

공개SW 실무 최종보고서

# 목차

**프로젝트 일정**

**프로젝트 개요**

**1) 프로젝트 제목**

**2) 선정한 OSS의 소스코드**

**3) C반 1조 Git Repository**

**4) 선정 이유**

**5) 사용하는 라이브러리&DB**

**선정한 OSS의 장･단점**

**타(他) 게임과의 비교**

**1) 애니팡**

**2) 캔디 크러쉬 사가**

**소스의 전체 구조**

**1) 디렉토리 구조**

**2) 파일 구조**

**2-1) Game 폴더**

**2-2) Source 폴더**

**2-3) Windows Installer 폴더**

**기능 설계**

**주요 메소드 분석**

**UML 다이어그램**

**주요 사용자 인터페이스(UI)**

**개선점**

# 프로젝트 일정

|  |  |  |
| --- | --- | --- |
| 작업 | 기한 | 참고 |
| 최종 제안서 | 2015.11.2 | 5일 발표자 김기범 |
| 시스템 설계서 | 2015.11.16 | 19일 발표자 김여진 |
| 상세 설계서 | 2015.11.30 |  |
| 최종 보고서 | 2015.12.7 | 10일 발표자 이현욱 |

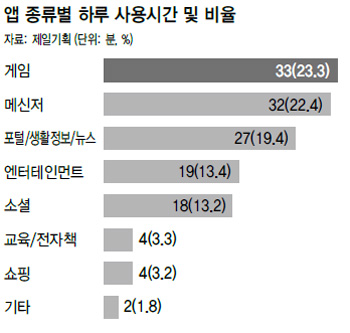
# 프로젝트 개요

**1) 프로젝트 제목:** Block Attack(블록 어택)

**2) 선정한 OSS의 소스코드:** http://sourceforge.net/p/blockattack/git/ci/master/tree/

**3) C반 1조 Git Repository:** <https://github.com/jay47/OSS_Project>

**4) 선정 이유:**

2시간 23분. 이 시간은 대한민국 국민의 하루 평균 스마트 폰 앱의 사용시간(전화, 문자 제외)이다. 그 중에서 ‘디지털 놀이’ 중 하나인 게임에 가장 많은 시간을 쓰고 있다는 걸 알 수 있다. 여기서 주목할 점들은 스마트폰 게임은 청소년들이 주로 즐길 것이라는 인식과 달리 게임을 가장 많이 하는 세대는 30~40대인 것이다. 그리고 스마트 폰을 가장 짧게 사용하는 50대 이상도 하루에 평균 1시간 28분을 사용할 정도로 전 연령층이 꽤 많은 시간을 스마트 폰을 사용하는데 쓰고 있다는 걸 알 수 있다. [[1]](#footnote-1)

또한, 게임물관리위원회가 발간한2014년 국내외 게임물 등급분류현황과 사후관리통계 등을 담은 ‘2015 게임물 등급분류 및 사후관리 연감’에 의하면 2014년 국내 등급 분류되어 유통된 게임물 총 521.355건 중 99.7%인 519.931건이 스마트 폰 게임인 것으로 확인됐다.[[2]](#footnote-2)

이처럼 우리나라 국민들은 스마트 폰 ‘게임’을 즐겨 하며 2014년 유통된 게임물 중 99.7%가 스마트 폰 게임일 정도로 현재 한국은 스마트 폰 게임이 대세라 할 수 있다. 그래서 우리는 최대한 남녀노소 누구나 쉽게 사용하고 즐길 수 있을만한 게임 위주로 찾아 보았다. 여러 가지를 고려한 결과 최대한 복잡하게 머리를 쓰지 않으면서 누구나 쉽게 접근할 수 있는 룰을 가진 퍼즐게임 Block Attack[[3]](#footnote-3)을 가(假)선정했다. 블록어택은 사용자가 선택한 블록의 위치를 바꾸어 같은 종류의 블록이 3개 이상 겹치게 만들면 점수를 획득하며 블록이 제거 되는 간단한 게임이다. 또한, 최초로 국내 사용자만으로 2000만 다운로드 이상을 돌파할 만큼 선풍적인 인기를 끌었던 ‘애니팡’이나 ‘캔디크러쉬사가’ 등 성공 사례도 여럿 있어 비교해가며 개선점을 찾는데 용이하다는 점도 플러스 요인이었다.

**5) 역할 분담**

|  |  |  |
| --- | --- | --- |
|  | 이름 | 역할 |
| 팀장 | 김기범 | 소스분석+제안서+일정조정 |
| 팀원1 | 이현욱 | 소스분석+최종 보고서 |
| 팀원2 | 김여진 | 소스분석+상세 설계서 |
| 팀원3 | 안지희 | 소스분석+문서작성 보조 |

**6) 사용하는 라이브러리&DB**

라이브러리 : SDL - www.libsdl.org

DB : 사용 안 함

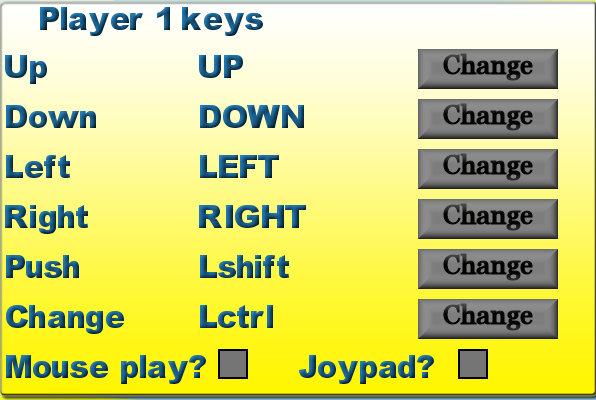
# 선정한 OSS의 장단〮점

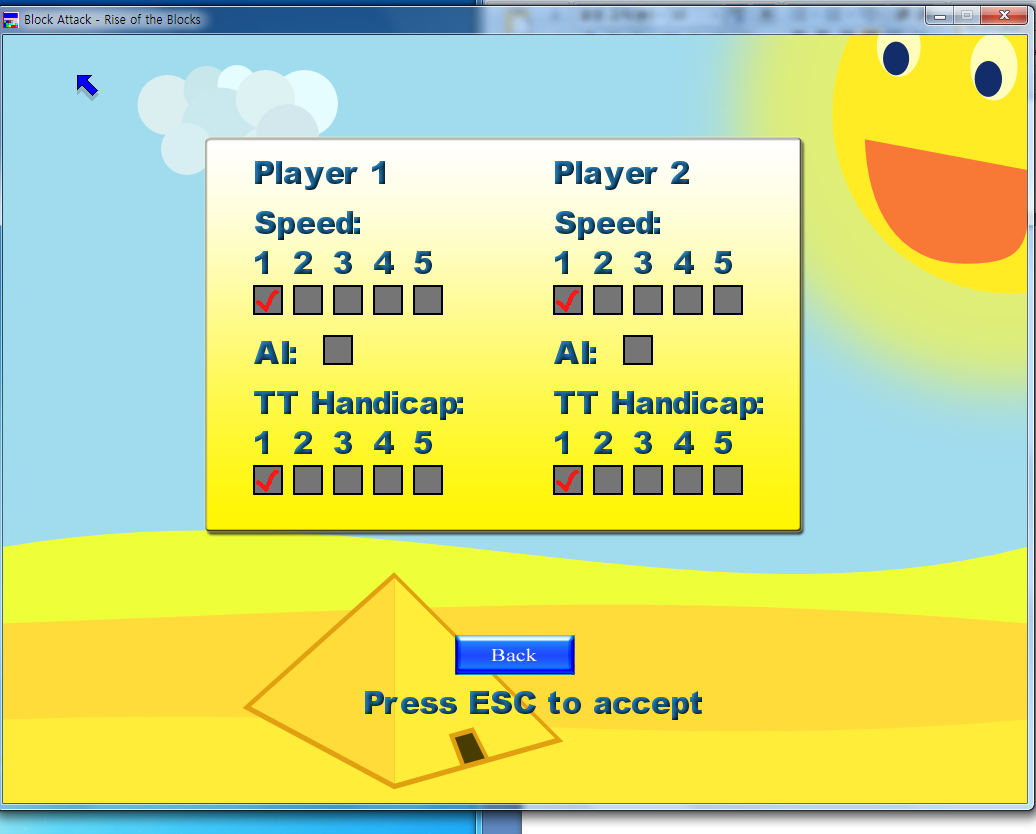
**장점 1)**



혼자 하기뿐만 아니라 둘이서 할 수도 있고 시간 제한, 스테이즈 클리어 등 다양한 게임 방식이 있다.

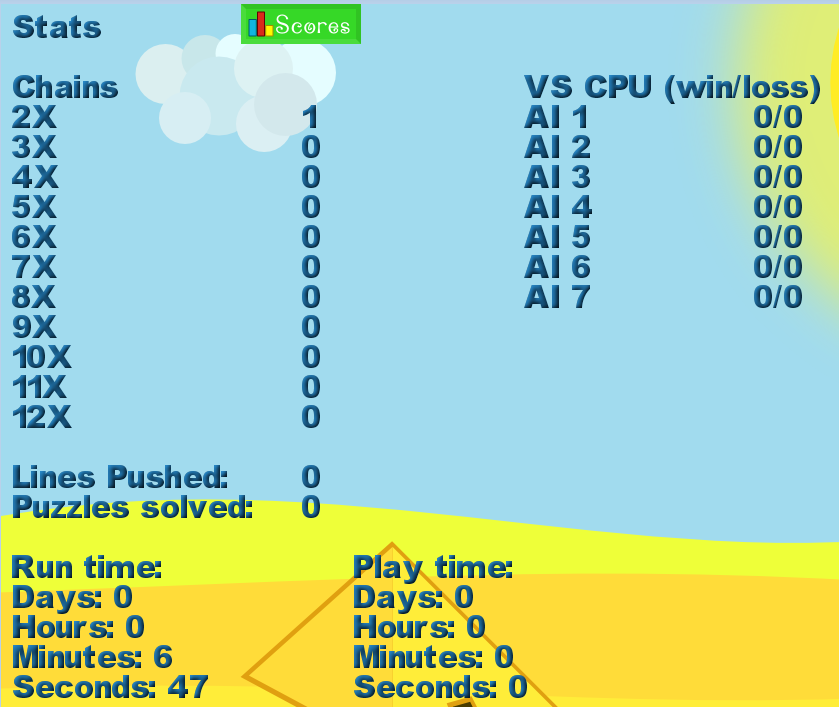
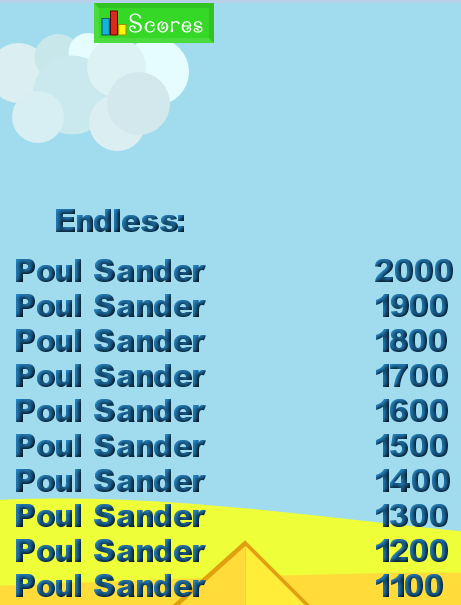
**장점 2)**



