



서경대학교

객체지향프로그래밍 (CS1149 - 03)

PUSH PUSH 게임 최종 보고서

2019316034 소프트웨어학과 정지홍

# ---목차---

- ▶ 개발 시스템 목표
- ▶ 유사 게임 벤치 마킹
- ▶ 주요 기능 및 차별화 요소
- ▶ 개발 일정
- ▶ 소스코드의 자료구조 및 로직
- ▶ 시스템 구성 파일 설명

# ▶▶개발 시스템 목표◀◀

- ❖ 단순한 게임이 아닌 두뇌를 사용하는 퍼즐형 두뇌 게임을 목표로 설정
- ❖ 흥미도를 높이기 위한 사운드를 추가하여 몰입도를 플레이어의 몰입도 증가
- ❖ 각 맵마다 여러 기능을 주로 사용해야만 클리어 가능하게 설정
- ❖ 누구나 쉽게 즐길 수 있는 복잡하지 않은 조작법을 목표

# ▶▶유사 게임 벤치마킹◀◀

## 1. 굴착소년 쿵(플래시 게임)

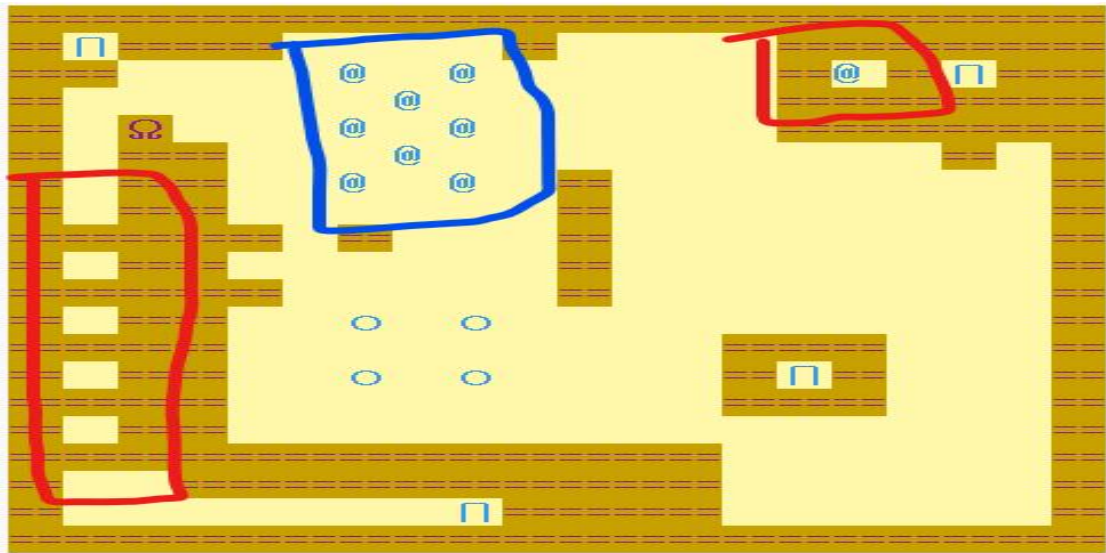


-땅을 파고 내려가는 방식 및 폭탄 아이템 벤치마킹



<그림1>

<그림2>



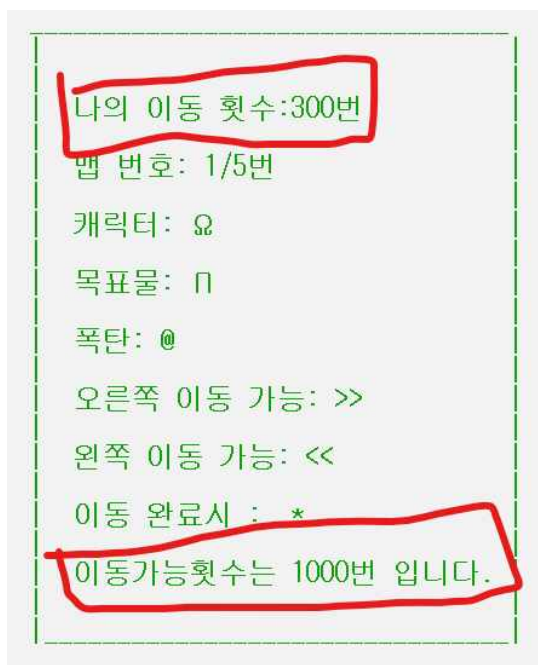
(그림 1의 땅파고 내려 가는 방식을 빨간 박스 구간)

