***상위 설계서***

***(Top Level Design)***

나나 존스와 보물성

설계완성

Ver. 2.00 (버전 관리를 위한 작업

프로젝트 완료시 1.00이 되어야 함)

반: IMCE 3반

이름: 김민경

**[ 목차 ]**

[1. 프로그램 개발 전략 3](#_Toc364449214)

[2. State Transition Diagram 4](#_Toc364449215)

[3. Main Layer 함수 정의 5](#_Toc364449216)

[3.1. void Game\_Manager(void) 5](#_Toc364449217)

[3.2. void Display\_Manager(void) 5](#_Toc364449218)

[3.3. void Operation\_Manager(void) 6](#_Toc364449219)

[4. Sub Layer 함수 정의 7](#_Toc364449220)

[4.1. void (void) 7](#_Toc364449221)

[4.2. void 7](#_Toc364449222)

4.3 void

4.4 void

4.5 void

4.6 void

4.7 void

4.8 void

4.9 void

4.10 void

4.11 void

4.12 void

4.13 void

4.14 void

4.15 void

4.16 void

4.17

4.18

4.19

4.20

4.21

4.22

4.23

4.24

4.25

2.26

4.27

4.28

4.29

4.30

[5. 전역변수(flag) 및 상수 정의 8](#_Toc364449223)

[5.1. 상수 정의 8](#_Toc364449224)

[5.2. 전역변수 정의 8](#_Toc364449225)

[5.3. flag 정의 8](#_Toc364449226)

# 프로그램 개발 전략

개발 전략은 기존에 있었던 UFO 게임을 어떻게 이 새로운 테마와 액션에 더욱더 efficient 하게 설계 하냐가 관건입니다.

함수와 변수, 구조체와 구조체 배열을 전에 했던 것처럼 단순하게 메인에 넣는 방식이 아닌, 함수로 빼돌리고 구현 하는 쪽을 선택한 것입니다. 물론, 매크로 함수와 다른 환경을 생각하기도 합니다.

UFO 게임의 기본 컨셉을 빌려다 썼으나 **나나존스**는 새로운 게임입니다. 아이템이라는 새로운 컨셉과 “비밀 통로”라는 새로운 곳을 거쳐야 새로운 레벨에 도전 할 수 있는, 다양화 된 레벨에 대한 개념과 미니 퀘스트를 통한 최고 점수 도전에 대한 심리를 키운 다는 것 입니다.

**강조하고 싶은 기술**

***1.1 Achievement Algorithm and Mechanics***

* 기본 게임의 슈팅과 자율 움직임은 그대로 입니다. 그러나 이제 4방으로 슈팅가능 합니다.
* 매우 단순해 보이지만 아이템 콜렉팅 이라는 아이디어가 . 기존 슈팅해서 포인트를 얻는 방법이 아닌, 아예 아이템을 얻어야만 이뤄질 수 있는 게임 엘레멘트입니다. 그리고 게임의 핵심이기도 합니다. 아이템을 다 얻지 못하면 다음 레벨로 넘어갈 수가 없습니다.
* 나나존스의 레벨 시스템은 조건을 둘 중 하나만 충족하여도 레벨 업을 할 수 있는 시스템입니다.

***1.2 Graphical Considerations***

* 새로 쓰는 알고리즘은 그래픽도 고려해야 합니다. 구조체로 아이템을 만드는 것과 아이템과 다른 객체들의 interaction 도 고려했음을 말씀 드립니다.
* 나나존스의 핵심이라고는 말 할 수 없지만 독특한 점은 레벨마다 누적 점수를 보여준다는 점일 것입니다.
* 각 레벨마다 배경이 바뀝니다

***1.3 Algorithmic and Programming Improvements***

* 나나존스의 꽃이라고 볼 수 있는 기능은 바로 나나 존스의 구조체 배열과 레벨마다 차별화된 난이도 레벨이라고 생각합니다. 5가지 레벨에서는 전 레벨 보다 더욱더 어려운 게임이 나옵니다.
* 기존의 게임에 나왔던 죽일 수 있는 몬스터 (UFO)는 물론 등장, 그리고 더욱더 차별화 시킨 죽일 수 없는 몬스터 (그러나 플레이어를 죽일 수 있는 몬스터)도 포함, 기존의 쓸쓸했던 게임 스크린이 채워졌습니다.