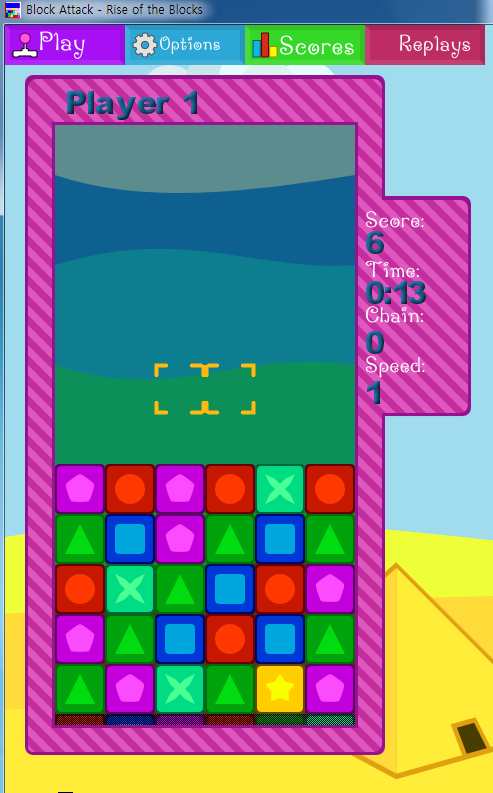
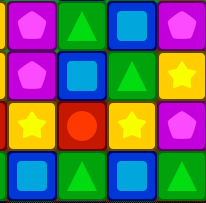


옵션에서 Key 설정과, 마우스&조이패드 플레이 설정, 속도와 핸디캡 설정 등 다양한 설정이 가능하다.

**장점 3)**

****스코어 등 다양한 통계자료도 볼 수 있다.

**단점 01)**

****일시 정지가 없고 ESC 키를 누르면 단번에 종료가 돼서 처음 플레이 해본 사용자라면 당황할 수 있다.

**단점 02)**

EMB000014dc5a95배경과 블록의 모양과 색상이 너무 단조롭고 밋밋하다.

**단점 3)**

블록 위치를 상하로는 바꿀 수 없고 좌우로만 바꿀 수 있다.

**단점 4)**

게임의 재미를 더할 콤보 시스템과 아이템 등이 없다.

**단점 5)**



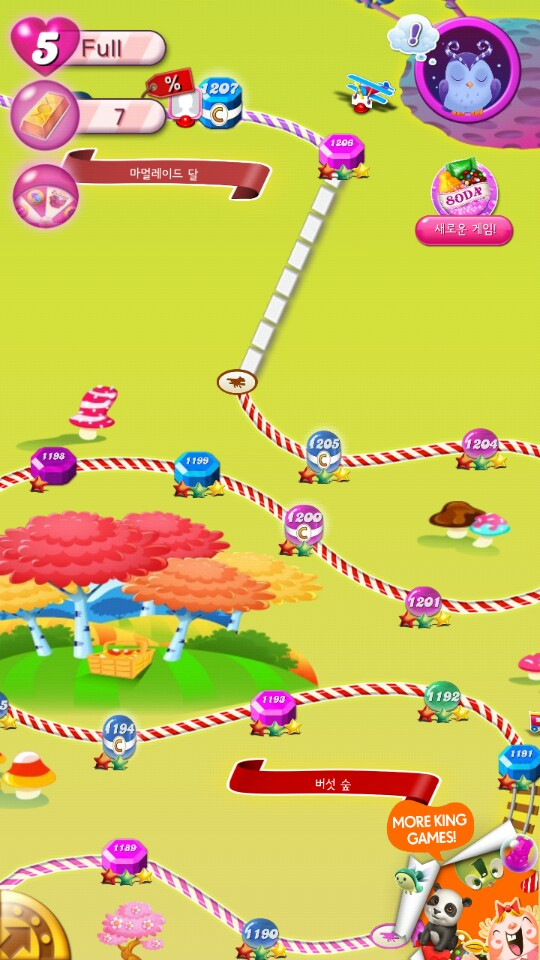
초기화면 모습인데 작동키는 무엇이며 룰은 어떤지 등 튜토리얼이 없어 처음 해보는 사용자라면 당황할 수 있다.

# 타(他) 게임과의 비교

**1) 애니팡[[4]](#footnote-4)**

애니팡의 장점은 여러 가지가 있다. 그 중에서도 동물을 사용한 아기자기한 UI로 여성유저들의 마음을 사로잡은 점이 특징이다. 또한 콤보 시스템과 다양한 아이템, 화려한 이펙트, 그리고 카카오톡을 연동한 친구들과의 경쟁시스템으로 유저들이 계속 게임을 하도록 유도한다는 점, 또한 특징이다.

**2) 캔디 크러쉬 사가[[5]](#footnote-5)**

캔디 크러쉬 사가는 페이스북을 시작으로 iOS, Android, Fire OS, Windows Phone, Windows 10 그리고 타이젠까지 다양한 OS에서 지원이 된다는 것이 큰 장점이다. 또한, 다양한 아이템과 자신의 레벨이 어느 정도까지 왔는지 맵으로 표시해주는 기능 등 유저가 직관적으로 게임을 파악하는데 도움을 준다는 것이 특징이다.

# 소스의 전체 구조

**1. 디렉토리 구조**

Block Attack



Windows installer



Game

fonts



music

sound

gfx



source



assets

code

manual

**2. 파일 구조**

**2-1. Game 폴더**

1) fonts 폴더 → FreeSerif.ttf, PenguinAttack.ttf

: 게임 내의 폰트들

2) gfx 폴더 → bricks, menu 등 각종 png 파일

: 게임 내에 쓰이는 각종 이미지들이 있음

3) music 폴더 → bgMusic.ogg, highbeat.ogg

: 게임 내의 BGM

4) sound 폴더 → applause.ogg, cameraclick.ogg 등

: 게임 내의 효과음

**2-2. source 폴더**

1) assets 폴더 → gfx 폴더 → bricks, buttons 폴더 등 → blue, buttons 등 각종 svg[[6]](#footnote-6) 파일

: svg는 2차원 벡터 그래픽을 표현하기 위한 XML 기반의 파일 형식임

2) code 폴더 → main.cpp, BlockGame.cpp, BlockGame.hpp 등 각종 코드

: BlockAttack의 소스파일들

3) manual 폴더 → block\_attack\_manual.tex

: 제작자가 영감을 얻은 게임(닌텐도의 ‘테트리스’), 사용한 라이브러리&프로그래밍 언어, 작동 방법 등 블록 어택에 대한 전반적인 소개를 한 파일.

**2-3. windows installer 폴더**

install\_script.nsi 파일

: 윈도우에서 BlockAttack을 설치하는데 도움을 주는 파일

# 기능 설계

|  |  |
| --- | --- |
| 에디터 | 설명 |
| 기능 | 에디터 기능(예를 들어 보드판 조정)등이 있다. |
| 소스파일 | TheBoard.hpp, TheBoard.cpp, BoardHolder.hpp, BoardHolder.cpp  EditorInterface.hpp, EditorInterface.cpp , editorMain.hpp, editorMain.cpp |
| 역학관계 | TheBoard→ BoardHolder→ EditorInterface→ editorMain 순으로 영향을 준다. |
| 클래스 | TheBoard의 클래스 TheBoard,  BoardHolder의 클래스 BoardHolder,  EditorInterface의 클래스 EditorInterface,  editorMain의 클래스 aButton, theEditor |
| 함수명 | TheBoard의 moveUp, moveDown 등의 함수로 보드판을 조정하고  BoardHolder의 saveBoards, removeBoard 등의 함수로 보드판을 저장하거나 지운다.  EditorInterface와 editorMain에서는 여러 Button 함수를 통해 보다 편하게 에디팅 할 수 있게 도움을 준다. |
| 참조 | editor 폴더에 소스파일들이 있음, 본 게임에서는 적용되지 않았음. |
| 코드 | <https://github.com/jay47/OSS_Project/tree/master/BlockAttack/source/code/editor> |

|  |  |
| --- | --- |
| 게임 스코어 | 설명 |
| 기능 | 파일명을 지정하여 점수를 저장해주고, 저장된 점수를 확인하여 등수를 정해준다. |
| 소스파일 | highscore.h, highscore.cpp |
| 클래스 | highscore.h에 있는 Highscore를 highscore.cpp에서 사용한다. |
| 함수명 | isHighScore는 저장된 점수가 가장 높은 점수인지 확인한다. addScore는 점수를 비교하여 등수를 정해준다. getScoreNumber와 getScoreName은 점수와 이름을 반환한다. |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/highscore.cpp>  <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/highscore.h> |

|  |  |
| --- | --- |
| 세이브&로드 | 설명 |
| 기능 | Save와 Load를 할 수 있게 해준다. |
| 소스파일 | listFiles.h, listFiles.cpp |
| 클래스 | listFiles.h에 있는 ListFiles를 listFiles.cpp에서 사용한다. |
| 함수명 | setDirectory를 이용해서 BlockAttack 폴더에 있는 파일을 찾고, getFileName은 파일의 이름을 리턴해준다. forward와 back으로 startFileNr을 감소하고 증가시킨다. |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/listFiles.cpp>  <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/listFiles.h> |

