓ 새로운 게임 모드 추가
싱글 모드 / PvP 모드 / 아이템 모드 / 새로운 블록 모드 추가
- ✓ 게임 가독성 업그레이드
테두리 변경 / 폰트 변경 / 이미지 개선
- ✓ 그 외
좌우 빠른 이동 개선 / Combo 기능 추가 / 등

개발 환경

i Development Environment

- ✓ 언어 : Python 3.7
- ✓ 툴 : Pycharm or Anaconda
- ✓ OS : Ubuntu

기대효과

목표	세부내용
a. 인터페이스 업그레이드	게임의 편의성 증진
b. 그래픽 업그레이드	게임의 집중도와 재미 증진
c. 사운드 추가	게임의 리듬감과 몰입도 증진
d. 난이도 추가	사용자의 숙련도에 맞는 개별 난이도 선택지 제공
e. 새로운 게임 모드 추가	새로운 모드의 테트리스 경험 제공
f. 게임 가독성 업그레이드	게임 화면의 직관성 증진
g. Combo 기능 추가	게임의 박진감과 긴장감 증진
h. 좌우 빠른 이동 개선	편리한 블록 이동성 제공

Commit 단위 역할 분담

i Role

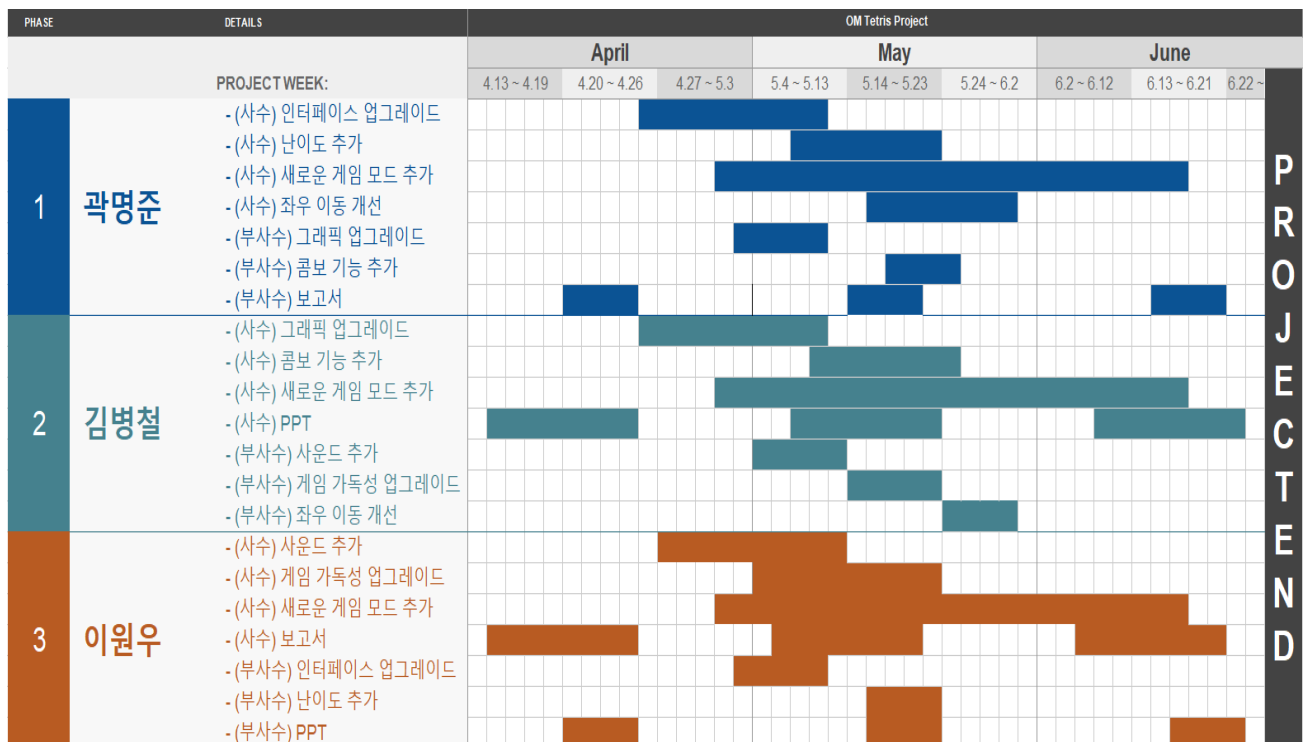
- ✓ 광명준 – 조장
 - <a. 사수> 시작 화면 개선 / 점수판 개선 / 종료 화면 개선 / 다음 블록 출력화면 개선 / 창 크기 조정
 - <b. 부사수> 그래픽 업그레이드 / 보고서
- ✓ 김병철
 - <b. 사수> 블록 그래픽 업그레이드 / Combo 그래픽 추가 / 그래픽 업그레이드 / Combo 기능 추가 / 새로운 게임 모드 추가 / ppt
 - <c. 부사수> 사운드 추가 / 좌우 이동 개선
- ✓ 이원우
 - <c. 사수> BGM 추가 / 효과음 추가 / 사운드 속도 조절 / Combo 사운드 추가 / 테두리 변경 / 폰트 변경 / 이미지 개선 / 보고서
 - <a. 부사수> 인터페이스 업그레이드 / ppt