(그림 2의 폭탄으로 특정 방해물 제거하는 방식)

## 2.카카오 프렌즈 타운(스마트폰 게임)



-이동 횟수에 제한을 두는 기능 벤치마킹



## ▶▶주요 기능 및 차별화 요소◀◀

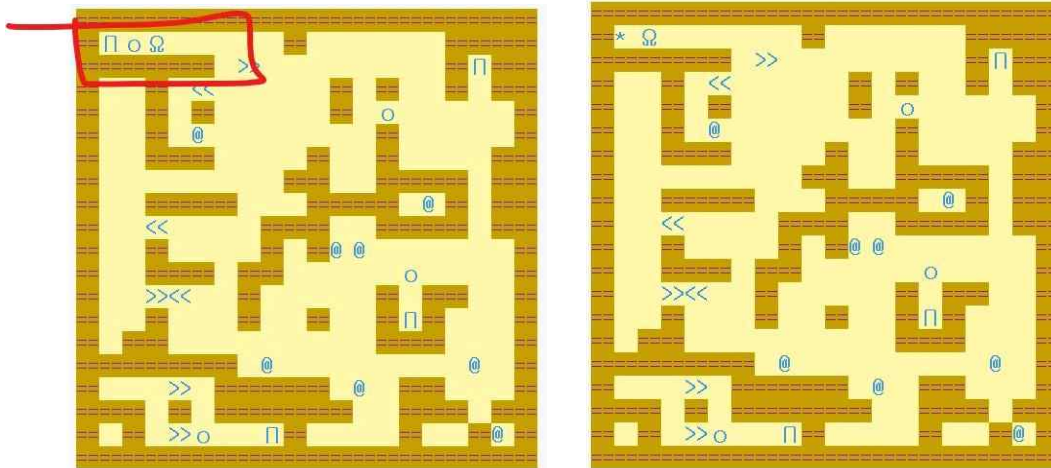
- ✔ 캐릭터 이동시에 사운드
- ✔ @(폭탄)충돌시 장애물 제거 및 사운드
- ✔ 폭탄과 목표물 충돌시 게임 종료 (그림a)
- ✔ 목표물 골인시 \*로 목표물 변경(그림 b,c)
- ✔ 이동가능 횟수 초과시 게임종료(그림 d,e)
- ✔ 공끼리 충돌시 이동 불가
- ✔ 게임 클리어시 게임 종료(그림f)
- ✔ >>,<<의 단방향 이동가능 구간(그림g)
- ✔ 맵 클리어시 사운드 효과