# State Transition Diagram

출력

동작

If Game\_Start\_Flag=1 game proceeds (all other state flags must = 1)

OM gives flags out for positive verification by the DM.

If DM gets the information from OM, DM displays on the screen what OM

requested.

# Main Layer 함수 정의

## **void Display\_Manager(void)**

|  |  |
| --- | --- |
| 기능 | 화면 업데이트  모든 객체들의 위치와 이미지 표시와 갱신  점수 및 라이프 표시 업데이트  객체 간의 충돌 표시 및 효과음 표현 |
| Parameter | void |
| Return Value | void |
| 호출 함수  (Sub Layer 함수) | Void Disp\_Nana (void)  Void Disp\_Fire (void)  Void Disp\_Mon (void)  Void Disp\_Heart (void)  Void Disp\_Coin (void)  Void Disp\_Stage (void)  Void Disp\_Fire\_Nana (void)  Void Disp\_Mon\_Nana (void)  Void Disp\_Heart\_Mon (void)  void Disp\_Lose (void)  void Disp\_Win (void)  void Disp\_Ini (void)  void Disp\_Info(void) |
| 사용하는  외부변수 및 flag | int Display\_flag  int Stage\_Num  int Stage\_flag  int score  Int Lose\_Flag  Int Win\_Flag  //background  Int e  Struct Ob1 Nana  Struct Ob2 Fire[MAX\_FIRE\_NUM]  Struct Ob2 Mon[MAX\_MON\_NUM]  Struct Ob3 Heart[MAX\_HEART\_NUM]  Struct Ob4 Coin[MAX\_COIN\_NUM] |
| 비고 |  |

## **void Op\_Manager(void)**

|  |  |
| --- | --- |
| 기능 | 물체 움직임 위치 갱신 및 이미지 플래그 갱신  충돌 판정 (나나/몬스터, 하트/몬스터)  비밀의 통로와 나나의 충돌 감지  점수 및 라이프 관리  여기 정보로 뉴스테이지/윈/루즈 플래그 활성화 |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) | Void Op\_Nana(void)  Void Op\_Fire (void)  Void Op\_Mon (void)  Void Op\_Heart(void)  Void Op\_Coin (void)  Void Op\_Fire\_Rand(void)  Void Op\_Mon\_Rand(void)  Void Op\_Mon\_Heart(void)  Void Op\_Fire\_Nana(void)  Void Op\_Mon\_Nana(void)  Void Op\_Coin\_Nana(void) |
| 사용하는  외부변수 및 flag | int Display\_flag  int Game\_start\_flag  int Stage\_flag  int Stage\_Num  int Stage\_flag  int score  int heartcnt  int sangtae  int Main\_Disp\_Flag  int Game\_Over\_Disp\_Flag  Int Lose\_Flag  Int Win\_Flag  int heartcnt  int monlife  int coinlife  //background  Int e  Struct Ob1 Nana  Struct Ob2 Fire[MAX\_FIRE\_NUM]  Struct Ob2 Mon[MAX\_MON\_NUM]  Struct Ob3 Heart[MAX\_HEART\_NUM]  Struct Ob4 Coin[MAX\_COIN\_NUM] |
| 비고 |  |