|  |  |
| --- | --- |
| 게임 설정 | 설명 |
| 기능 | 게임의 속도, 핸디캡, 커서, 레벨 등 여러 가지를 설정 할 수 있다. |
| 소스파일 | BlockGame.cpp, BlockGame.hpp |
| 클래스 | puzzlehandler.hppBlockGame.hpp에 있는 클래스 BlockGame을 가져와 BlockGame.cpp에서 사용하고 있다. |
| 함수명 | NewStageGame, NewPuzzleGame 등의 함수로 게임을 시작하고 setPlayerWon, setDraw 등의 함수가 플레이어의 승패 여부를 알려준다. |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/BlockGame.cpp>  <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/BlockGame.hpp> |

|  |  |
| --- | --- |
| Ball 설정 | 설명 |
| 기능 | Ball의 속도를 설정하고 Ball의 최대개수의 제한과  Ball의 색상을 설정하는 기능을 수행하고 있다. |
| 소스파일 | Main.cpp |
| 클래스 | Main.cpp에 있는 클래스aBall, Ballmanager, addBall, Ballupdate를 사용 |
| 함수명 | rand함수를 사용하여 Ball의 속도(velocity)값을 설정, Ballmanager 및 Ballupdate로 Ball의 상태를 계속 갱신해준다. |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/main.cpp> |

|  |  |
| --- | --- |
| 레벨 설정 | 설명 |
| 기능 | 사용자(player)의 선택에 따라 puzzle-level을 바꿔주는 기능을 수행 |
| 소스파일 | Main.cpp |
| 클래스 | Main.cpp에 있는 drawIMG, SDL\_Event event, SDL\_Getmousestate, Nfont를 사용 |
| 함수명 | drawIMG를 통한 게임상 puzzle선택화면 구현, SDL\_event와Getmousestate로 키보드 및 마우스 인식, Nfont를 사용해 화면의 글꼴 설정 |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/main.cpp> |

|  |  |
| --- | --- |
| 퍼즐 설정 | 설명 |
| 기능 | level에 따른 puzzle의 개수, 움직임 횟수 등 퍼즐에 관한 것을 설정한다. |
| 소스파일 | puzzleHandler.hpp, puzzleHandler.cpp |
| 클래스 | X |
| 함수명 | 함수는 없고 LoadPuzzleStages 변수 등으로 Level에 따라 퍼즐을 설정한다. |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/puzzlehandler.cpp>  <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/puzzlehandler.hpp> |

|  |  |
| --- | --- |
| 게임 모드 설정 | 설명 |
| 기능 | 사용자(player)가 선택에 따라 원하는 puzzle의 mode를 고를 수 있는 기능을 수행 |
| 소스파일 | Main.cpp |
| 클래스 | X |
| 함수명 | SinglePlayerEndless, SinglePlayerTimeTrial. TwoPlayerTimeTrial, TwoPlayerVs를 사용하여 각각 기본모드, 타임어택, 2명의사용자 타임어택, 대결모드를 구현하였다. 또한 각 싱글모드를 2명의 사용자가 따로 동시에 play가능하다. |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/main.cpp> |

|  |  |
| --- | --- |
| 메뉴 01 | 설명 |
| 기능 | 메뉴와 관련된 기능들을 수행한다. |
| 소스파일 | MenuSystem.cpp, MenuSystem.h |
| 클래스 | common.h, CppSdlImageHolder.hpp, MenuSystem.h에 있는 ButtonGfx,  Button, Menu, FileMenu를 가져와 MenuSystem.cpp에서 사용한다. |
| 함수명 | performClick으로 클릭했을떄 수행하는 것을 알려주고 placeButtons, addButton으로 메뉴 버튼을 추가하고 위치를 지정해준다. |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/MenuSystem.cpp>  <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/MenuSystem.h> |

|  |  |
| --- | --- |
| 메뉴 02 | 설명 |
| 기능 | global.hpp, MenuSystem.h, common.h를 include하여서  controls를 저장하거나 게임타입이나 레벨을 설정해주는 등 메뉴에 있는 설정을 정의해준다. |
| 소스파일 | menudef.cpp |
| 클래스 | Button\_changekey를 이용해 키설정을 바꾸어준다. |
| 함수명 | runGame으로 게임타입과 레벨을 설정해주고, buttonAction~로 메뉴 버튼을 눌렀을 경우에 대한 것을 정의해준다. MainMenu로 메뉴이름을 정해주는 등 메뉴를 정의해준다. |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/menudef.cpp> |

|  |  |
| --- | --- |
| 블록파괴 | 설명 |
| 기능 | 만약 animation이 작동되면 object를 스크린(게임화면)으로부터  제거시키는 기능 및 화면상의 text를 pop-up(끌어올리는) 기능을 수행한다. |
| 소스파일 | Main.cpp |
| 클래스 | Main.cpp에 있는 클래스 explosion, explosionmanager, textmanager, textupdate를 사용 |
| 함수명 | boolean함수를 사용하여 true,false 값 변환 사용, explosion과 text manager를 통한 블록파괴 및 pop-up 역할 수행 |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/main.cpp> |

|  |  |
| --- | --- |
| 조이패드 | 설명 |
| 기능 | 키보드뿐만 아니라 조이패드로도 플레이 할 수 있게 한다. |
| 소스파일 | joypad.h, joypad.cpp |
| 클래스 | joypad.h에 있는 Joypad를 joypad.cpp에서 사용한다. |
| 함수명 | Joypad 함수와 SDL 라이브러리에서 제공하는 함수 등을 사용하여 조이패드로 플레이 할 수 있게 한다. |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/joypad.cpp>  <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/joypad.h> |

|  |  |
| --- | --- |
| 키보드 | 설명 |
| 기능 | SDL 라이브러리에서 제공하는 키보드 기능을 사용한다.  특정 키보드를 누르면 그것을 입력 받아 기능을 지정한 대로 수행한다. |
| 소스파일 | ReadKeyboard.h, ReadKeyboard.cpp |
| 클래스 | ReadKeyboard.h에 있는 ReadKeyboard를 가져와 ReadKeyboard.cpp에서 사용한다. |
| 함수명 | ReadKey를 통해 SDLKey를 입력 받아 true값이면 일을 수행하게 한다. putchar와 removeChar를 이용해서 문자를 입력 받아 입력과 제거를 수행한다. |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/ReadKeyboard.cpp>  <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/ReadKeyboard.h> |

|  |  |
| --- | --- |
| 통계 | 설명 |
| 기능 | 블록 어택 게임의 상단 카테고리 중 Score의 3페이지에 있는 여러 통계들을 나타낸다. |
| 소스파일 | stats.h, stats.cpp |
| 클래스 | stats.h에 있는 클래스 Stats를 stats.cpp에서 사용한다. |
| 함수명 | save, load 등의 함수로 여러 통계들을 저장하고 나타낸다 |
| 참조 | stat은 구어(口語)로 ‘통계량’이라는 뜻 |
|  |  |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/stats.h>  <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/stats.cpp> |

|  |  |
| --- | --- |
| 폰트 01 | 설명 |
| 기능 | SDL 라이브러리에서 제공하는 폰트 기능이다.  이 기능을 쓰면 이미지&글자가 animate, 즉 움직이는 효과를 줄 수 있다.  NFont는 사용자가 어떠한 이미지 포맷을 쓰든 NFont화(이미지&글자를 움직이게 해줌) 시킬 수 있다. |
| 소스파일 | NFont.h, NFont.cpp |
| 클래스 | NFont.h에 있는 클래스 NFont를 가져와 NFont.cpp에서 사용하고 있다. |
| 함수명 | copyString, getPixel 등의 함수를 통해 이미지&글자를 읽어 들이고 draw, set 등의 함수가 이미지&글자를 움직이게 한다. |
| 참조 | Libs 폴더에 소스파일들이 있음. |
| 코드 | <https://github.com/jay47/OSS_Project/tree/master/BlockAttack/source/code/Libs> |

|  |  |
| --- | --- |
| 폰트 02 | 설명 |
| 기능 | TTFONT는 NFont처럼 SDL 라이브러리에서 제공하는 폰트다. |
| 소스파일 | ttfont.h, ttfont.cpp |
| 클래스 | ttfont.h에 있는 클래스 TTFont를 ttfont.cpp에서 사용한다. |
| 함수명 | getTextHeight, getTextWidth 함수 등으로 글자를 받아 writeText 함수로 글자를 출력한다. |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/ttfont.cpp>  <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/ttfont.h> |

|  |  |
| --- | --- |
| 이미지 설정 | 설명 |
| 기능 | 게임의 이미지를 설정해 주는 역할을 한다. |
| 소스파일 | CppSdlImageHolder.cpp, CppSdlImageHolder.hpp |
| 클래스 | CppSdlImageHolder.hpp에 있는 CppSdlImageHolder를 가져와  CppSdlImageHolder.cpp 에서 사용하고 있다. |
| 함수명 | Cut, PaintTo 등의 함수로 이미지를 설정한다. |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/CppSdlImageHolder.cpp>  <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/CppSdlImageHolder.hpp> |

|  |  |
| --- | --- |
| physfs | 설명 |
| 기능 | 이 헤더는 fstream과 유사한 클래스와 typedef를 포함하고 있다.  기능에서도 유사하고 주된 차이점은 namespace가 있지 않다는 것이다. |
| 소스파일 | physfs\_stream.hpp |
| 클래스 | physfs\_filebuf, physfs\_ifstream |
| 함수명 | X |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/physfs_stream.hpp> |
| 참조 | https://icculus.org/physfs/ |

|  |  |
| --- | --- |
| 플랫폼 모음 | 설명 |
| 기능 | 게임을 위한 거의 모든 멀티플랫폼을 포함하고 있다. |
| 소스파일 | common.h, common.cpp |
| 클래스 | common.h에 있는 Config와 TimeHandler를 사용한다.  Config는 설정 변수를 저장할 맵 구성 클래스이다.  TimeHandler는 전체 런타임을 돌려주거나,  밀리 초 단위의 숫자를 받아 commonTme형식으로 반환하거나, 시간을 기록한다. |
| 함수명 | set함수를 이용해서 값을 변경하고 get함수를 이용해서 값을 반환시킨다. |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/common.cpp>  <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/common.h> |

|  |  |
| --- | --- |
| 변수 모음 01 | 설명 |
| 기능 | CppSdlImageHolder.hpp와 Libs/NFont.h에 있는 (외부)변수들을 한 곳에 모아 정리한다. |
| 소스파일 | global.hpp |
| 클래스 | X |
| 함수명 | MainMenu, ResetFullscreen |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/global.hpp> |

|  |  |
| --- | --- |
| 변수모음02 | 설명 |
| 기능 | 주로(Main) 쓰이는 변수(Var)들을 한 곳에 모아 정리한다. |
| 소스파일 | mainVars.hpp |
| 클래스 | X |
| 함수명 | X |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/mainVars.hpp> |

|  |  |
| --- | --- |
| 예외처리 | 설명 |
| 기능 | 예외를 처리하는 기능이다. |
| 소스파일 | CppSdlException.cpp, CppSdlException.hpp |
| 클래스 | CppSdlException.hpp에 있는 CppSdlException를 가져와 CppSdlException.cpp  에서 사용하고 있다. |
| 함수명 | GetErrorNumber, GetSubSystem등으로 예외처리가 가능하다. |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/CppSdlException.cpp>  <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/CppSdlException.hpp> |