## <그림 a>

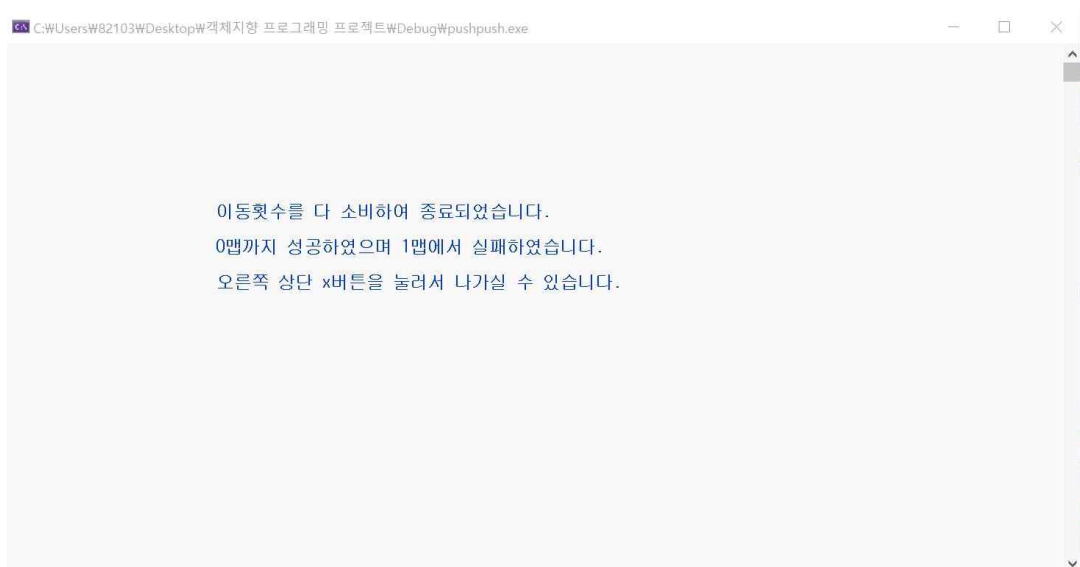
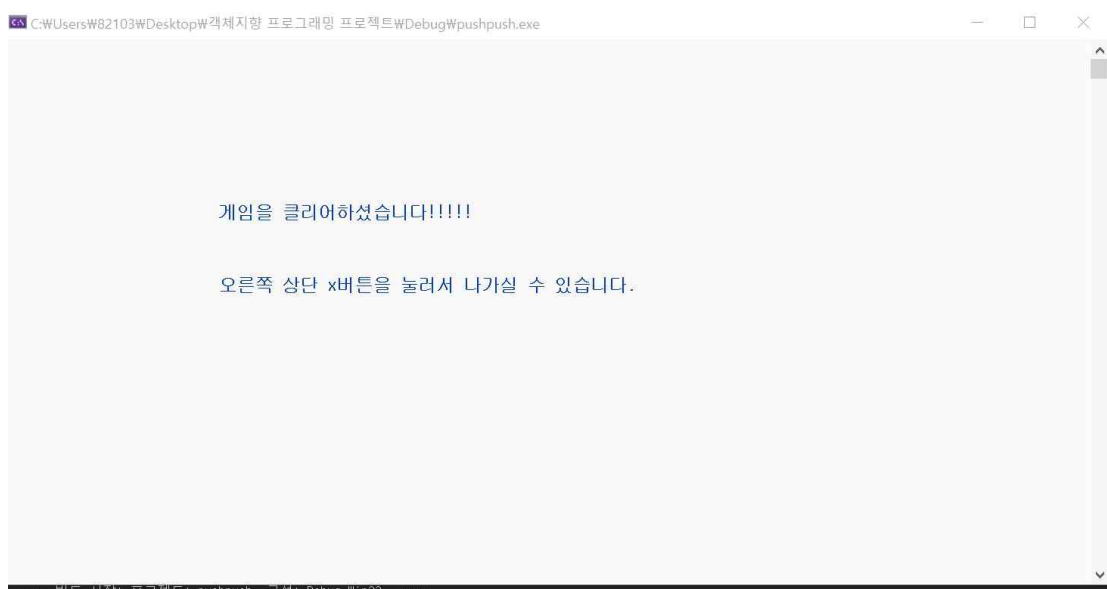