Branch 단위 역할 분담

i Role

- ✓ **곽명준 – 조장**
 - <사수> 기존 점수 방식 개선 / 난이도 조절 기능 추가
 - <부사수> Combo 기능 / PvP 모드 / 인터페이스 개선
- ✓ **김병철**
 - <사수> Combo 기능 추가 / PvP 모드 추가 / 인터페이스 개선
 - <부사수> 좌우 이동 개선/ 아이템 모드 추가
- ✓ **이원우**
 - <사수> 아이템 모드 추가 / 좌우 빠른 이동 기능 추가
 - <부사수> 기존 점수 방식 개선 / 난이도 조절 기능 추가

프로젝트 타임라인



* 공통 사항인 코드 분석, 오류 검토, 아이디어 회의, 상호 협력, 발표 준비 등은 상시 진행하도록 함

* 별도 파일 첨부

부록 1. 협업 규칙

i Project Rule

- ✓ 별도 파일 첨부

2020-1-OSSP1-OPENMIND-1

팀프로젝트 협업 규칙

1. 프로젝트 개발환경 세팅

1) 자료보관

Develop 브랜치에 올라간 이슈에 관련한 모든 자료는 깃헙에서 공유해야함
데이터 셋이나 발표자료는 구글드라이브에서 공유

2) 라이브러리

```
import pygame  
import operator
```

3) 개발환경

Anaconda 4.8.2
Python 3.7.4
Ubuntu18.04

2. 깃헙 사용규칙

1) Commit 메시지 작성 방법

from <http://blog.weirdx.io/post/33832>

- 제목과 본문을 한 줄 띄워 분리하기
- 제목은 영문 기준 50 자 이내로
- 제목 첫글자를 대문자로
- 제목 끝에 . 금지

- 제목은 명령조로
- 본문은 영문 기준 72 자마다 줄 바꾸기
- 본문은 어떻게보다 무엇을, 왜에 맞춰 작성하기

2) Branch 이름

from <https://gmlwjd9405.github.io/2018/05/11/types-of-git-branch.html>

1. Develop branch 다음 출시 버전을 개발하는 브랜치 즉, 모든 기능이 추가되고 버그가 수정되어 배포 가능한 안정적인 상태라면 master 브랜치에 merge 한다.
평소 개발의 기준
2. Feature branch 새로운 기능 개발 및 버그 수정이 필요할 때마다 develop 브랜치로부터 분기한다. 로컬 저장소에서 관리한다.
개발이 완료되면 develop 브랜치로 merge 하여 공유한다.
 1. develop 브랜치에서 새로운 feature 브랜치를 분기한다.
 2. 새로운 기능에 대한 작업을 수행한다.
 3. 작업이 끝나면 develop 브랜치로 병합한다.
 4. 더 이상 필요하지 않은 feature 브랜치는 삭제한다.
 5. 새로운 기능에 대한 feature 브랜치를 중앙 원격 저장소에 올린다(push)
3. feature 브랜치 이름 기준
 - master, develop, release-(RB_), or hotfix- 제외
 - [feature/기능요약] 형식을 추천 EX) feature/login