# 주요 메소드 분석

|  |  |
| --- | --- |
| 게임 설정 | 설명 |
| 기능 | 게임의 속도, 핸디캡, 커서, 레벨 등 여러 가지를 설정 할 수 있다. |
| 소스파일 | BlockGame.cpp, BlockGame.hpp |
| 클래스 | puzzlehandler.hppBlockGame.hpp에 있는 클래스 BlockGame을 가져와 BlockGame.cpp에서 사용하고 있다. |
| 메소드 | -rand2  : 난수를 지정한다.  -firstUnusedChain  : bool 값을 사용하여 첫번째 chain을 지정한다.  -BlockGame  : blockgame에 사용되는 변수들을 지정하고 for문을 통해 board판을 지정한다.  -setGameSpeed  : globalSpeenLeverl로 게임의 스피드를 지정한다.  -setHandicap  : globalHandicap으로 핸디캡을 적용해준다.  -setAIlevel  : 레벨을 설정한다.  -GetScore  : score 값을 반환한다.  -GetHandicap  : handicap 값을 반환한다.  -isGameOver  : bGameOver 값을 반환한다.  -GetGameStartedAt  : gameStartedAt 값을 반환한다.  -GetGameEndedAt  : gameEndedAfter 값을 반환한다.  -isTimeTrial  : timetrial 값을 반환한다.  -isStageClear  : stageClear 값을 반환한다.  -isVsMode  : vsMode 값을 반환한다.  -isPuzzleMode  : puzzleMode 값을 반환한다.  -GetLinesCleared  : linesCleared 값을 반환한다.  -GetStageClearLimit  : stageClearLimit 값을 반환한다.  -GetChains  : chain 값을 반환한다.  -GetPixels  : pixels 값을 반환한다.  -GetSpeedLevel  : speedLevel 값을 반환한다.  -GetTowerHeight  : TowerHeight 값을 반환한다.  -GetCursorX  : cursorx 값을 반환한다.  -GetCursorY  : cursory 값을 반환한다.  -MoveCursorTo  : cursorx와 cursory의 커서값을 지정해준다.  -GetIsWinner  : hasWonTheGame 값을 반환한다.  -NewGame  : 변수들을 지정하고 생성자 ticks를 지정한다.  -NewTimeTrialGame  : 새로운 timetrial게임을 지정하는 메소드이다.  -NewStageGame  : stage를 클리어하면 새로운 stage가 나타나는 메소드이다.  -NewPuzzleGame  : 새로운 board를 가져와 퍼즐게임을 시작한다.  -retryLevel  : 퍼즐모드면 새로운 퍼즐게임을 하고 stage를 클리어하면 새로운 스테이지로 넘어간다.  -nextLevel  : 다음 레벨로 올라갈 수 있는지 확인해주는 메소드이다.  -NewVsGame  : putStartBlocks메소드를 사용하여 2명의 플레이어가 게임을 할 수 있다.  -setPlayerWon  :if문을 통해 AI\_Enabled을 결정해 승자를 알려주는 기능을 한다.  -setDraw  : AI\_Enabled값이 아니면 비기는 경우를 나타낸다.  -LineEmpty  : bool값이 참이면 블록으로 한 줄이 채워지면 라인을 지워준다.  -BoardEmpty  : bool 값이 참이면 보드판을 비워준다.  -hasStaticContent  : for문을 통하여 조건을 만족시키면 내용물을 채워준다.  -putStartBlocks  : 경우에 따라서 블록이 어떻게 나올지 지정해준다.  -ReduceStuff  : howMuchHang이 0보다 클 경우 stuff를 경우에 따라 줄여준다.  -CreateGarbage  : LineEmpty이거나 startPosition이 29이면 garbage을 생성한다.  -CreateGrayGarbage  : 회색 garbage을 생성한다.  -GarbageClearer  : garbage를 제거하는 역할을 한다.  -GarbageMarker  : garbageToBeCleared 배열로 garbage를 표시해준다  -FirstGarbageMarker  : garbageToBeCleared 배열로 처음의 garbage를 표시해준다.  -ClearBlocks  : for문을 통해 블록들을 조건에 따라 없애준다.  -SetGameOver  : 게임이 끝나는 경우를 알려주는 메소드이다.  -GetAIenabled  : AI\_Enabled 값을 반환한다.  -FallBlock  : board 배열을 통해 블록이 떨어지는 것을 나타내준다.  -GarbageFall  : for문을 통해 조건을 만족하면 garbage가 떨어지는 것을 나타내준다.  -FallDown  : bool값을 false로 지정하고 board배열을 통해 아래로 떨어지는 것을 나타낸다.  -MoveCursor  : cursorx와 cursory로 커서를 움직이게 해준다.  -SwitchAtCursor  : if절에서 board배열에 해당하는 조건이 만족되면 커서를 바꾸게 해준다.  -PushLine  : PushLineInternal 메소드를 받는다.  -PushLineInternal  : 블록이 꽉 차면 한 줄씩 밀려나게 지정해준다.  -PushPixels  : 픽셀을 지정해준다.  -FindTowerHeight  : 가장 높은 곳이 어딘지 찾아준다.  -nrOfType  : for문을 통해 counter값을 증가시킨다..  -horiInLine  : nrOfType을 늘려준다.  -horiClearPossible  : 클리어를 할 수 있도록 도와준다.  -lineHasGarbage  : 라인이 garbage로 찼는지 확인한다.  -nrOfRealTypes  : type을 확인한다.  -ThereIsATower  : 제일 높은 곳을 나타낸다.  -firstInLine1  : 첫 번째 라인을 지정해준다.  -closetTo  : 가까이가게해주는 메소드이다.  -AI\_ClearTower  : 꼭대기를 clear 해준다.  -AI\_ClearHori  : hori를 clear 해준다.  -veriClearPossible  : color나 line을 클리어 가능한지 확인하는 메소드이다.  -closetTo  : 위치를 지정해준다.  -AI\_ClearVertical  : 세로 줄을 클리어 해주는 메소드이다.  -AI\_Move  : AI 를 움직여준다.  -Update  : 업데이트 한 것을 나타내준다.  -IsNearDeath  : 게임이 끝나갈 때를 알려준다.  -UpdateInternal  : internal을 업데이트한다.  -PerformAction  : 액션에 따른 수행 결과들을 알려주는 메소드이다. |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/BlockGame.cpp>  <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/BlockGame.hpp> |

|  |  |
| --- | --- |
| 메뉴 01 | 설명 |
| 기능 | 메뉴와 관련된 기능들을 수행한다. |
| 소스파일 | MenuSystem.cpp, MenuSystem.h |
| 클래스 | common.h, CppSdlImageHolder.hpp, MenuSystem.h에 있는 ButtonGfx,  Button, Menu, FileMenu를 가져와 MenuSystem.cpp에서 사용한다. |
| 메소드 | DrawING - 이미지를 나타내준다.  setSurfaces - surface 의 width 와 height을 지정해준다.  Button - gfx, labal, marked, action 을 지정한다.  setLabel - label을 지정해준다.  setAction - action을 지정해준다.  isClicked - x와 y값이 일치하면 true값을 반환시키고 아니면 false를 반환시킨다.  doAction - 수행이 일어나지 않으면 error문구를 띄어준다.  drawTo - gfx의 값이 나타내는 값에 따라 draw결과를 알려준다.  setPopOnRun - popOnRun 메모리를 할당해서 저장해준다.  isPopOnRun - popOnRun 값을 반환한다.  setGfx - gfx 메모리를 할당해서 저장해준다.  getHeight - ysize 값을 반환한다.  drawSelf - 버튼이 눌렸을 때 처음 화면을 DrawIMG을 통해 스스로 그려진다.  performClick - isClicked와 isPopOnRun 값을 통해 수행하는 메소드이다.  placeButtons - nextY 와 x값을 지정해 버튼의 위치를 나타낸다.  addButton- 버튼을 추가해 placeButtons메소드로 넘어간다.  Menu - 벡터를 통해서 buttons을 나타내고 isSubmenu가 true값을 나타낸다.  Run - event의 type에 따라 게임을 실행시키는 것을 결정하고 만약 마우스 버튼이 눌리면  bMouseUp의 값을 true로 받는다. |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/MenuSystem.cpp>  <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/MenuSystem.h> |

|  |  |
| --- | --- |
| 메뉴 02 | 설명 |
| 기능 | global.hpp, MenuSystem.h, common.h를 include하여서 controls를 저장하거나 게임타입이나 레벨을 설정해주는 등 메뉴에 있는 설정을 정의해준다. |
| 소스파일 | menudef.cpp |
| 클래스 | Button\_changekey를 이용해 키설정을 바꾸어준다. |
| 함수명 | runGame으로 게임타입과 레벨을 설정해주고, buttonAction~로 메뉴 버튼을 눌렀을 경우에 대한 것을 정의해준다. MainMenu로 메뉴이름을 정해주는 등 메뉴를 정의해준다. |
| 메소드 | PrintHi – “Hi”를 출력한다.  getKeyName – 표시하는 키의 이름을 반환해준다.  Button\_changekey – 키의 설정을 바꾸어 준다.  doAction – 사용자가 입력한 키보드의 key대로 버튼을 바꿔준다.  InitMenues – init code이다. Button의 font와 surfaces를 설정한다.  runSinglePlayerEndless – runGame(0,0)으로 게임 모드를 지정한다.  runSinglePlayerTimeTrial – runGame(1,0)으로 게임 모드를 지정한다.  runStageClear – runGame(2,0)으로 지정한다.  runSinglePlayerPuzzle – runGame(3,0)으로 지정한다.  runSinglePlayerVs – runGame(2, b->Generic1)으로 게임 모드를 지정한다.  runTwoPlayerTimeTrial – runGame(10,0)으로 설정한다.  runTwoPlayerVs – runGame(11,0)으로 설정한다.  runHostGame – runGame(12,0)으로 설정한다.  runJoinGame – runGame(13,0)으로 설정한다.  buttonActionMusic – button의 라벨을 설정해준다.  buttonActionSound – sound의 라벨을 설정해준다.  buttonActionFullscreen – fullscreen의 라벨을 지정해준다.  buttonActionPlayer1Name – Player1의 이름을 바꿀 수 있게 한다.  buttonActionPlayer2Name – Player2의 이름을 바꿀 수 있게 한다.  buttonActionPortChange – 네트워크 대결에서 포트번호를 바꿔준다.  buttonActionIpChange – 네트워크 대결에서 IP번호를 바꿔준다.  buttonActionHighscores – 점수 페이지를 열어준다.  ChangeKeysMenu – 화면에 키를 바꾸는 화면을 보여준다.  ConfigureMenu – 메뉴들을 모두 모아 스크린에 보여준다.  SinglePlayerVsMenu – Single Play Vs 모드의 메뉴를 설정해서 버튼을 추가한다.  MultiplayerMenu – mulitplay의 메뉴를 설정하여 버튼을 추가한다.  MainMenu – 메인 메뉴의 라벨과 버튼을 추가해준다. |