## <그림 b,c>

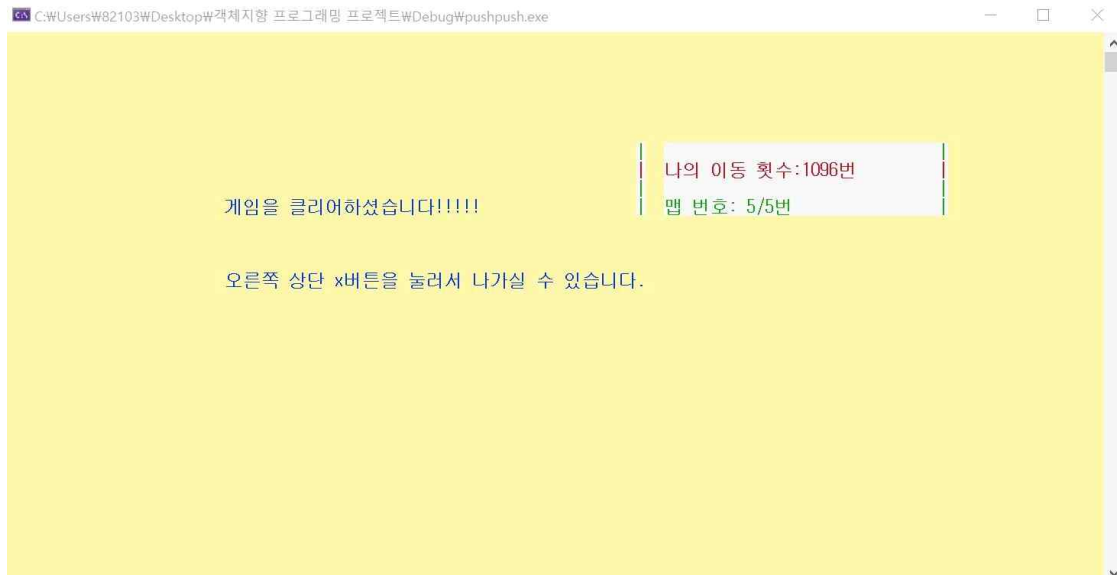




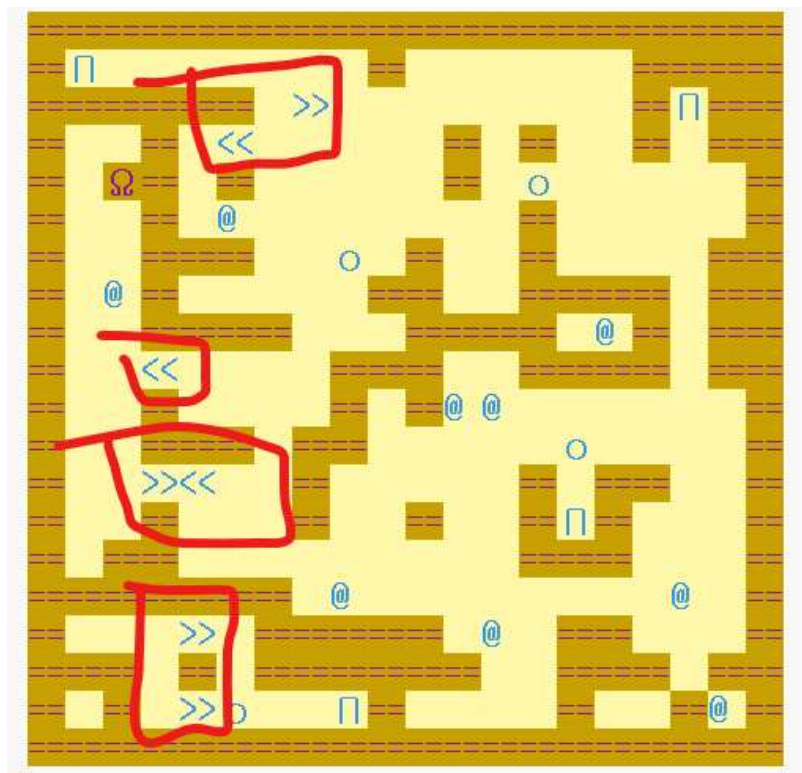
## <그림 d,e>



## <그림 f>



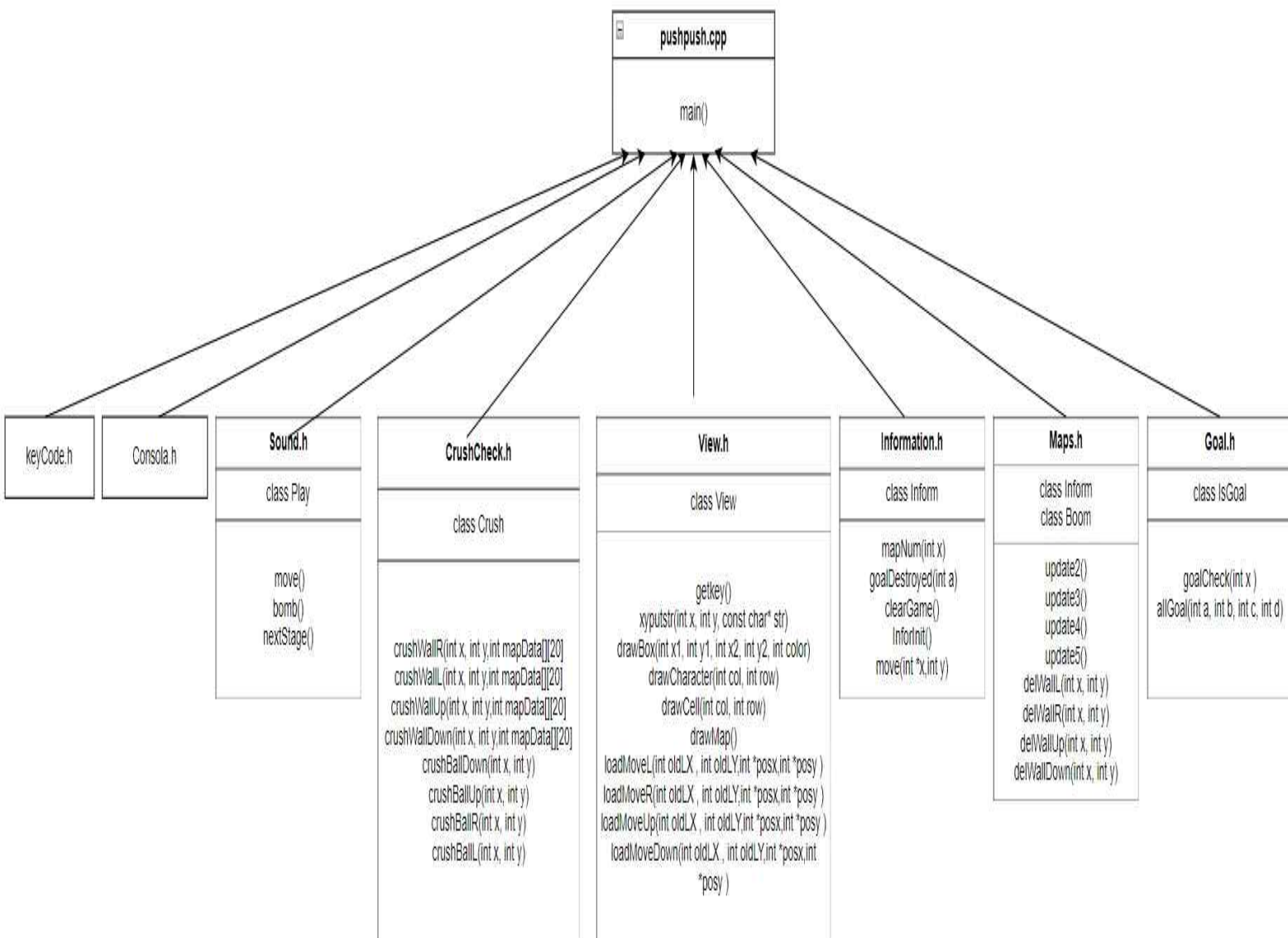
## <그림 g>



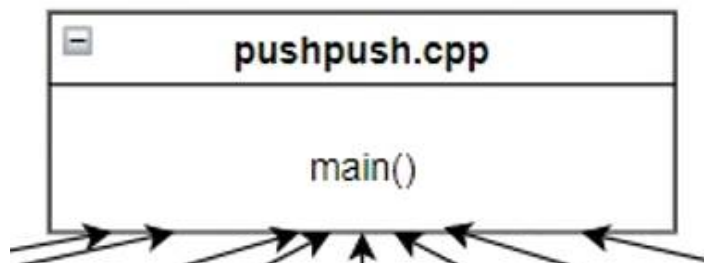