3) 변수 작성 방법

변수는 한 목적으로만 사용하기

변수는 항상 소문자로 시작하고, 다음 단어는 대문자로 구분하기

함수의 매개변수 역시 그 의미를 명확히 표현하는 단어로 작성하기

4) 함수 작성 방법

함수명은 항상 대문자로 시작하고, 다음단어는 대문자로 구분하기

변수명과 함수명을 같게 하지 말기

함수의 접두사 공통으로 사용하기 함수

5) 버전 표기 기준

버전 x.y.z

- x: 기존 버전과 호환되지 않게 바뀔 시(대규모 업데이트)
- y: 기존 버전과 호환되면서 새로운 기능을 추가할 때

추후 추가되는 함수는 순차적으로 번호 부여

- z: 기존 버전과 호환되면서 버그 수정했을 시 수정하지 않을 시: 0
수정할 시: 1
- 부가설명: 기여한 부분 설명 예) GetInput() 버그수정 및 새로운 기능(Enter 를 누를 시 게임 일시정지) 추가. 기존 버전과 호환됨: 0.2.1-enter

6) Merge 규칙

(1) Merge 확인 작업

각자 맡은 코드 수정 후, Issue 표에 수정한 내용을 정리한다. 팀원은 Pull request 에 있는 해당 Issue 를 보고 피드백을 남긴다.(merge 찬성 혹은 개선 내용 추가)

(2) Merge 결정 후

Merge 를하기로 결정났으면 merge 를 진행한다.

7) Issue 규칙

한 기능 당 하나의 issue 를 생성하여, 댓글을 통해 의견을 주고받는다.

8) 문서화 규칙

수정할 곳에 관한 issue 를 생성하여, 댓글을 통해 피드백을 남긴다.

IMPROVE FOUND OPEN SOURCE CODE TO CLEAN CODE BASED ON READABILITY

- 전역변수의 사용 줄이기: 하나의 함수 내에서만 쓰이는 전역 변수를 해당 함수 내에 넣어주어 지역변수로 변환하여 준다.

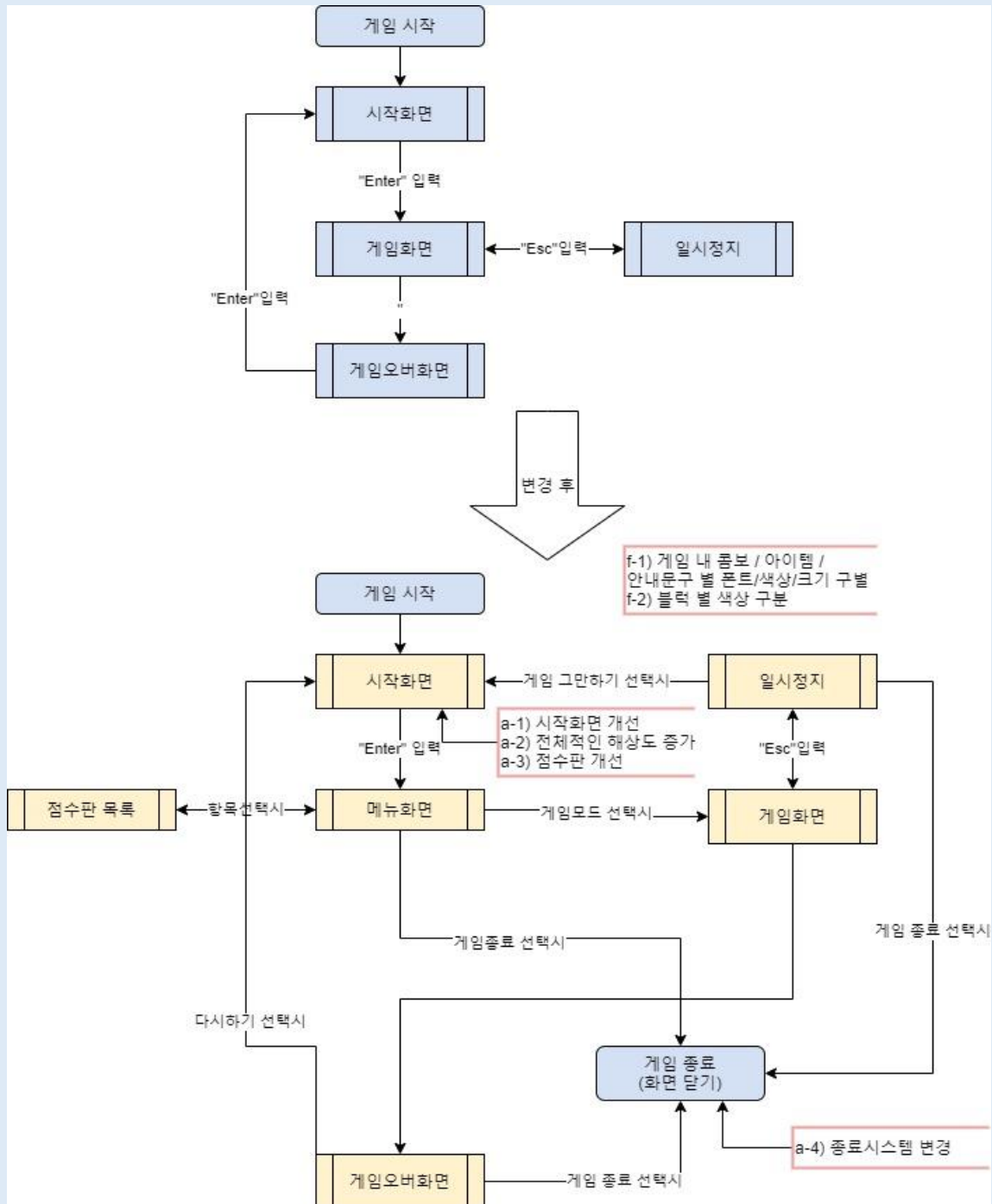
- 같은 의미를 갖는 변수들 구조체로 선언하기
- 하나의 함수에는 하나의 기능을 하도록 설계하기
- 매크로 사용하기: 매크로번화를 통해 상수의 가독성을 높여준다.
- 조건식의 함수화: 조건식을 변수나 비교연산자로 표현하기 보단 함수로 표현하여 코드의 가독성을 높인다.
- 함수 분리: 조건식을 변수나 비교연산자로 표현하기 보단 함수로 표현하여 코드의 가독성을 높인다.