# Sub Layer 함수 정의

본 단계의 함수들은 가급적 전역변수나 flag을 사용하지 않는 것이 좋음

Parameter로 전달하거나 call by address 등으로 구현하는 것이 함수의 독립성과 유지보수성 제고에 도움이 됨

## **void Stage\_Ini (void)**

|  |  |
| --- | --- |
| 기능 | 게임/레벨 초기화 |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) | Void Disp\_Ini (void) |
| 사용하는  외부변수 및 flag | int Game\_start\_flag  int Stage\_flag  int Stage\_Num  int Lose\_Flag  int Win\_Flag  int heartcnt  int monlife  int coinlife  Struct Ob1 Nana  Struct Ob2 Fire[MAX\_FIRE\_NUM]  Struct Ob2 Mon[MAX\_MON\_NUM]  Struct Ob3 Heart[MAX\_HEART\_NUM]  Struct Ob4 Coin[MAX\_COIN\_NUM] |
| 비고 |  |

## **void Stage\_Ini(void)**

|  |  |
| --- | --- |
| 기능 | 다음 레벨로 올라갈 시 모든 변수 초기화 |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) |  |
| 사용하는  외부변수 및 flag | int Game\_start\_flag  int Stage\_flag  int Stage\_Num  int Lose\_Flag  int Win\_Flag  int stagefire  int stagemon  int stagecoin  int hearts  //background  Int e  Struct Ob1 Nana  Struct Ob2 Fire[MAX\_FIRE\_NUM]  Struct Ob2 Mon[MAX\_MON\_NUM]  Struct Ob3 Heart[MAX\_HEART\_NUM]  Struct Ob4 Coin[MAX\_COIN\_NUM] |
| 비고 | This just initializes everything for a new stage (including the number of hearts, lives, etc) |

## **void Op\_Nana (void)**

|  |  |
| --- | --- |
| 기능 | 나나/플레이어를 움직인다 (key 값 받을시) |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) | Void Key\_Get\_Pressed(); |
| 사용하는  외부변수 및 flag | Struct Ob1 Nana |
| 비고 | 2 left  4 right  1 up  3 down |

## **void Op\_Fire (void)**

|  |  |
| --- | --- |
| 기능 | 파이어 볼의 위치를 갱신 (타이머) |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) | Void Op\_Fire\_Rand () |
| 호출 함수  (디바이스 드라이버) |  |
| 사용하는  외부변수 및 flag | Struct Ob2 Fire[MAX\_FIRE\_NUM]  Int b |
| 비고 | X, Y 좌표 플래그 조종 |

## **void Op\_Mon(void)**

|  |  |
| --- | --- |
| 기능 | 몬스터들의 동작을 갱신 (타이머) |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) | Void Op\_Mon\_Rand () |
| 호출 함수  (디바이스 드라이버) |  |
| 사용하는  외부변수 및 flag | Struct Ob2 Mon[MAX\_MON\_NUM]  Int a |
| 비고 | X, Y 좌표 플래그 조종 |

## **4.6 void Op\_Heart (void)**

|  |  |
| --- | --- |
| 기능 | 키를 입력 받을 시 하트를 activate 시킨다  (나나와 동일 위치) |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) | Void Key\_Get\_Pressed(); |
| 사용하는  외부변수 및 flag | Struct Ob3 Heart[MAX\_HEART\_NUM]  Int c |
| 비고 | Button 5 shoots |

## **void Op\_Coin (void)**

|  |  |
| --- | --- |
| 기능 | 코인들의 위치를 알려줌 |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) |  |
| 사용하는  외부변수 및 flag | Struct Ob4 Coin[MAX\_COIN\_NUM]  Int d |
| 비고 | X, Y 좌표 플래그 조종 |

## **void Op\_Fire\_Rand (void)**

|  |  |
| --- | --- |
| 기능 | 파이어 볼의 동작 랜덤화 |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) |  |
| 사용하는  외부변수 및 flag | Struct Ob2 Fire[MAX\_FIRE\_NUM] |
| 비고 | 스피드 조종 (랜드화) |

## **void Op\_Mon\_Rand (void)**

|  |  |
| --- | --- |
| 기능 | 몬스터의 동작 랜덤화 |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) | rand |
| 사용하는  외부변수 및 flag | Struct Ob2 Mon[MAX\_MON\_NUM] |
| 비고 | 스피드 조종 (랜드화) |