|  |  |
| --- | --- |
| 이미지 설정 | 설명 |
| 기능 | 게임의 이미지를 설정해 주는 역할을 한다. |
| 소스파일 | CppSdlImageHolder.cpp, CppSdlImageHolder.hpp |
| 클래스 | CppSdlImageHolder.hpp에 있는 CppSdlImageHolder를 가져와  CppSdlImageHolder.cpp 에서 사용하고 있다. |
| 메소드 | CppSdImageHolder - data나 rw가 없을 경우 에러 메시지를 뜨게 한다. OptimizeForBlit 메소드를 사용한다.  GetRawDataInsecure - Initialized 메소드에서 data값을 반환한다.  GetHeight - IsNull이면 0값을 반환하고 그렇지 않으면 area.h를 반환한다.  Cut - area.x, area.y, area.w, area.h 값들로 크기를 지정한다.  PaintTo - dest.x와 dest.y값을 지정하고 SDL\_BlitSurface메소드를 사용한다.  OptimizeForBlit - allowAlpha면 tmp값을 SDL\_DisplayFormatAlpha로 지정하고 그렇지 않으면 SDL\_DisplayFormat으로 지정한다.  Intalized - data값이 비어있으면 오류 메시지를 보여준다.  IsNull - data나 counter이 비어있으면 true값을 반환하고 아니면 false를 반환한다.  MakeNull - null값을 지정해준다. |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/CppSdlImageHolder.cpp>  <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/CppSdlImageHolder.hpp> |

|  |  |
| --- | --- |
| 예외처리 | 설명 |
| 기능 | 예외를 처리하는 기능이다. |
| 소스파일 | CppSdlException.cpp, CppSdlException.hpp |
| 클래스 | CppSdlException.hpp에 있는 CppSdlException를 가져와 CppSdlException.cpp  에서 사용하고 있다. |
| 메소드 | CppSdlException - 변수들을 지정해준다  What - \_messae.c\_str() 값을 반환한다.  GetErrorNumber - \_errorNumber 값을 반환한다.  GetSubSystem - \_subsystem 값을 반환한다. |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/CppSdlException.cpp>  <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/CppSdlException.hpp> |

|  |  |
| --- | --- |
| Ball 설정 | 설명 |
| 기능 | Ball의 속도를 설정하고 Ball의 최대개수의 제한과  Ball의 색상을 설정하는 기능을 수행하고 있다. |
| 소스파일 | Main.cpp |
| 클래스 | Main.cpp에 있는 클래스aBall, Ballmanager, addBall, Ballupdate를 사용 |
| 함수명 | rand함수를 사용하여 Ball의 속도(velocity)값을 설정, Ballmanager 및 Ballupdate로 Ball의 상태를 계속 갱신해준다. |
| 메소드 | aBall- 생성자 메소드  ~aBall- 소멸자 메소드  getColor- Ball에 색상을 입혀준다.  getX- X축, 가로의 길이를 설정해준다.  getY- Y축, 세로의 길이를 설정해준다.  update- Ball의 현재 상태를 확인하고 갱신해준다.  addBall- 필요에 따라 Ball을 증가시켜준다.  ballManager- Ball의 속력과 상태를 관리한다. |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/main.cpp> |

|  |  |
| --- | --- |
| 레벨 설정 | 설명 |
| 기능 | 사용자(player)의 선택에 따라 puzzle-level을 바꿔주는 기능을 수행 |
| 소스파일 | Main.cpp |
| 클래스 | Main.cpp에 있는 drawIMG, SDL\_Event event, SDL\_Getmousestate, Nfont를 사용 |
| 함수명 | drawIMG를 통한 게임상 puzzle선택화면 구현, SDL\_event와Getmousestate로 키보드 및 마우스 인식, Nfont를 사용해 화면의 글꼴 설정 |
| 메소드 | drawIMG- 게임창의 크기와 초기화면을 구현해준다.  getmousestate- 프로그램이 마우스를 인식할 수 있도록 해준다.  getkeyboardstate- 프로그램이 키보드를 인식할 수 있도록 해준다.  Nfont - 게임창의 메뉴 및 화면에서 표시되는 글꼴의 모양 색상을 설정해준다.  button- 어떤 키를 사용할 지 설정 해준다. |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/main.cpp> |

|  |  |
| --- | --- |
| 블록파괴 | 설명 |
| 기능 | 만약 animation이 작동되면 object를 스크린(게임화면)으로부터  제거시키는 기능 및 화면상의 text를 pop-up(끌어올리는) 기능을 수행한다. |
| 소스파일 | Main.cpp |
| 클래스 | Main.cpp에 있는 클래스 explosion, explosionmanager, textmanager, textupdate를 사용 |
| 함수명 | boolean함수를 사용하여 true,false 값 변환 사용, explosion과 text manager를 통한 블록파괴 및 pop-up 역할 수행 |
| 메소드 | anExplosion- 폭발animation 생성자 메소드  ~anExplosion- 폭발animation 소멸자 메소드  getFrame- animation 프레임을 설정해준다.  getX- animation의 X축, 가로값을 설정해준다.  getY- animation의 Y축, 세로값을 설정해준다.  remove- animation효과를 제거해준다.  addText- 게임상의 조건에 따른 Text를 추가 시킨다.  textManager- Text을 추가할 지 아닐지를 결정해준다  update- 게임상의 Text의 현재 상태를 확인하고 갱신해준다.  addExplosion- Ball이 조건에 만족하게되면 폭발시켜주게된다.  ExplosionManager- 폭발을 발생시킬지 아닐 지를 결정해준다.  update- Ball의 상태를 확인하여 폭발의 여부를 확인해준다. |
| 코드 | <https://github.com/jay47/OSS_Project/blob/master/BlockAttack/source/code/main.cpp> |

|  |  |
| --- | --- |
| 폰트 01 | 설명 |
| 기능 | SDL 라이브러리에서 제공하는 폰트 기능이다.  이 기능을 쓰면 이미지&글자가 animate, 즉 움직이는 효과를 줄 수 있다.  NFont는 사용자가 어떠한 이미지 포맷을 쓰든 NFont화(이미지&글자를 움직이게 해줌) 시킬 수 있다. |
| 소스파일 | NFont.h, NFont.cpp |
| 클래스 | NFont.h에 있는 클래스 NFont를 가져와 NFont.cpp에서 사용하고 있다. |
| 메소드 | ･setAnimData - userVar변수에 data 값 넣기  ･setBuffer - 버퍼 값 초기화 시키고 새로운 값 받기  ･copyString - 문자 복사  ･getPixel - x축 y축 값 얻기  ･rectUnion - x축, y축의 최소값, 최대값을 구하고 그 둘의 차를 리턴한다.  ･verticalGradient - 수직 경사도를 구하는 메소드  ･init - height, baseline 등 각종 변수들을 0으로 초기화  ･load - 데이터 값을 로드한다. 만약 값이 없다면 Unable to load TrueType font, Unable to initialize SDL\_ttf 등의 문장을 출력한다.  ･drawToSurface - 늘어난 글자들을 올바른 인덱스로 옮긴다.  ･drawToSurfacePos - 최종 x축, y축 값을 얻기 위한 메소드다.  ･draw - x축, y축 값을 받아 drawToSurface 메소드로 리턴한다.  ･drawPos - drawToSurfacePos로 리턴한다.  ･drawAll - 움직이는 효과를 start, print 그리고 end 한다.  ･getDest - dest 접근자 선언  ･getSurface - src 접근자 선언  ･getHeight - 높이 값을 구한다  ･getWidth - 너비 값을 구한다.  ･getAscent - getPixel 메소드로 최고점을 구한다.  ･getDescent - getPixel 메소드로 최저점을 구한다.  ･getSpacing - letterSpacing 접근자 선언  ･getLineSpacing - lineSpacing 접근자 선언  ･getBaseline - baseline 접근자 선언  ･getMaxWidth - maxWidth 접근자 선언  ･setSpacing - letterSpacing 설정자 선언  ･setLineSpacing - lineSpacing 설정자 선언  ･setDest - dest의 설정자 선언  ･setBaseline - baseline의 설정자 선언 |
| 참조 | Libs 폴더에 소스파일들이 있음. |
| 코드 | <https://github.com/jay47/OSS_Project/tree/master/BlockAttack/source/code/Libs> |

|  |  |
| --- | --- |
| 폰트 02 | 설명 |
| 기능 | NFont처럼 SDL 라이브러리에서 제공하는 폰트다. |
| 소스파일 | ttfont.h, ttfont.cpp |
| 클래스 | ttfont.h에 있는 클래스 TTFont를 ttfont.cpp에서 사용한다. |
| 함수명 | getTextHeight, getTextWidth 함수 등으로 글자를 받아 writeText 함수로 글자를 출력한다. |
| 메소드 | TTFont – TTF를 init해주고, 비어있다면 비어있다는 메시지를 출력해준다  ~TTFont – TTFont를 닫게 되면 ‘Closing a font’를 출력한다.  getTextHeight – font height를 리턴해준다.  getTextWidth – font width를 리턴해준다.  writeText - RGB(255,255,255) 중에서 원하는 색깔로 글씨를 쓸 수도 있게 한다. 만약 에러가 나면 ‘Error writing text:’를 출력한다. |