IMPROVE ERRORS OF FOUND OPEN SOURCE CODE

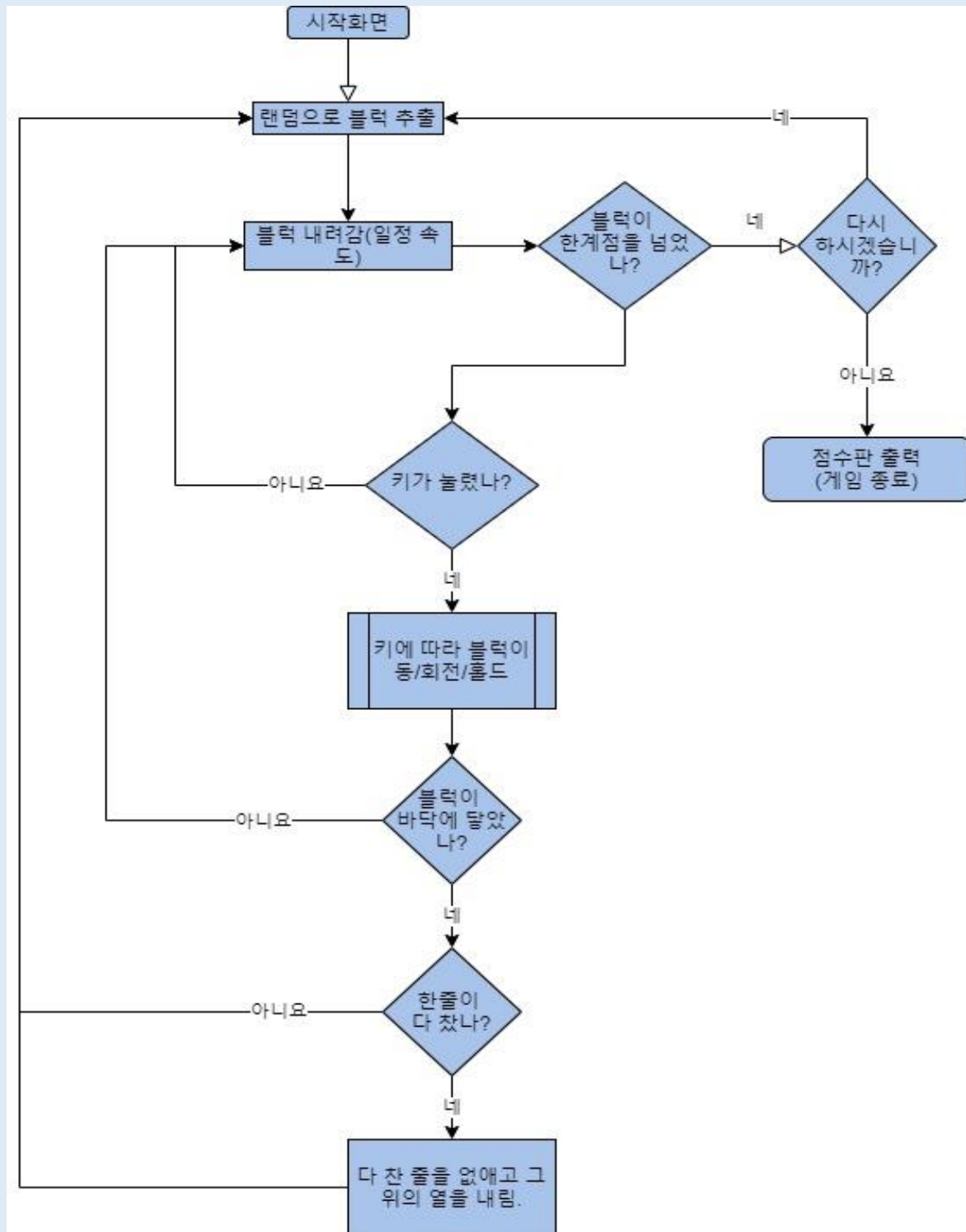
- 컴파일 환경 Warning Level 오류 개선: Warning Level 을 최상으로 했을 때의 빌드 오류를 수정하여 준다.
- 조건문 else 문 오류 개선: 조건문을 작성할 때 if 문만 사용하고 else 문을 써주지 않아 오류처리에 실패할 경우가 있기 때문에 항상 if 문을 써주면 else 문을 반드시 써서 오류처리를 해준다.
- 반복문 return 오류 개선: 조건문과 마찬가지로 반복문에서도 항상 return 을 잘 활용하여 무한루프에 빠지지 않게 하고, 함수일 경우 return 값을 잘 정의해 주어야 한다.