## **void Op\_Mon\_Heart (void)**

|  |  |
| --- | --- |
| 기능 | 몬스터를 하트로 격침 했을 시 일어나는 일입니다. |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) |  |
| 사용하는  외부변수 및 flag | Struct Ob2 Mon[MAX\_MON\_NUM]  Struct Ob3 Heart[MAX\_HEART\_NUM] |
| 비고 | 몬스터 격침 시 포인트 증가 |

## **void Op\_Fire\_Nana(void)**

|  |  |
| --- | --- |
| 기능 | 파이어 볼이 나나와 충돌시에 일어나는 일입니다. |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) |  |
| 사용하는  외부변수 및 flag | Struct Ob1 Nana  Struct Ob2 Fire[MAX\_FIRE\_NUM] |
| 비고 | 파이어볼이 나나와 충돌시 포인트와 생명 감소 |

## **void Op\_Mon\_Nana(void)**

|  |  |
| --- | --- |
| 기능 | 몬스터와 나나가 충돌시에 일어나는 일입니다. |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) |  |
| 사용하는  외부변수 및 flag | Struct Ob1 Nana  Struct Ob2 Mon[MAX\_MON\_NUM] |
| 비고 | 몬스터가 나나와 충돌시 포인트와 생명 감소 |

## **void Op\_Coin\_Nana(void)**

|  |  |
| --- | --- |
| 기능 | 나나와 코인의 위치가 만났을 시 일어나는 일입니다. |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) |  |
| 사용하는  외부변수 및 flag | Struct Ob1 Nana  Struct Ob4 Coin[MAX\_COIN\_NUM] |
| 비고 | Plus points and if coin#==stagecoin numbers stage win and go over to the next stage (monster kill, too?) |

## **void Disp\_Fire (void)**

|  |  |
| --- | --- |
| 기능 | 파이어볼 이미지 출력 |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) | Void LCD\_Draw\_BMP (void)  Void LCD\_Draw\_Back (int x, int y, int x2, int y2, const unsigned short int \*fp)  Void Lcd\_Set\_Shape\_Mode(int, COLOR)  Void Lcd\_Printf() |
| 사용하는  외부변수 및 flag | Struct Ob2 Fire[MAX\_FIRE\_NUM] |
| 비고 | 이미지 배열 사용 |

## **void Disp\_Mon (void)**

|  |  |
| --- | --- |
| 기능 | 몬스터 이미지 출력 |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) | Void LCD\_Draw\_BMP (void)  Void LCD\_Draw\_Back (int x, int y, int x2, int y2, const unsigned short int \*fp)  Void Lcd\_Set\_Shape\_Mode(int, COLOR)  Void Lcd\_Printf() |
| 사용하는  외부변수 및 flag | Struct Ob2 Mon[MAX\_MON\_NUM] |
| 비고 | 이미지 배열 사용 |

## **void Disp\_Heart (void)**

|  |  |
| --- | --- |
| 기능 | 하트슈터 이미지 출력 |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) | Void LCD\_Draw\_BMP (void)  Void LCD\_Draw\_Back (int x, int y, int x2, int y2, const unsigned short int \*fp)  Void Lcd\_Set\_Shape\_Mode(int, COLOR)  Void Lcd\_Printf()  Void Key\_Get\_Pressed(); |
| 사용하는  외부변수 및 flag | Struct Ob3 Heart[MAX\_HEART\_NUM] |
| 비고 | 이미지 배열 사용 |

## **void Disp\_Coin (void)**

|  |  |
| --- | --- |
| 기능 | 매직코인 이미지 출력 |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) | Void LCD\_Draw\_BMP (void)  Void LCD\_Draw\_Back (int x, int y, int x2, int y2, const unsigned short int \*fp)  Void Lcd\_Set\_Shape\_Mode(int, COLOR)  Void Lcd\_Printf() |
| 사용하는  외부변수 및 flag | Struct Ob4 Coin[MAX\_COIN\_NUM] |
| 비고 | 이미지 하나  만약 이미지 플래그가 0가 되면 사라지고 포지션 리셋 |

