

## main.c

### 역할

- 프로그램 메인 루프 처리, 초기화 및 해제

## resource.c

### 역할

- 외부 리소스 및 전역변수 선언

### 함수

- void load\_resource(void);
  - 설명 : 이미지, 폰트, 오디오 파일 등의 필요한 외부파일 로드
- void destroy\_resource(void);
  - 설명 : 종료시 리소스 해제
- ALLEGRO\_BITMAP\* convert\_to\_grayscale(ALLEGRO\_BITMAP\* original);
  - 입력 : ALLEGRO\_BITMAP\* 형식의 원본
  - 설명 : 비트맵을 회색조로 변경
  - 출력 : 원본에서 회색조로 변경된 이미지의 포인터

## keyboard.c

### 역할

- 키보드 이벤트 처리

### 구조체

- Direction 열거형
  - 설명 : UP, DOWN, LEFT, RIGHT의 4가지 방향을 표현함.

### 함수

- void keyboard\_init(void);
  - 설명 : 최초 실행시 키보드 초기화
- void keyboard\_update(ALLEGRO\_EVENT\*);
  - 입력 : ALLEGRO\_EVENT\* 형식의 이벤트 함수
  - 설명 : 키보드를 눌렀을 때, 키보드를 땔때 각각의 이벤트에 따라 키보드의 눌린 상태 업데이트, timer 이벤트 발생시 모든 키보드에 대해 플래그 초기화

## gamescreen.c

### 역할

- 게임 실행시 메인 루프 처리

### 함수

- void gamescreen(void);
  - 설명 : 게임 실행 중에 플레이어 입력에 따라 이미지를 화면에 출력하는 기능을 담당

## level.c

### 역할

- 난이도 관련 처리

### 함수

- void level\_up(void);
  - 설명 : 디버프 선택 화면 출력
- int debuff\_button(int, int, int, int);
  - 입력 : 현재 마우스의 x, y 정수 좌표값, 버튼의 x, y 정수 좌표값
  - 설명 : 마우스가 버튼 위치를 눌렀는지 판정
  - 출력 : 마우스가 버튼을 눌렀는지에 대한 true / false
- void debuff\_action(int);
  - 입력 : 0~5 사이의 정수값
  - 설명 : 입력된 flg 값에 따른 난이도 증가 처리
- 

## hud.c

### 역할

- 게임 실행 중 플레이어의 정보를 실시간으로 출력하는 기능

### 함수

- void hud\_init(void);
  - 설명 : 게임 재시작 시, 점수 초기화
- void hud\_draw(void);
  - 설명 : 플레이어의 점수 및 보유 재화와 재화에 따른 기능 활성화/비활성  
가시화를 구현

## init.c

### 역할

- 게임에 필요한 자원 초기화

### 함수

- void reset\_game\_state(void);
  - 설명 : 게임 재시작 시, 이전 게임에 사용된 자원들을 모두 초기화
- void reset\_keyboard\_event(void);
  - 설명 : 게임 재시작 시, 키보드 버퍼에 있는 입력 초기화
- void init(void);
  - 설명 : 게임 최초 실행 시, 게임에 필요한 자원 초기화

## object.c

### 역할

- 플레이어, 소환수, 적 생성 및 **collision detection** 처리

### 구조체

- **Player** 구조체
  - 설명 : 플레이어의 스테이터스 및 좌표 저장
- **Summon** 구조체
  - 설명 : 소환수와 적 스테이터스 및 좌표 저장
- **Bullet** 구조체
  - 설명 : 플레이어 및 적의 총알 좌표 및 방향 저장
- **LaserTarget** 구조체 :
  - 설명 : 레이저를 고정할 때 필요한 플레이어의 좌표 저장

### 함수

- + 플레이어 관련 함수
  - **void move\_player(void);**
    - 설명 : 플레이어 이동 처리
  - **void player\_enhance\_sp(void);**
    - 설명 : 플레이어 공격속도 강화 처리
  - **void player\_enhance\_dm(void);**
    - 설명 : 플레이어 데미지 강화 처리
  - **void player\_enhance\_bu(void);**
    - 설명 : 플레이어 볼릿 개수 강화 처리
  - **void fire\_bullet(void);**
    - 설명 : 플레이어 볼릿 생성 함수
  - **void move\_bullets(void);**
    - 설명 : 플레이어 볼릿의 이동 처리
  - **void check\_bullet\_collision(void);**
    - 설명 : 플레이어 볼릿 충돌 처리
- + 필살기 관련 함수
  - **void Special\_moves(void);**
    - 설명 : 플레이어의 필살기 함수, 사용시 적의 체력을 일정량 감소시키고 플레이어에게 무적 시간 부여.
  - **void apply\_screen\_shake(int, int);**
    - 입력 : **render\_screen(void)**함수에서 제공한 랜덤 좌표값
    - 설명 : 화면의 흔들림 효과 구현
  - **void render\_screen();**
    - 설명 : 지진 랜덤 좌표값 설정