부록 2. 기능별 Flow Chart

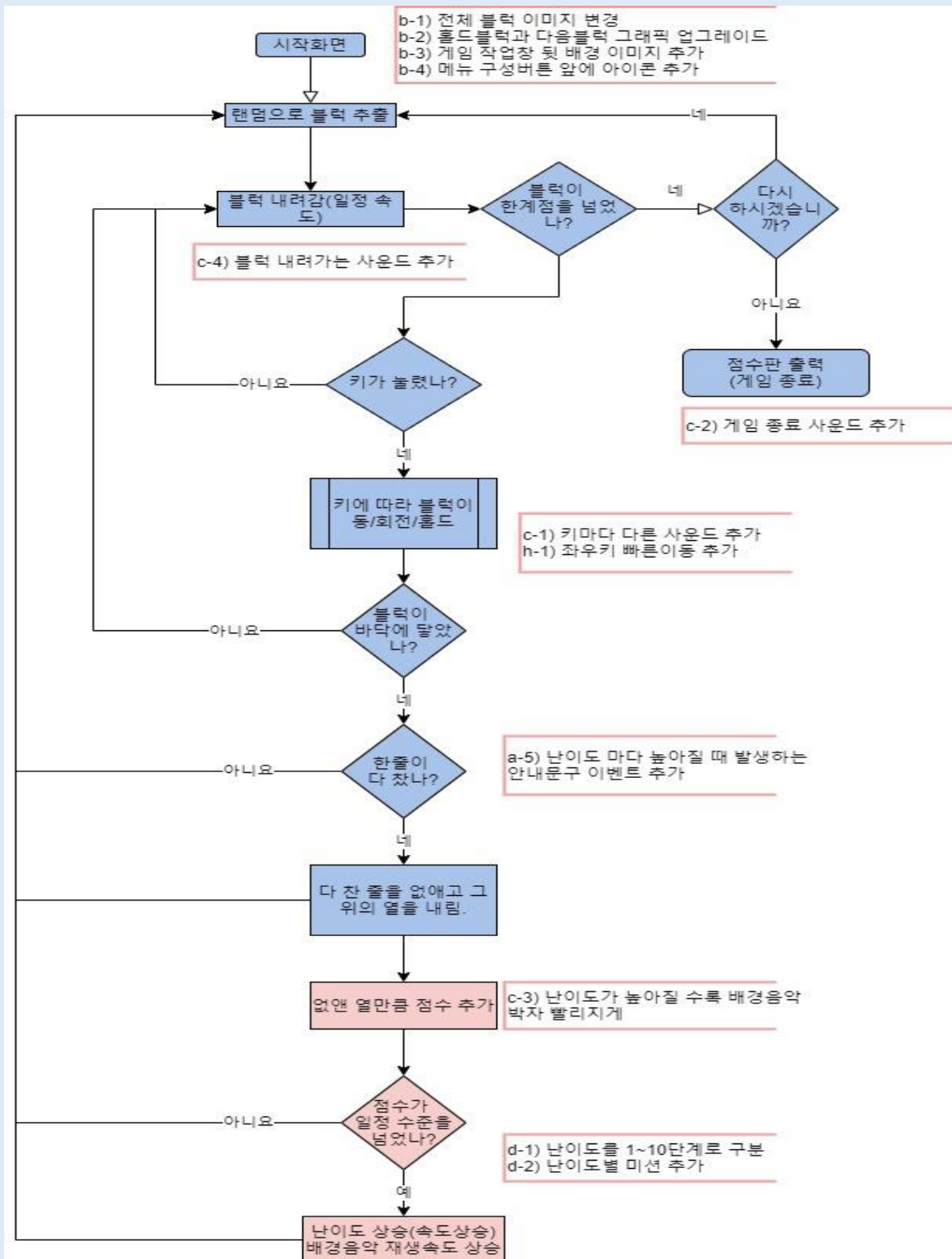
i Screen Flow Chart



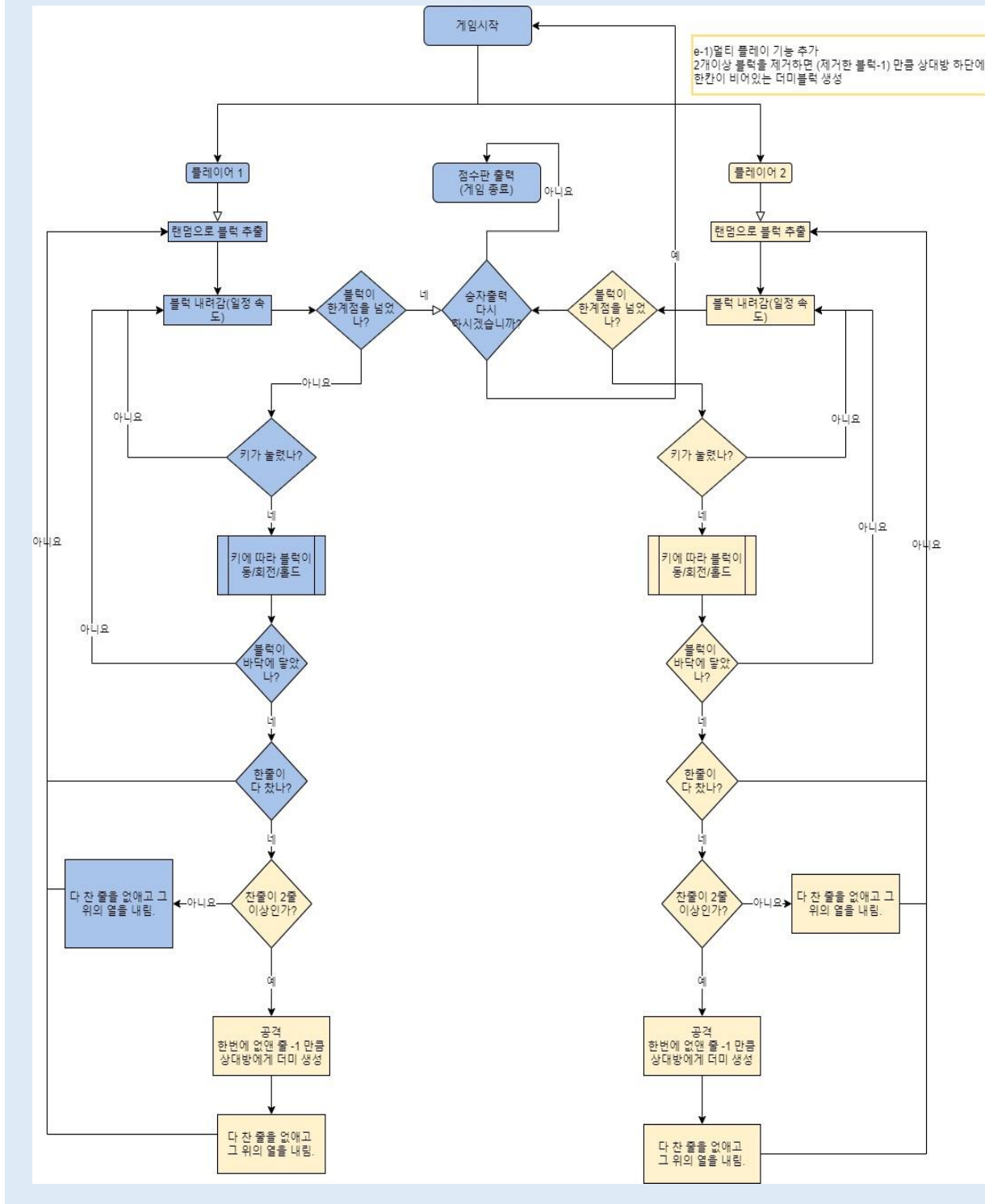