## **void Disp\_Nana(void)**

|  |  |
| --- | --- |
| 기능 | 나나의 이미지 출력 |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) | Void LCD\_Draw\_BMP (void)  Void LCD\_Draw\_Back (int x, int y, int x2, int y2, const unsigned short int \*fp)  Void Lcd\_Set\_Shape\_Mode(int, COLOR)  Void Lcd\_Printf()  Void Key\_Get\_Pressed(); |
| 사용하는  외부변수 및 flag | Struct Ob1 Nana |
| 비고 | 이미지 배열 사용 |

## **Void Disp\_Stage (void)**

|  |  |
| --- | --- |
| 기능 | Initial Stage 출력 |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) | Void LCD\_Draw\_BMP (void)  Void LCD\_Draw\_Back (int x, int y, int x2, int y2, const unsigned short int \*fp)  Void Lcd\_Set\_Shape\_Mode(int, COLOR)  Void Lcd\_Printf() |
| 사용하는  외부변수 및 flag | int Game\_start\_flag  int Stage\_flag  int Stage\_Num  int Lose\_Flag  int Win\_Flag  int stagefire  int stagemon  int stagecoin  int hearts  //background  Int e  Struct Ob1 Nana  Struct Ob2 Fire[MAX\_FIRE\_NUM]  Struct Ob2 Mon[MAX\_MON\_NUM]  Struct Ob3 Heart[MAX\_HEART\_NUM]  Struct Ob4 Coin[MAX\_COIN\_NUM] |
| 비고 | 스테이지/게임 시작 시 게임 자체의 첫 화면 출력 |

## **Void Disp\_Fire\_Nana (void)**

|  |  |
| --- | --- |
| 기능 | 나나와 불 충돌시 출력 |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) | Void LCD\_Draw\_BMP (void)  Void LCD\_Draw\_Back (int x, int y, int x2, int y2, const unsigned short int \*fp)  Void Lcd\_Set\_Shape\_Mode(int, COLOR)  Void Lcd\_Printf() |
| 사용하는  외부변수 및 flag | Struct Ob1 Nana  Struct Ob2 Fire[MAX\_FIRE\_NUM] |
| 비고 | 이미지 출력 및 포지션 리셋 |

## **Void Disp\_Mon\_Nana (void)**

|  |  |
| --- | --- |
| 기능 | 나나와 몬스터 충돌시 출력 |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) | Void LCD\_Draw\_BMP (void)  Void LCD\_Draw\_Back (int x, int y, int x2, int y2, const unsigned short int \*fp)  Void Lcd\_Set\_Shape\_Mode(int, COLOR)  Void Lcd\_Printf() |
| 사용하는  외부변수 및 flag | Struct Ob1 Nana  Struct Ob2 Mon[MAX\_MON\_NUM] |
| 비고 | 이미지 출력및 포지션 리셋 |

## **void Disp\_Heart\_Mon (void)**

|  |  |
| --- | --- |
| 기능 | 슈터 하트와 몬스터 충돌시 출력 |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) | Void LCD\_Draw\_BMP (void)  Void LCD\_Draw\_Back (int x, int y, int x2, int y2, const unsigned short int \*fp)  Void Lcd\_Set\_Shape\_Mode(int, COLOR)  Void Lcd\_Printf() |
| 사용하는  외부변수 및 flag | Struct Ob2 Mon[MAX\_MON\_NUM]  Struct Ob3 Heart[MAX\_HEART\_NUM]] |
| 비고 | 이미지 출력 및 포지션 리셋 (몬스터가 사라진다) |