# ▶▶개발 일정◀◀

1. 캐릭터 및 맵 이동가능 구간 구현
2. 공(load)추가 및 벽,공끼리 충돌시 처리 구현
3. 목표물 구현 및 공이 들어갔을 때 효과 구현
4. >>,<< 단방향 이동 구간 구현
5. 폭탄 추가 및 벽,목표물,단방향 이동구간 충돌효과 구현
6. 목표물 4개 성공시 다음 맵으로 이동하는 기능 구현
7. 게임판의 오른쪽에 게임 내용을 알리는 구역 구현
8. 이동 가능 횟수,맵 클리어,목표물 파괴시 게임 종료 구현
9. 캐릭터 이동 및 폭탄 사용시 사운드 효과 구현
10. 게임 내 맵을 아이템을 이용해 자세히 구현
11. 맵 클리어 후 다음 맵 이동시 사운드 효과 구현
12. 테스트 후 오류 점검 및 문제점 보완

## ▶▶ 소스코드의 자료구조 및 로직 ◀◀



## ▶▶시스템 구성 파일 및 설명◀◀



1)pushpush.cpp

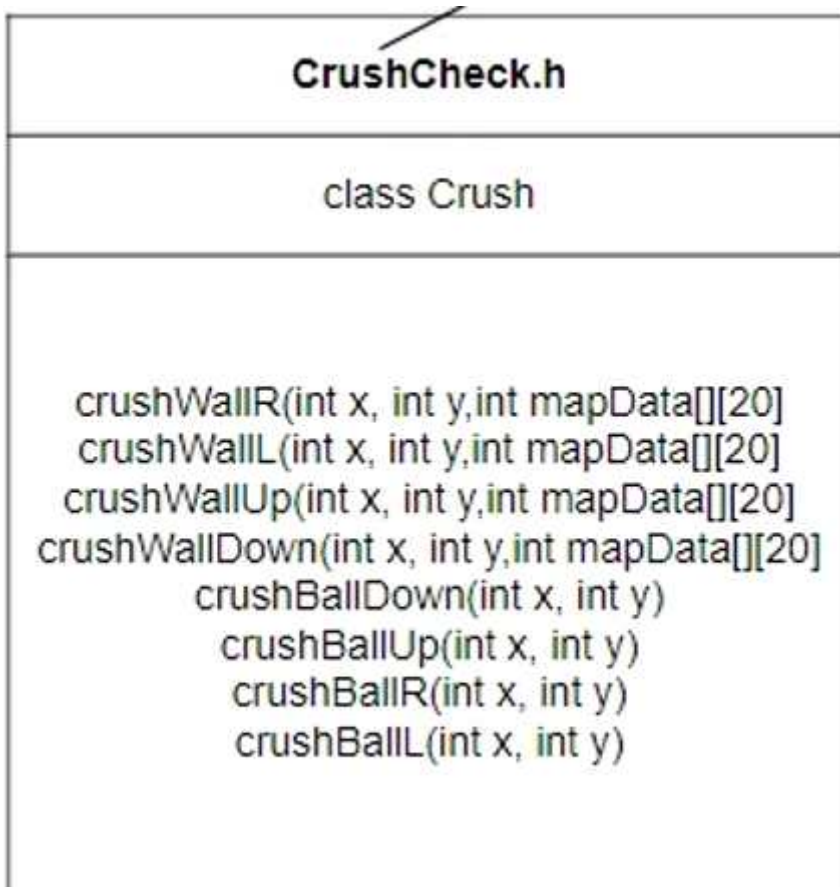
-main함수()