- + 몬스터 공통 함수
  - void spawn\_enemy(int);
    - 입력 : 1과 2
    - 설명 : 적 생성 함수, 1일 때 기사, 2일 때 마법사 몬스터를 소환
  - void clear\_summons(int);
    - 입력 : 1과 11로 나누어 한개의 함수에서 소환수와 적 초기화
    - 설명 : 소환수 및 적 초기화 함수
  - void check\_collision();
    - 설명 : 적과 소환수 충돌 함수
  - void move\_summons();
    - 설명 : 소환수 이동 함수
  - void update\_animation2();
    - 설명 : 적 공격모션 함수
  
- + 전사 몬스터 관련 함수
  - void check\_player\_collision();
    - 설명 : 전사 몬스터와 플레이어 충돌 함수
  
- + 마법사 몬스터 관련 함수
  - void move\_boss\_bullets(void);
    - 설명 : 마법사 몬스터 볼릿 이동 함수
  - void move\_boss\_lasers();
    - 설명 : 마법사 몬스터 레이저 이동 함수
  - void attack\_boss(void);
    - 설명 : 마법사 몬스터 볼릿 생성 함수
  - void attack\_laser\_boss();
    - 설명 : 마법사 몬스터 레이저 생성 함수
  - void check\_boss\_bullet\_collision(void);
    - 설명 : 마법사 몬스터 볼릿 충돌 함수
  - void check\_boss\_laser\_collision(void);
    - 설명 : 마법사 레이저 충돌 함수
  - void draw\_rotated\_laser(int);
    - 입력 : 레이저 생성 시 보스의 인덱스 값
    - 설명 : 레이저 사진 회전 함수
  
- + 소환수 관련 함수
  - void spawn\_summon(int);
    - 입력 : 입력값을 1과 11로 나누어 소환수와 적을 한개의 함수에서 처리
    - 설명 : 소환수 및 적 생성 함수
  
- + 공격 모션 함수
  - void update\_animation();
    - 설명 : 소환수 공격모션 함수
  - void update\_animation2();
    - 설명 : 적 공격모션 함수

## menu.c

### 역할

- 게임의 전반적인 UI/메뉴 처리 기능

### 구조체

- **Player1** 구조체
  - 설명 : 플레이어의 점수 기록을 위해 이름, 점수 저장

### 함수

- + 메인 메뉴 관련
  - **void draw\_menu(void);**
    - 설명 : 메뉴 배경, 화면 출력
  - **bool is\_button\_clicked(int mouse\_x, int mouse\_y, int button\_y\_offset);**
    - 설명 : 마우스 클릭이 특정 메뉴 버튼 범위 안에 있는지를 판단
  - **void exit\_game();**
    - 설명 : 게임 종료
- + 랭킹 메뉴 관련
  - **void show\_rankings(ALLEGRO\_FONT\* font);**
    - 설명 : 상위 5명의 랭킹을 출력하고, Back / Clear 버튼을 통해 메뉴로 돌아가거나 기록 초기화
  - **int read\_rankings(Player1 players[], int max\_players);**
    - 설명 : 텍스트 파일(rank.txt)에서 최대 max\_players명의 랭킹 정보를 읽어와 players 배열에 저장
  - **int compare\_players(const void\* a, const void\* b);**
    - 설명 : 점수를 기준으로 두 플레이어를 비교하여 내림차순 정렬에 사용
  - **void rank(void);**
    - 설명 : 랭킹 화면 진입 함수
  - **void name(ALLEGRO\_FONT\*);**
    - 설명 : 게임 종료 후 플레이어의 이름을 입력받는 화면
  - **void save\_score(const char\*, long);**
    - 설명 : 게임 종료 후 플레이어의 이름과 점수를 rank.txt에 저장
  - **int compare\_players(const void\* a, const void\* b);**
    - 설명 : 점수를 기준으로 두 플레이어를 비교하여 내림차순 정렬에 사용
  - **void rank(void);**
    - 설명 : 랭킹 화면 진입 함수
  - **void save\_score(const char\*, long);**
    - 설명 : 게임 종료 후 플레이어의 이름과 점수를 rank.txt에 저장
  - **void clear\_rankings();**
    - 설명 : rank.txt 내의 플레이어 랭킹(데이터) 초기화