## **void Disp\_Lose (void)**

|  |  |
| --- | --- |
| 기능 | 루즈 화면 출력 |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) | Void LCD\_Draw\_BMP (void)  Void LCD\_Draw\_Back (int x, int y, int x2, int y2, const unsigned short int \*fp)  Void Lcd\_Set\_Shape\_Mode(int, COLOR)  Void Lcd\_Printf() |
| 사용하는  외부변수 및 flag | Int Lose\_flag |
| 비고 |  |

## **Void Disp\_Win (void)**

|  |  |
| --- | --- |
| 기능 | 윈 화면 출력 |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) | Void LCD\_Draw\_BMP (void)  Void LCD\_Draw\_Back (int x, int y, int x2, int y2, const unsigned short int \*fp)  Void Lcd\_Set\_Shape\_Mode(int, COLOR)  Void Lcd\_Printf() |
| 사용하는  외부변수 및 flag | Int Win\_flag |
| 비고 |  |

## **Void Disp\_Ini (void)**

|  |  |
| --- | --- |
| 기능 | 첫 화면 출력 (단 한번뿐) |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) | Void LCD\_Draw\_BMP (void) |
| 사용하는  외부변수 및 flag |  |
| 비고 |  |

## **void Disp\_Info(void)**

|  |  |
| --- | --- |
| 기능 | 점수, 라이프, 스테이지, 미사일 디스플레이 (just nums) |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) | Void LCD\_Draw\_BMP (void)  Void LCD\_Draw\_Back (int x, int y, int x2, int y2, const unsigned short int \*fp)  Void Lcd\_Set\_Shape\_Mode(int, COLOR)  Void Lcd\_Printf() |
| 사용하는  외부변수 및 flag |  |
| 비고 |  |

## **void Disp\_Between\_Stage(void) ?????**

|  |  |
| --- | --- |
| 기능 | 스테이지 사이사이 디스플레이 (?) |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) | Void LCD\_Draw\_BMP (void)  Void LCD\_Draw\_Back (int x, int y, int x2, int y2, const unsigned short int \*fp)  Void Lcd\_Set\_Shape\_Mode(int, COLOR)  Void Lcd\_Printf() |
| 사용하는  외부변수 및 flag |  |
| 비고 |  |

## **void Op\_Nana\_Lim(void)**

|  |  |
| --- | --- |
| 기능 | 나나 행동반경 |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) | Void Disp\_Ini (void) |
| 사용하는  외부변수 및 flag | Struct Ob1 Nana |
| 비고 |  |

## **void Op\_Mon\_Lim(void)**

|  |  |
| --- | --- |
| 기능 | 몬스터 행동 반경 |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) | Void Disp\_Ini (void) |
| 사용하는  외부변수 및 flag | int stagemon  Struct Ob2 Mon[MAX\_MON\_NUM] |
| 비고 |  |

## **void Op\_Fire\_Lim (void)**

|  |  |
| --- | --- |
| 기능 | 파이어 행동반경 |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) | Void Disp\_Ini (void) |
| 사용하는  외부변수 및 flag | Struct Ob2 Fire[MAX\_FIRE\_NUM]  int stagefire |
| 비고 |  |

## **void Op\_Heart\_Lim (void)**

|  |  |
| --- | --- |
| 기능 | 하트 행동반경 |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) | Void Disp\_Ini (void) |
| 사용하는  외부변수 및 flag | Struct Ob3 Heart[MAX\_HEART\_NUM] |
| 비고 |  |

# 전역변수(flag) 및 상수 정의

프로그램이 사전에 기획이 잘 되어 있다면 main layer에서 사용할 전역변수들이 미리 정의가 다 되어 있어야 한다 따라서 Main Layer의 함수들간에 서로 공유하면서 사용할 전역변수, 상수정의 등이 미리 완성되어 있어야 한다