i Single Mode Flow Chart I



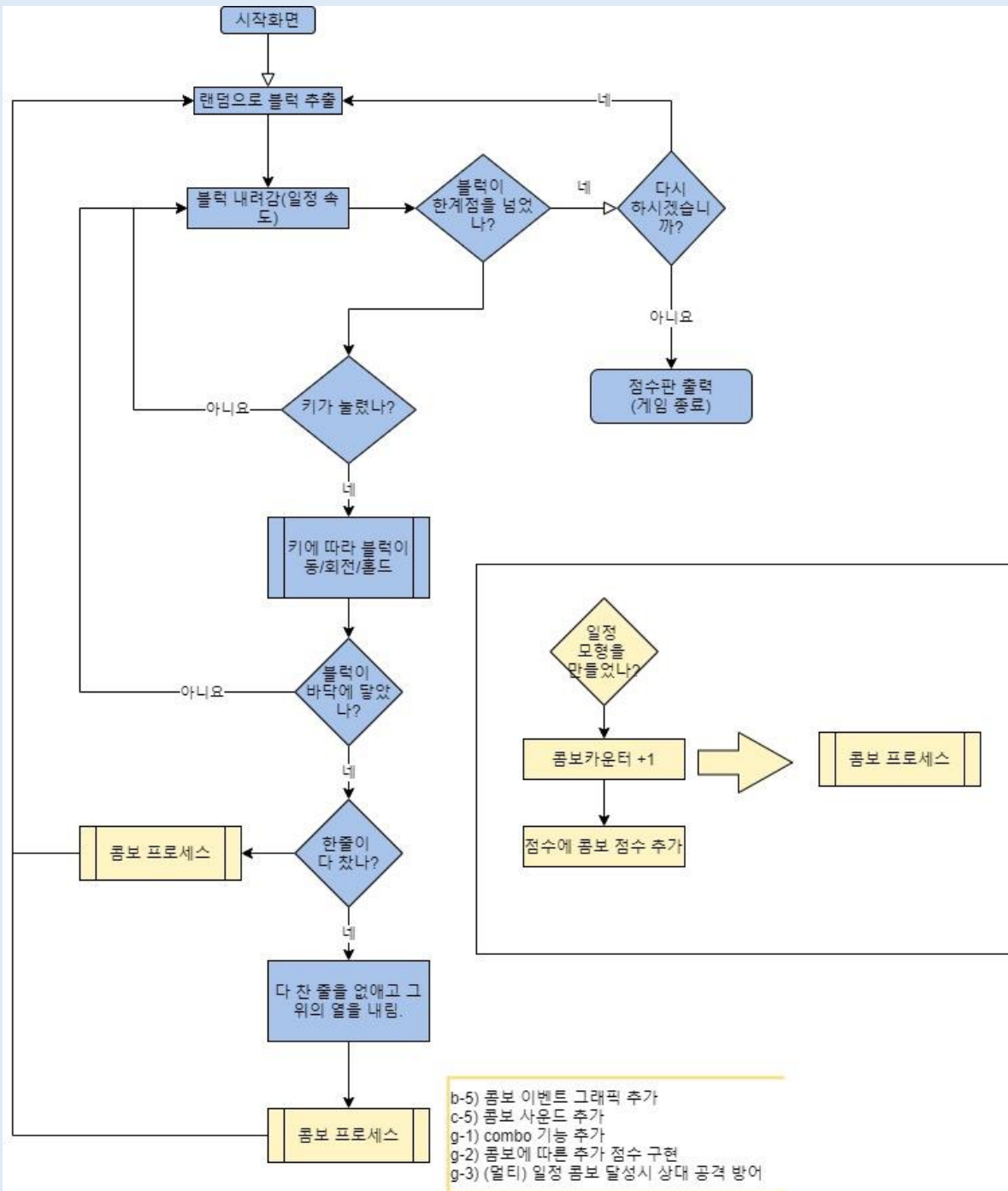
i Single Mode Flow Chart II



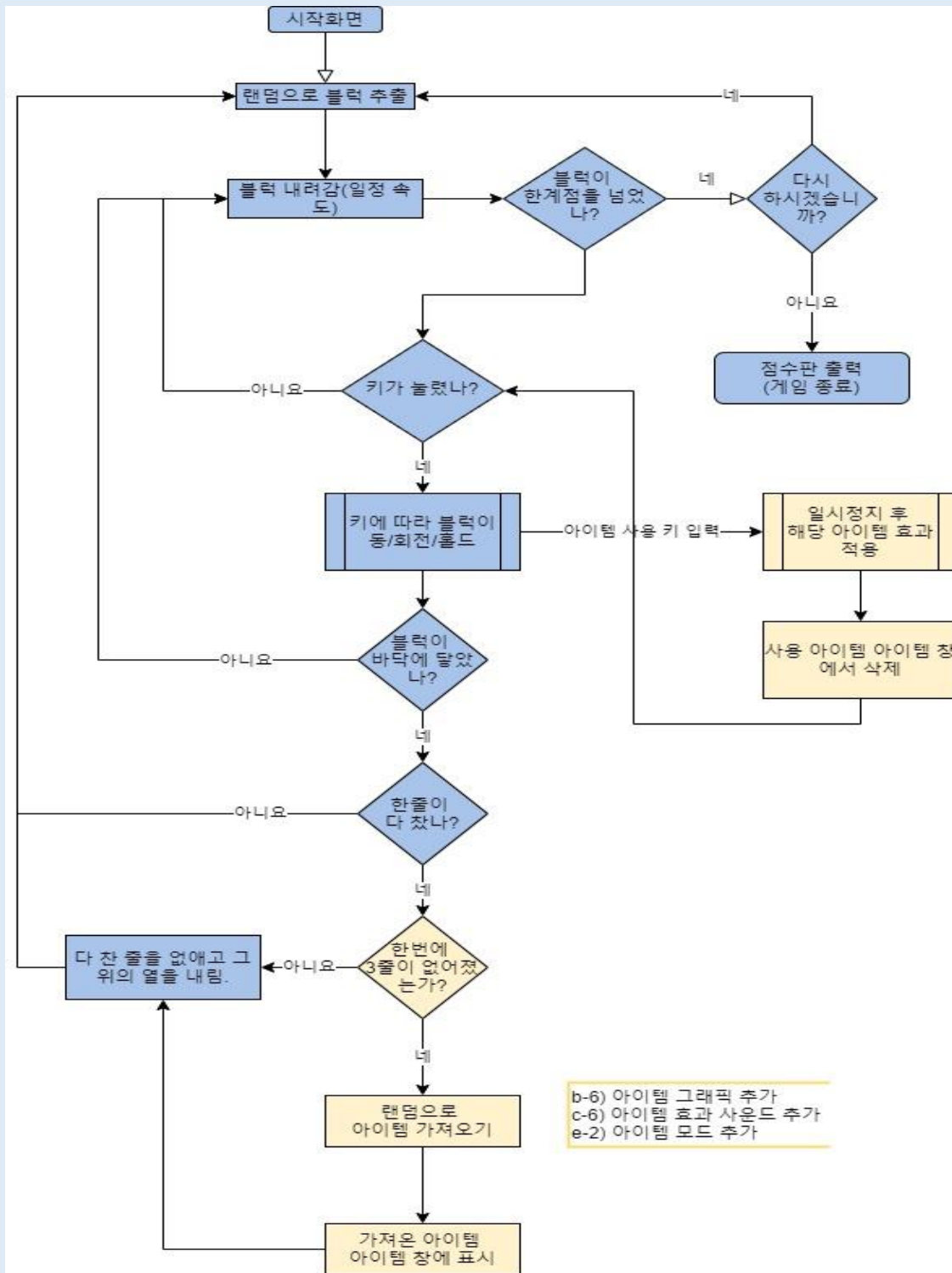
i PvP Mode Flow Chart



i Combo Flow Chart



i Item Mode Flow Chart



부록 3. Reference



Reference

- ✓ Tetris-Battle

URL : <https://github.com/xuyuwei/tetris-battle>

License : No License (No License 에 대한 규정 <https://choosealicense.com/no-permission/>)

활용 : 그래픽, 사운드, 이미지 소스