|  |  |
| --- | --- |
| 키보드 | 설명 |
| 기능 | SDL 라이브러리에서 제공하는 키보드 기능을 사용한다. 특정 키보드를 누르면 그것을 입력 받아 기능을 지정한 대로 수행한다. |
| 소스파일 | ReadKeyboard.h, ReadKeyboard.cpp |
| 클래스 | ReadKeyboard.h에 있는 ReadKeyboard를 가져와 ReadKeyboard.cpp에서 사용한다. |
| 함수명 | ReadKey를 통해 SDLKey를 입력받아 true값이면 일을 수행하게 한다. putchar와 removeChar를 이용해서 문자를 입력 받아 입력과 제거를 수행한다. |
| 메소드 | ReadKeyBoard(void) – length, maxLength, position, strcpy를 지정한다.  ReadKeyBoard(const char \*oldName) – character에 textstring을 지정한다.  CharsBeforeCursor – position을 return해서 커서의 위치를 반환한다.  ReadKey – keyPressed로 입력되는 키보드와 비교 후, 입력 된 키에 따라서, putchar를 지정하거나, position을 지정한다. key가 accepted되면 true를 반환한다.  putchar – char타입의 thing을 바당서 textstring에 지정해준다.  removeChar – position에 따라 index를 옮겨서 문자를 제거한다. |

|  |  |
| --- | --- |
| 플랫폼 모음 | 설명 |
| 기능 | 게임을 위한 거의 모든 멀티플랫폼을 포함하고 있다. |
| 소스파일 | common.h, common.cpp |
| 클래스 | common.h에 있는 Config와 TimeHandler를 사용한다. Config는 설정 변수를 저장할 맵 구성 클래스이다. TimeHandler는 전체 런타임을 돌려주거나, 밀리 초 단위의 숫자를 받아 commonTme형식으로 반환하거나, 시간을 기록한다. |
| 함수명 | set함수를 이용해서 값을 변경하고 get함수를 이용해서 값을 반환시킨다. |
| 메소드 | itoa, double2str – int,double형의 num을 받아서 문자열로 반환해준다.  str2double – 문자열을 구문 분석하고 문자열의 값을 double로 반환해준다. 만약 string이 double이 아니라면 0.0을 return해준다. 이러한 방법으로 사용 가능한 값을 항상 반환해준다.  str2int – 문자열을 구문 분석하고 문자열의 값을 int로 반환해준다.  만약 string이 int가 아니라면 0을 return해준다. 이러한 방법으로 사용 가능한 값을 항상 반환해준다.  getMyDocumentsPath - windows에서 my Documents의 경로를 찾아 알려준다.  getPathToSaveFiles – 모든 settings가 저장되어야 될 곳의 경로를 알려준다. 유닉스 계열 시스템은 ~/.gamesaves/GAMENAME가 홈 폴더이다. Windows에서는 MyDocuments/My Games이다.  peekTime – total runtime을 return해준다. 이 때, config file에는 작성하지 않는다.  addTime – peekTime과 같은 기능을 하지만 config file에 작성을 한다.  Config – load, configMap을 clear하고, shuttingDown을 0으로 지정한다.  load – 세이브 파일 경로 중에서 /configFile을 불러온다.  getInstance – instance가 0이면 새로운 Config 객체를 만들고 아니면 instance를 return해준다.  save – config가 disk에 쓰여지도록 한다. 프로그램이 정상적으로 종료 되었으면 발생한다.  exists – key varName이 존재하면, true를 리턴해준다. configMap.find를 사용해서 발견되지 않는다면, Map의 끝 iterator를 리턴해준다.  setDefault – 변수에 대해서 default 값을 설정해준다.  setShuttiongDown – 사용자가 프로그램의 shutdown을 요청하면 설정된다.  isShuttingDown – shuttingDown을 리턴해준다.  setString – config 변수의 varName 키를 가지고 있는 곳에 값을 설정해준다.  setInt – config 에서 varName 키를 가지고 있는 변수에 content라는 스트링값을 int로 변환한 값을 설정해준다.  setValue - config에서 varName 키를 가지고 있는 변수에 content라는 스트링 값을 double로 변환한 값을 설정해준다.  getInt – varName이 존재하면 configMap[varName]을 int로 변환된 값을 리턴해주고 아니면 0을 반환한다.  getValue - varName이 존재하면 configMap[varName]을 double로 변환된 값을 리턴해주고 아니면 0을 반환한다. |

|  |  |
| --- | --- |
| 에디터 | 설명 |
| 기능 | 에디터 기능(예를 들어 보드판 조정)등이 있다. |
| 소스파일 | TheBoard.hpp, TheBoard.cpp, BoardHolder.hpp, BoardHolder.cpp  EditorInterface.hpp, EditorInterface.cpp , editorMain.hpp, editorMain.cpp |
| 역학관계 | TheBoard→ BoardHolder→ EditorInterface→ editorMain 순으로 영향을 준다. |
| 클래스 | TheBoard의 클래스 TheBoard,  BoardHolder의 클래스 BoardHolder,  EditorInterface의 클래스 EditorInterface,  editorMain의 클래스 aButton, theEditor |
| 메소드 | ･saveBoards - 보드판을 세이브한다.  ･getModel - theBoards[nr]의 접근자  ･setModel - theBoards[nr]의 생성자  ･getNumberOfBoards - theBoards.size 접근자  ･addBoard - theBoards.size가 49 미만이면 false로 반환한다.  ･removeBoard - 보드판 제거  ･moveBoardBack - 보드판을 back으로 움직이는 건데 실제 게임에 반영 되지 않았음.  ･moveBoardForward - 보드판을 forward로 움직이는 건데 실제 게임에 반영 되지 않았음.  ･ selectColor - int color를 통해 colorSelected의 값을 설정한다.  ･ getSelectedColor - colorSelected의 접근자 선언  ･ drawOnModel - colorSelected 값에 따라 블록을 set할지 up할지 down 할지를 정한다.  ･ saveFile - 에디터 변경 후 세이브  ･ openFile - 에디터 한 파일을 오픈한다.  ･ newFile - 새로운 파일을 만든다.  ･ getActiveBoardNr - boardActive의 접근자  ･ getNumberOfMoves - tb.getNumberOfMoves의 접근자  ･ newBoard - file 세이브가 실패하면 보드를 생성한다.  ･getNumberOfBoards - bh.getNumberOfBoards의 접근자  ･isSaved - saved의 접근자  ･fileIsSaved - fileSaved의 접근자  ･editorMain.cpp에서 각 클래스의 Button&click - 버튼생성과 클릭을 하면 해당 클래스가 작동하는 메소드다.  ･copyFrom - 보드판의 너비와 높이를 카피한다.  ･setBrick - 블록이 보드판 크기보다 더 많이 들어오지 못하게 한다.  ･getBrick - 블록값을 리턴한다.  ･moveUp - 블록이 한 칸씩 위로 올라오게 한다.  ･moveDown - 블록을 한 칸씩 삭제하도록 한다.  ･moveRight - 블록들을 한 칸씩 오른쪽으로 가게 한다.  ･moveLeft - 블록들을 한 칸씩 왼쪽으로 가게 한다.  ･equals - 보드판 크기와 블록 개수가 맞도록 조정한다. |
| 참조 | editor 폴더에 소스파일들이 있음, 본 게임에서는 적용되지 않았음. |
| 코드 | <https://github.com/jay47/OSS_Project/tree/master/BlockAttack/source/code/editor> |

|  |  |
| --- | --- |
| 게임스코어 | 설명 |
| 기능 | 파일명을 지정하여 점수를 저장해주고, 저장된 점수를 확인하여 등수를 정해준다. |
| 소스파일 | highscore.h, highscore.cpp |
| 클래스 | highscore.h에 있는 Highscore를 highscore.cpp에서 사용한다. |
| 함수명 | isHighScore는 저장된 점수가 가장 높은 점수인지 확인한다. addScore는 점수를 비교하여 등수를 정해준다. getScoreNumber와 getScoreName은 점수와 이름을 반환한다. |
| 메소드 | getMyDocumentsPath1 - windows에서 my Documents의 경로를 리턴해준다.  Highscore – unix와 window인 경우, 그 외의 경우에 따라 파일이름에 경로를 맞게 설정해준다.  writeFile – unix와 window인 경우, 그 외의 경우에 따라 파일이름에 경로를 맞게 설정해준다.  isHighScore – 새로운 점수가 원래 있던 점수보다 높은지 아닌지에 따라 true와 false를 반환해준다.  addScore – 새로운 점수를 랭킹에 등록해준다.  getScoreNumber – index로 room을 받아서 해당되는 점수를 return해준다.  getScoreName – index를 받아서 해당되는 이름을 반환해준다. |

|  |  |
| --- | --- |
| 조이패드 | 설명 |
| 기능 | 키보드뿐만 아니라 조이패드로도 플레이 할 수 있게 한다. |
| 소스파일 | joypad.h, joypad.cpp |
| 클래스 | joypad.h에 있는 Joypad를 joypad.cpp에서 사용한다. |
| 함수명 | Joypad 함수와 SDL 라이브러리에서 제공하는 함수 등을 사용하여 조이패드로 플레이 할 수 있게 한다. |
| 메소드 | Joypad\_init - init하는 code를 포함한다. 0일 경우 return을 해주고 아니면 false를 반환한다.  Joypad\_getStatus – joypad의 status를 반환해준다.  JoyPad – 등록할 수 있는 조이패드 버튼을 제한한다. 사용자가 아무리 많은 버튼을 갖고 있는 조이패드라도 전부 다 사용할 수는 없다.  ~JoyPad- working중 이라면 JoyStrick를 닫는다.  update - 이미 등록한 조이패드 버튼을 수정할 수 있게 한다. |