특히 이 전역변수들은 어느함수가 사용(참조)하고 어느 함수가 변경하는지에 대한 사전 기획과 기술이 반드시 필요하다

<예시>

## **상수 정의**

상수는 말그대로 프로그램에 사용될 값들을 사전 정의하는 것으로 작성할 프로그램의 기획이 잘 되어 있다면 상수 정의도 미리 완료가 되어 프로그램 본문에서는 이 상수 정의들을 사용하여 개발할 수 있게 된다 가급적 코드 자체에 상수값이 직접 쓰이는 경우는 자제할 필요가 있다

|  |  |  |  |
| --- | --- | --- | --- |
| No. | 변수명 | 값 | 설명 |
| 1 | #define MAX\_MON\_NUM | 10 | MON 최대 개수 |
| 2 | #define MAX\_HEART\_NUM | 5 | 하트 슈터 최대 개수 |
| 3 | #define MAX\_FIRE\_NUM | 5 | 파이어 볼 최대 개수 |
| 4 | #define MAX\_COIN\_NUM | 10 | 매직코인 최대 개수 |
| 5 | #define Width | 320 | 가로 |
| 6 | #define Height | 240 | 세로 |

이 상수 정의들과 사용하는 함수들의 선언은 모두 해더파일에 작성하여 두는 것이 좋다 물론 소스코드에서는 이 헤더파일을 #include 하여 사용한다

사용하는 typedef이나 구조체, 공용체의 형틀은 반드시 사전에 기획하여 문서에 기록하여야 한다 (주로 mytypes.h 등의 문서로 만들어서 공유한다)

## **전역변수 정의**

전역변수는 주로 Main Layer의 함수들이 직접 사용하는 변수들이다 구조체 또는 배열 뿐 아니라 일반 변수들도 많이 정의가 된다 가급적 이 변수들을 직접 사용하는 계층은 Main Layer로 한정하는 것이 좋고 Sub Layer에서 이 변수들을 사용할 경우는 call ba address나 parameter로 전달하는 것이 좋은 구조이다

특히 이 변수들은 어느 함수에서 사용(참조)하고 어느 함수에서 값을 변경하는 지를 잘 적어 두어야 나중에 프로그램 동작 디버깅시 유리하다 물론 변수를 사용하는 함수들이 매우 많을 경우는 기술이 어렵겠지만 이 변수들을 직접 사용하는 Main Layer 함수들은 꼭 기록해 두어야 한다 이것이 기록되어 있지 않다면 사전 기획 없이 프로그램을 설계한 것으로 판단할 수 밖에 없다

만약 main layer 함수에서 이 변수들을 직접 변경하는 일은 없지만 sub layer의 함수를 호출하면서 call by address로 전달하여 간접적으로 값을 변경할 수도 있는 경우라면 이 경우는 main layer 함수에서 이 값을 변경하는 것으로 보아야 한다. 따라서 변수의 변경 주체에 반드시 main layer 함수명을 기입하여야 한다

다만, Sub Layer의 경우 이 변수들을 paramter나 call by address로 전달 받는 경우는 참조, 변경에 대하여 기록을 남길 필요는 없으나 만약 Sub Layer 함수들에서도 이 변수들의 값을 직접 변경하는 경우가 있다면 반드시 이를 기술 하여야 한다

만약 프로그램 설계중 이 변수들의 의미가 변경되거나 기존 변수가 제거되거나 아니면 새로운 변수가 추가되어야 하는 경우가 있다면 반드시 문서를 개정한 이후 개발을 진행하여야 한다 그리고 이 문서가 개정될 때마다 Ver을 기입하여 기존 문서와 신규 문서의 비교가 가능하도록 하는 습관이 필요하다