View.h
class View
<div>getkey() xyputstr(int x, int y, const char* str) drawBox(int x1, int y1, int x2, int y2, int color) drawCharacter(int col, int row) drawCell(int col, int row) drawMap() loadMoveL(int oldLX , int oldLY,int *posx,int *posy ) loadMoveR(int oldLX , int oldLY,int *posx,int *posy ) loadMoveUp(int oldLX , int oldLY,int *posx,int *posy ) loadMoveDown(int oldLX , int oldLY,int *posx,int *posy )</div>

2)View.h

-getkey(): 키 입력을 받는 함수

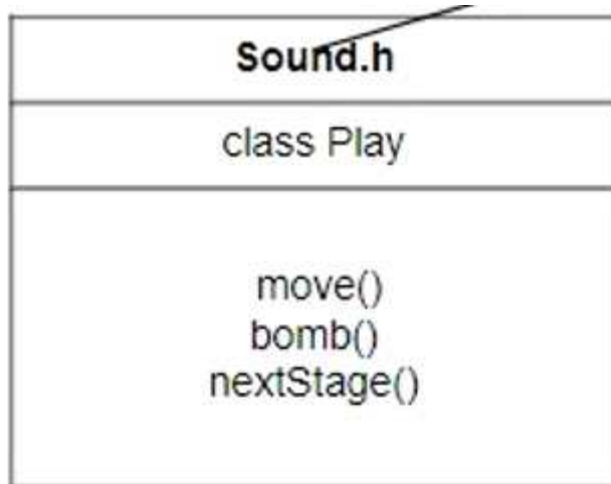
-xyputstr(int x,int y,const char\*str):맵을 그릴 때 사용하는 함수  
 -drawBox(int x1,int y1, int x2, int y2, int color):게임 박스를 그리는 함수  
 -drawCharacter(int col,int row):캐릭터를 맵에 그리는 함수  
 -drawCell(int col,int row):해당 좌표의 셀을 그리는 함수  
 -drawMap():맵을 그리는 함수  
 -loadMoveL(int oldLX, int oldLY, int \*posx, int \*posy):공을 왼쪽으로 이동시킬 때 사용하는 함수  
 -loadMoveR(int oldLX, int oldLY, int \*posx, int \*posy):공을 오른쪽으로 이동시킬 때 사용하는 함수  
 -loadMoveDown(int oldLX, int oldLY, int \*posx, int \*posy):공을 아래쪽으로 이동시킬 때 사용하는 함수  
 -loadMoveUp(int oldLX, int oldLY, int \*posx, int \*posy):공을 위쪽으로 이동시킬 때 사용하는 함수



### 3)CrushCheck.h

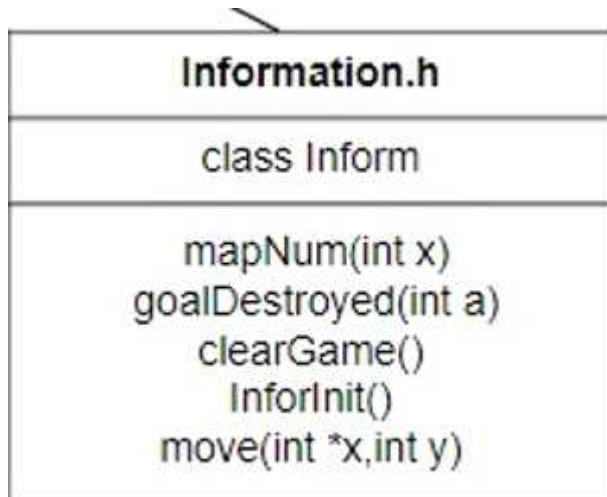
-crushWallR(int x, int y, int mapData[][20]):캐릭터가 벽과 충돌하는지 체크하는 함수  
 -crushWallL(int x, int y, int mapData[][20]):캐릭터가 벽과 충돌하는지 체크하는 함수  
 -crushWallUp(int x, int y, int mapData[][20]):캐릭터가 벽과 충돌하는지 체크하는 함수  
 -crushWallDown(int x, int y, int mapData[][20]):캐릭터가 벽과 충돌하는지 체크하는 함수  
 -crushBallDown(int x, int y):공이 공끼리 충돌 혹은 단방향 이동불력과 충돌하는지 체크하는 함수