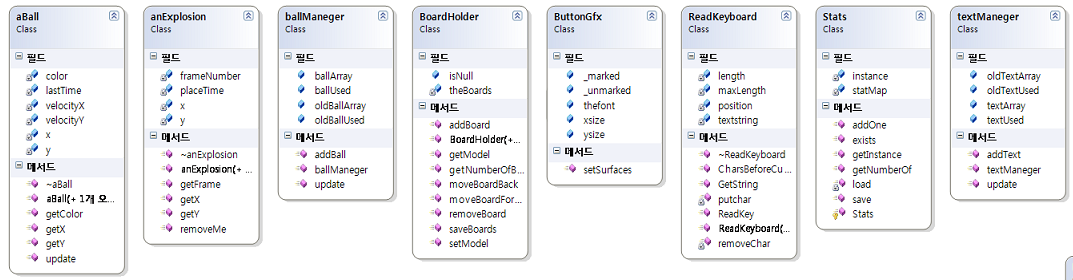
|  |  |
| --- | --- |
| 세이브&로드 | 설명 |
| 기능 | Save와 Load를 할 수 있게 해준다. |
| 소스파일 | listFiles.h, listFiles.cpp |
| 클래스 | listFiles.h에 있는 ListFiles를 listFiles.cpp에서 사용한다. |
| 함수명 | setDirectory를 이용해서 BlockAttack 폴더에 있는 파일을 찾고, getFileName은 파일의 이름을 리턴해준다. forward와 back으로 startFileNr을 감소하고 증가시킨다. |
| 메소드 | setDirectory – 첫 번째 파일을 BlockAttack folder에서 찾는다.  isInList – list에 있는지 확인한다.  setDirectory2 – 두 번째 파일도 setDirectory처럼 역시 찾는다.  getFileName – startFile의 번호에 입력받은 nr을 더한 것이 file의 최대수보다 작다면, filenames를 리턴해준다. 아니라면 빈 String을 반환한다.  fileExists – emptyString을 정해서 filenames가 비어있다면 false, 아니라면 true를 리턴한다.  back – startFile의 번호가 첫 파일보다 크다면 -10을 해주고, 첫 파일보다 작다면, 그것을 첫 파일로 지정해준다.  forward – startFileNr을 10만큼 증가 시킨다.  getRandom – file의 번호를 임의로 선택한다. |

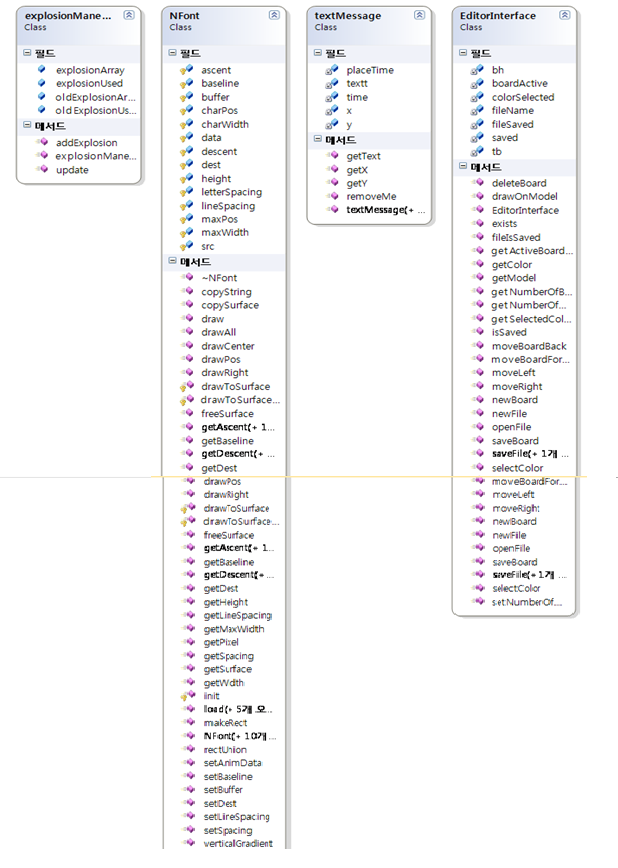
|  |  |
| --- | --- |
| 퍼즐 설정 | 설명 |
| 기능 | Level에 따른 puzzle의 개수, 움직임 횟수 등 퍼즐에 관한 것을 세팅한다. |
| 소스파일 | puzzleHandler.hpp, puzzleHandler.cpp |
| 클래스 | X |
| 함수명 | 함수는 없고 LoadPuzzleStages 변수 등으로 Level에 따른 퍼즐을 세팅한다. |
| 메소드 | PuzzleNumberOfMovesAllowed – 퍼즐 움직임의 횟수를 레벨에 따라 반환해준다.  PuzzleGetBrick – 퍼즐 레벨을 반환해준다.  PuzzleGetNumberOfPuzzles – 퍼즐이 몇 개 있는지 반환해준다.  PuzzleIsCleared – 레벨에 맞게 Puzzle이 Clear 되었는지 반환해준다.  PuzzleGetName – PuzzleName(FileName)을 반환한다.  PuzzleSetName – 이름을 받아서 set 해준다.  PuzzleSavePath – file의 경로를 받아서 설정해준다.  PuzzleSetClear – 퍼즐 레벨에 따라 세이브 한 퍼즐들을 불러온다. 단 에러가 나면 ‘Error writing to file: 퍼즐 세이브 경로’ 를 출력한다.  LoadPuzzleStages – 모든 퍼즐 레벨을 로드해준다. |

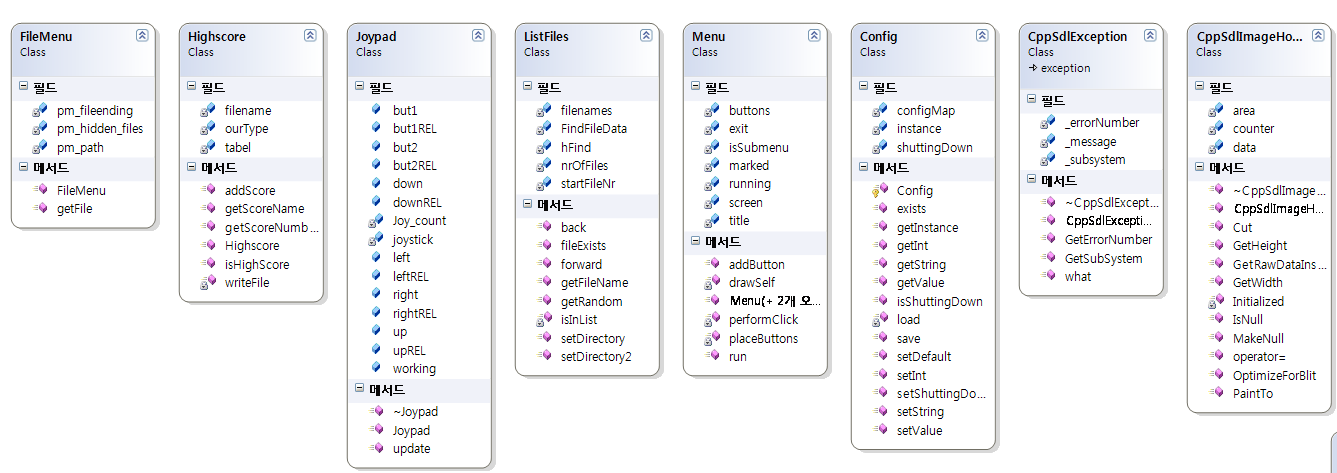
|  |  |
| --- | --- |
| 통계 | 설명 |
| 기능 | 블록 어택 게임의 상단 카테고리 중 Score의 3페이지에 있는 여러 통계들을 나타내는 역할을 한다. |
| 소스파일 | stats.h, stats.cpp |
| 클래스 | stats.h에 있는 클래스 Stats를 stats.cpp에서 사용한다. |
| 함수명 | save, load 등의 함수로 여러 통계들을 저장하고 나타내준다. |
| 참조 | stat은 구어(口語)로 ‘통계량’이라는 뜻이다. |
| 메소드 | Stats – 통계 맵을 호출한다.  load – 통계들을 나타내준다.  getInstance – instance가 비어있으면 stats객체를 새로 만들어 instance를 return 해준다.  save – 통계들을 저장해준다.  getNumberOf – 통계 이름이 존재한다면 그 이름으로 통계 맵 이름을 정해준다.  addOne – 통계 맵보다 글자수가 많아지만 통계 맵을 하나 더 만든다.  exists – 발견되지 않는 경우 ‘find’를 사용하면 map의 끝iterator를 리턴해준다. |