구조체, 공용체 등의 경우 반드시 그 형식을 미리 문서에 기록하여 둬야 한다

이 형식이 여러 파일에서 사용되는 경우 반드시 my\_types.h 등을 만들어서 typedef과 함께 관리하여야 한다

|  |  |  |
| --- | --- | --- |
| No. | 변수명 | 기능 |
| 참조 주체 | 변경 주체 |
| 1 | Struct Ob1 Nana | 나나 관리 |
| GM, OM, DM | GM, OM, (충돌시에만 DM) |
| 2 | Struct Ob2 Mon [MAX\_MON\_NUM] | 몬스터 객체 관리 |
| GM, OM, DM | DM, OM, (stage renew 시에만 GM) |
|  |  |  |
| 3 | Struct Object Fire[MAX\_FIRE\_NUM] | 파이어 볼 객체 관리 |
|  | GM, OM, DM | DM, OM, (stage renew 시에만 GM) |
| 4 | Struct Object Heart[MAX\_HEART\_NUM] | 하트 슈터 객체 관리 |
|  | GM, OM, DM | DM, OM, (stage renew 시에만 GM) |
| 5 | Struct Object Coin[MAX\_COIN\_NUM] | 매직코인 객체관리 |
|  | GM, OM, DM | DM, OM, (stage renew 시에만 GM) |
| 6 | int Stage\_Num | 스테이지 넘버 |
|  | GM, OM, DM | OM GM |
| 7 | int score | 스코어 관리 |
|  | GM, OM, DM | OM GM |
| 8 | Int stagefire | 스테이지 마다 파이어볼 관리 |
|  | GM, OM, DM | OM GM |
| 9 | Int stagemon | 스테이지 마다 몬스터 관리 |
|  | GM, OM, DM | OM GM |
| 10 | Int stagecoin | 스테이지 마다 코인 관리 |
|  | GM, OM, DM | OM GM |
| 11 | Int hearts | 하트 개수 관리 |
|  | OM, DM | OM GM |
| 12 | Int e | 배경 이미지 관리 |
|  | OM, DM | OM GM |
| 13 | Int heartcnt | 하트 슈터 불렛 카운트 |
|  |  |  |
| 14 |  |  |
|  |  |  |

참고 : OM, DM, GM 등은 Main Layer의 함수 약어 표시임

## **flag 정의**

flag은 전역변수와 동일하지만 주로 어떤 프로그램의 운용 상태를 나타내는 용도로 사용된다 특별히 분류가 애매하면 전역변수에 추가하면 된다

|  |  |  |
| --- | --- | --- |
| No. | 변수명 | 기능 |
| 참조 주체 | 변경 주체 |
| 1 | int Game\_start\_flag | Game이 실행 상태 표시  1: 실행중 0. 대기중(디폴트) |
| GM, OM | GM, OM |
| 2 | int Display\_flag | 화면 갱신 필요 표시  1. 화면 갱신 필요  0. 갱신 필요 없음(디폴트) |
| GM, OM | OM |
| 3 | Int Stage\_flag  GM, OM | 지지 않았을 경우 다음 스테이지로 넘어감  0 start initial (디폴트)  1 move to stage 1  2 move to stage 2  3 move to stage 3  4 move to stage 4  5 move to stage 5  6 end game (win)  OM |
| 4 | Int Lose\_flag  GM, OM | 졌는가 판단 플래그  (각 스테이지 당 3 생명이 다 떨어졌을 시)  0 지지 않음 (디폴트)  1 짐  리스타트는 졌을 시 시작 부터? 그 스테이지 부터?  OM |
| 5 | Int Win\_flag  GM, OM | 이겼는가 판단 플래그  Stage 5 에서 이길 경우 (stage\_flag 6)일 경우  0 이기지 않음 (디폴트)  1 이김  OM |

* 프로그램의 파일이 여러 개로 작성되는 경우 상수선언, 함수선언은 반드시 헤더파일로 만들어서 #include하여 공유하도록 한다
* 전역변수와 flag은 헤더로 만드는 것 보다는 main.c나 그 변수를 주로 사용하는 파일에 선언하고 다른 파일에서는 extern으로 선언하여 사용하는 것이 좋다 다만, extern으로 선언한 파일에서는 절대 이 변수를 초기화 하여서는 안된다