- crushBallUp(int x, int y):공이 공끼리 충돌 혹은 단방향 이동블럭과 충돌하는지 체크하는 함수
- crushBallL(int x, int y):공이 공끼리 충돌 혹은 단방향 이동블럭과 충돌하는지 체크하는 함수
- crushBallR(int x, int y):공이 공끼리 충돌 혹은 단방향 이동블럭과 충돌하는지 체크하는 함수



#### 4)Sound.h

- move():캐릭터 이동시 사운드 효과 처리 함수
- bomb():폭탄이 사용될 때 사운드 효과 처리 함수
- nextStage():다음 맵으로 이동시 사운드 효과 처리 함수



#### 5)Information.h

- clearGame():게임 클리어 화면 처리 함수
- goalDestroyed(int a):목표물 파괴시 게임 종료 화면 처리 함수
- mapNum(int x):맵 번호를 알려주는 구간을 처리하는 함수
- move(int \*x,int y):이동 횟수를 알려주는 구간을 처리하는 함수
- InforInit():콘솔창 오른쪽에 알림창을 그리는 함수



Maps.h
class Inform class Boom
update2() update3() update4() update5() delWallL(int x, int y) delWallR(int x, int y) delWallUp(int x, int y) delWallDown(int x, int y)

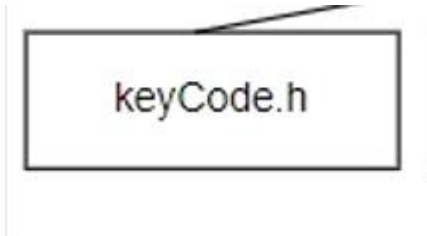
## 6)Maps.h

- update2():다음 맵으로 업데이트 처리하는 함수
- update3():다음 맵으로 업데이트 처리하는 함수
- update4():다음 맵으로 업데이트 처리하는 함수
- update5():다음 맵으로 업데이트 처리하는 함수
- delWallL(int x , int y): 폭탄 사용시 적용된 좌표를 수정하는 함수
- delWallR(int x , int y): 폭탄 사용시 적용된 좌표를 수정하는 함수
- delWallUp(int x , int y): 폭탄 사용시 적용된 좌표를 수정하는 함수
- delWallDown(int x , int y): 폭탄 사용시 적용된 좌표를 수정하는 함수

Goal.h
class IsGoal
goalCheck(int x ) allGoal(int a, int b, int c, int d)

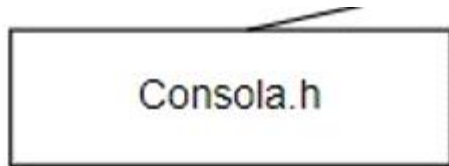
## 7)Goal.h

- goalCheck(int x): 전달 받은 변수가 목표물을 저장한 변수(2)인지 확인하는 함수
- allGoal(int a, int b , int c ,in d): 모든 목표물에 load가 들어왔는지 확인하는 함수



## 8)keyCode.h

- #define M\_UPKEY 0xe048: 방향키 위의 키 코드 값
- #define M\_DOWNKEY 0xe050:방향키 아래의 키 코드 값
- #define M\_RIGHTKEY 0xe04d: 방향키 오른쪽의 키 코드 값
- #define M\_LEFTKEY 0xe04b: 방향키 왼쪽의 키 코드 값



## 9)Consola.h