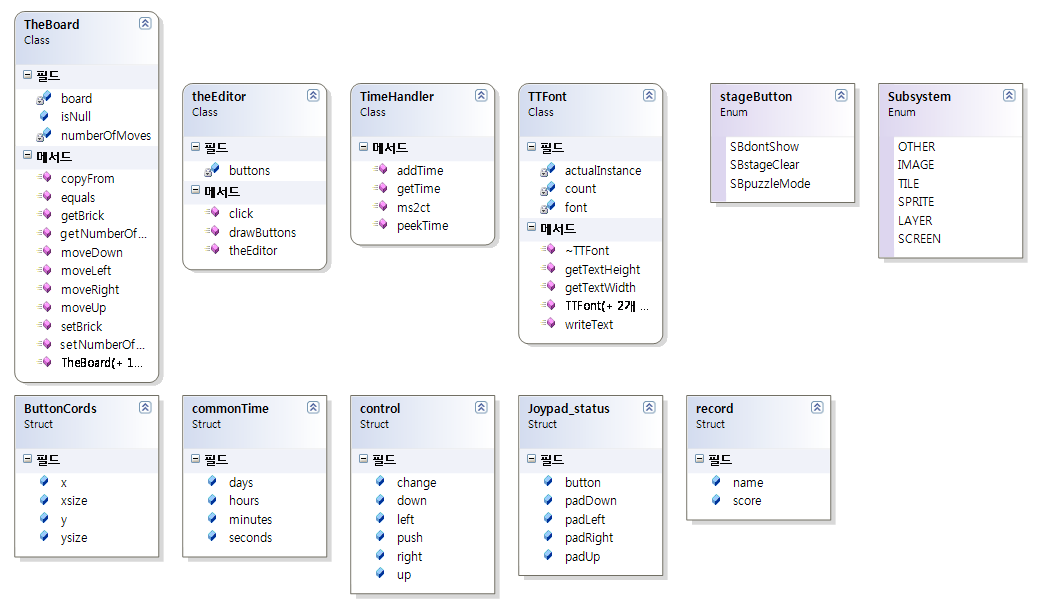
# UML 다이어그램

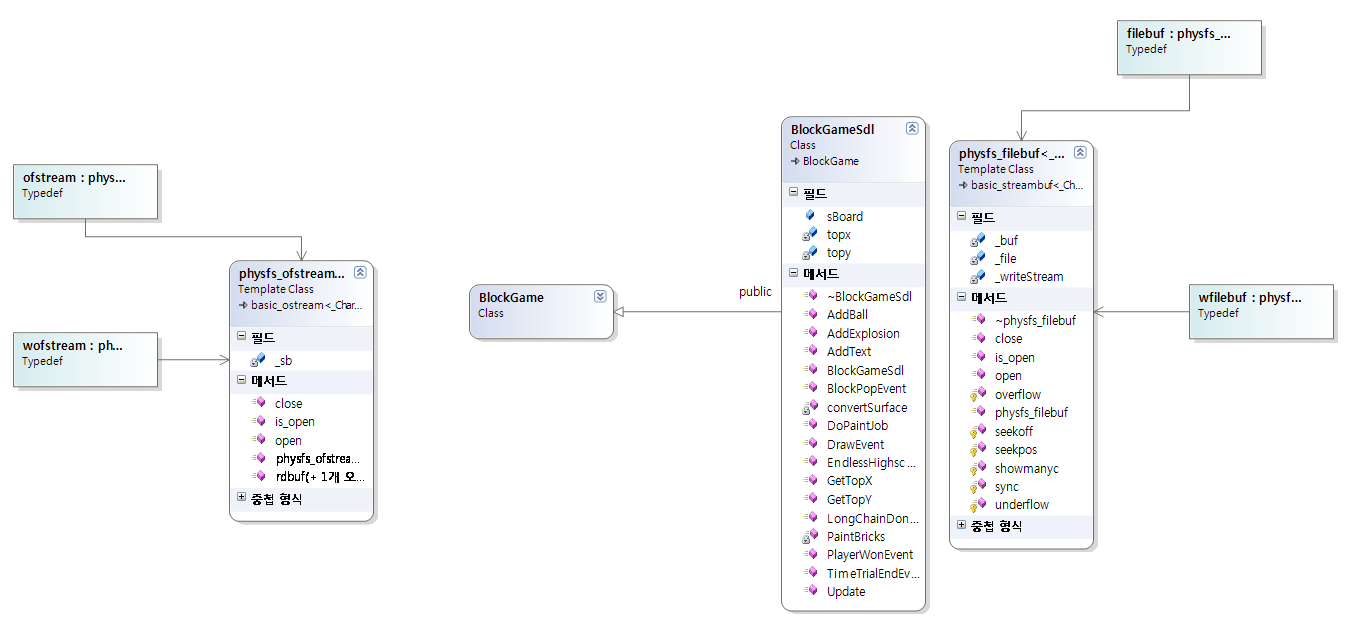


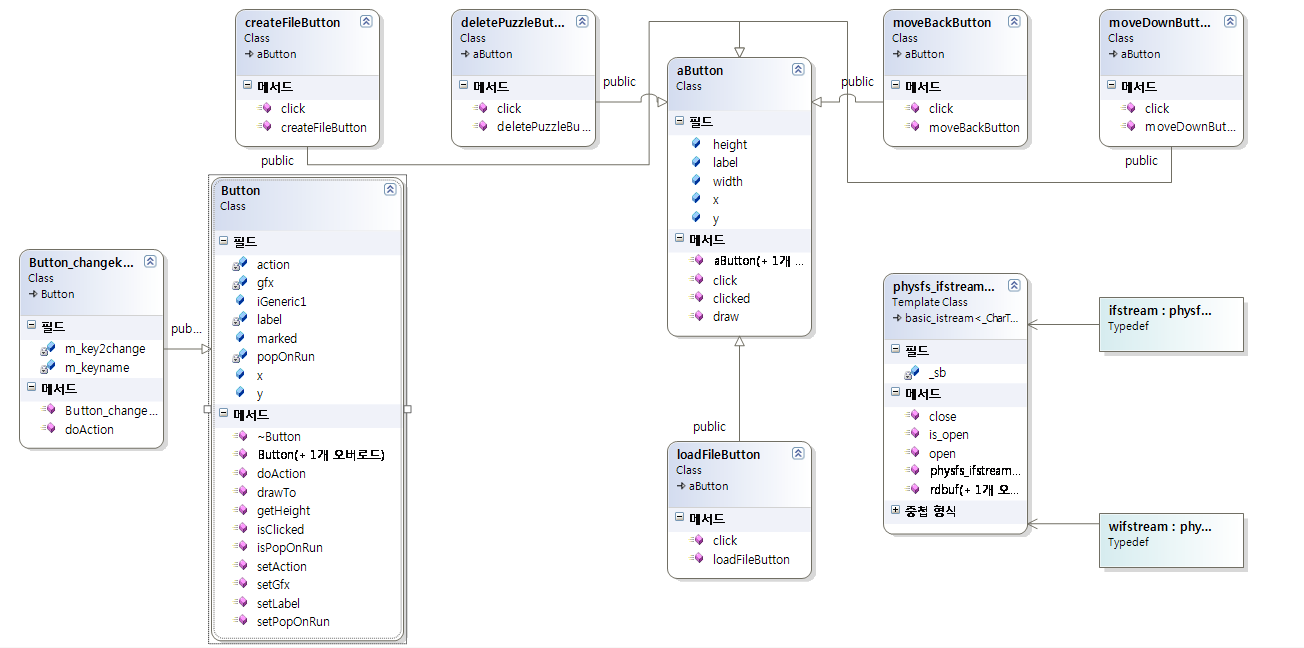










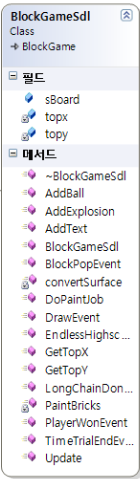


# 주요 사용자 인터페이스 (ui)

|  |  |
| --- | --- |
| C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\background.png | C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\BackBoard.png |
| background  ▶게임전체화면에서 배경에 해당된다. | **BackBoard**  ▶게임창에서 배경에 해당된다. |
| C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\animations\bomb\bomb_1.png | C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\animations\cursor\1.png |
| bomb  ▶폭탄을 나타낸다. | **cursor**  ▶블록을 선택하는 커서이다. |
| C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\animations\explosion\0.png C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\animations\explosion\1.png | C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\animations\ready\ready_1.png C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\animations\ready\ready_2.png |
| explosion  ▶폭탄이 터지는 것을 나타낸다. | **ready**  ▶게임의 준비를 나타낸다. |
| C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\bricks\blue.png C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\bricks\green.png C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\bricks\grey.png C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\bricks\red.png C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\bricks\turkish.png C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\bricks\yellow.png C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\bricks\purple.png | C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\smileys\1.png C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\smileys\2.png C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\smileys\3.png C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\smileys\0.png |
| bricks  ▶게임에 사용되는 블록이다. | **smileys**  ▶게임 결과에 따라 나타내는 이모티콘이다. |
| C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\options.png | C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\boardBackBack.png |
| Options  ▶게임의 옵션을 선택할 수 있는 화면이다. | **boardBackBoard**  ▶BackBoard UI 뒤에 사용되는 화면이다. |
| C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\balls\ballBlue.pngC:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\balls\ballGray.pngC:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\balls\ballGreen.pngC:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\balls\ballPurple.pngC:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\balls\ballRed.pngC:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\balls\ballTurkish.pngC:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\balls\ballYellow.png | C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\counter\1.pngC:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\counter\2.pngC:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\counter\3.png |
| balls  ▶게임에 사용되는 ball이다. | **counter**  ▶카운트를 셀 때 나타난다. |
| C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\bNewGame.png C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\bTwoPlayers.png  C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\bOnePlayer.png C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\bNetwork.png | C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\bOptions.png C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\bTimeTrial.png  C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\bEndless.png C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\bPuzzle.png |
| bNewGame, bOnePlayer, bTwoPlayers, bNetWork  ▶플레이어를 결정하기 위해 사용되는 UI이다. | **bOptions, bEndless, bTimeTrial, bPuzzle**  ▶옵션을 선택할 때 나타나는 UI이다. |
| C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\bExit.png | C:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\iWinner.pngC:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\iLoser.pngC:\Users\김여진\Desktop\2학년 2학기\공개SW실무\프로젝트\각종 UI\iDraw.png |
| bExit  ▶게임을 끝내기 위한 UI이다. | **iWinner, iLoser, iDraw**  ▶게임의 결과를 알려준다. |

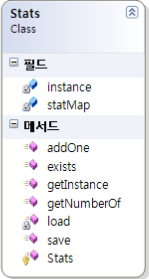
# 개선점

**1. 현재는 블록의 위치를 좌우로만 바꿀 수 있는데 이것을 상하&대각선으로도 바꿀 수 있게 한다.**



블록의 커서 방향을 추가하기 위해 BlockGameSdl 클래스 메소드에 좌우 커서 및 상하&대각선으로도 바꿀 수 있는 메소드를 추가한다.

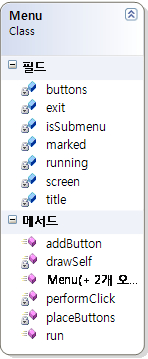
**2. 일시 정지 버튼 생성**



게임 내의 상태를 관리하는 Stats클래스의 하위메소드에 load, save 기능 말고도

중간에 게임을 멈출 수 있는 pause메소드를 추가하여 일시 정지 할 수 있도록 한다.

**3. 현재는 메인 화면에서 ESC키를 누르면 바로 게임이 종료되는데 종료 전에 ‘종료하십니까? Yes/No’ 팝업 띄우기**



Menu클래스 필드에 속해있는exit클래스에 키보드 ESC버튼 클릭시 팝업창을 띄우는 명령어를 추가시킨다.

1. http://www.hani.co.kr/arti/economy/economy\_general/710003.html [↑](#footnote-ref-1)
2. http://www.gamemeca.com/news/view.php?gid=913647 [↑](#footnote-ref-2)
3. http://sourceforge.net/projects/blockattack/?source=directory [↑](#footnote-ref-3)
4. https://ko.wikipedia.org/wiki/%EC%95%A0%EB%8B%88%ED%8C%A1 [↑](#footnote-ref-4)
5. https://en.wikipedia.org/wiki/Candy\_Crush\_Saga [↑](#footnote-ref-5)
6. https://ko.wikipedia.org/wiki/%EC%8A%A4%EC%BC%80%EC%9D%BC%EB%9F%AC%EB%B8%94\_%EB%B2%A1%ED%84%B0\_%EA%B7%B8%EB%9E%98%ED%94%BD%EC%8A%A4 [↑](#footnote-ref-6)