***상위 설계서***

***(Top Level Design)***

***구현완료 버전***

나나 존스와 보물성

Ver. 1.00 (구현 완료)

반: IMCE 3반

이름: 김민경

**[ 목차 ]**

[1. 프로그램 개발 전략 4](#_Toc364449214)

[2. State Transition Diagram 5](#_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 9](#_Toc364449220)

[4.1. void Init\_Game\_Data(void) 9](#_Toc364449221)

[4.2. void Stage\_Ini (void) 9](#_Toc364449222)

4.3 void Op\_Nana (void)…………………………………………………10

4.4 void Op\_Fire (void)…………………………………………………..10

4.5 void Op\_Mon(void)………………………………………………….10

4.6 void Op\_Heart (void)…………………………………………………11

4.7` void Op\_Mon\_Heart (void)……………………………………………11

4.8 void Op\_Fire\_Nana(void)……………………………………………..11

4.9 void Op\_Mon\_Nana(void)……………………………………………12

4.10 void Op\_Coin\_Nana(void)…………………………………………..12

4.11 void Disp\_Fire (void)…………………………………………………..12

4.12 void Disp\_Mon (void)………………………………………………….13

4.13 void Disp\_Heart (void)………………………………………………..13

4.14 void Disp\_Coin (void)…………………………………………………13

4.15 void Disp\_Nana(void)…………………………………………………13

4.16 Void Disp\_Stage (void)………………………………………………..14

4.17 void Disp\_Lose (void)…………………………………………………..14

4.18 Void Disp\_Win (void)…………………………………………………..15

4.19 Void Disp\_Ini (void)………………………………………………………15

4.20 void Disp\_Info(void)…………………………………………………15

4.21 void Op\_Nana\_Lim(void)………………………………………………..16

4.22 void Op\_Mon\_Lim(void)………………………………………………16

4.23 void Op\_Fire\_Lim (void)………………………………………………16

4.24 void Op\_Heart\_Lim (void)……………………………………………..16

[5. 전역변수(flag) 및 상수 정의 18](#_Toc364449223)

[5.1. 상수 정의 18](#_Toc364449224)

[5.2. 전역변수 정의 18](#_Toc364449225)

5.3 FLAG 정의…………………………………………………………………..19

5.4 STRUCT DEF …………………………………………………………….20-21

# 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.
* 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.
* 나나존스는 레벨이 있습니다. 코인을 모으거나, 몬스터를 다 죽이거나 해야합니다. 그러므로 더욱더 강화된 player enjoyment 와 achievement를 강조합니다.

***1.2 Graphical Considerations***

* 새로 쓰는 알고리즘은 그래픽도 고려해야 합니다. 구조체로 아이템을 만드는 것과 아이템과 다른 객체들의 interaction 도 고려했음을 말씀 드립니다 (새로 배운 Double Buffering test)
* 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. 레벨 1은 보물 3개, 두 가지 종류의 적 각각 2마리 지만, 레벨이 높아질수록 보물과 적의 수가 더 많아 집니다.
* 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

시작

게임 상태

출력

동작

If Stage\_flag!=0 (Game in-Progress)

If Stage\_flag==0 (Game is not in Progress/default/start value)

If OM signals for game over, the game shuts down from the main level.

In other words, GM is in the main layer in a for loop, and GM and OM are within GM’s for loop.

Win/Lose flags are also used

# Main Layer 함수 정의

## **void Game\_Manager(void)**

|  |  |
| --- | --- |
| 기능 | 모든 변수의 초기화 (시작/스테이지 등등)  스테이지 판정 (새로운 스테이지 넘어가기)  Background image 관리  게임의 시작/끝 여부와 관리 |
| Parameter | void |
| Return Value | void |
| 호출 함수  (Sub Layer 함수) | void Init\_Game\_Data(void)  void Stage\_Ini(void)  void Lcd\_Draw\_BMP()  void Disp\_Win()  void Disp\_Lose()  void Timer3\_Buzzer\_Beep()  Void lcddrawbackgr (int x, int y, int x2, int y2, const unsigned short int \*fp) |
| 외부 변수 | int Game\_start\_flag  int Stage\_flag  int Stage\_Num  int Lose\_Flag  int Win\_Flag  int heartcnt  int monlife  Int coinlife  int sangtae    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]  const unsigned short \* Nanaim[] = {Nanaleft, Nanaright, Nanaup, Nanadown};  const unsigned short \* Heartim[] = {heartleft, heartright, heartup, heartdown};  const unsigned short \* Monim[] = {Mon1, Mon2, Mon3, Mon4};  const unsigned short \* levelim[] = {level1, level2, level3, level4, level5}; |
| 비고 |  |

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

|  |  |
| --- | --- |
| 기능 | 화면 업데이트  모든 객체들의 위치 이미지 표시와 갱신  점수 및 라이프 표시 업데이트 |
| 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\_Ini (void)  void Disp\_Info(void)  Void lcddrawbackgr (int x, int y, int x2, int y2, const unsigned short int \*fp) |
| 사용하는  외부변수 및 flag | int Display\_flag  int Stage\_Num  int Stage\_flag  int score  Int Lose\_Flag  Int Win\_Flag  Int monlife  Int coinlife  int heartcnt  int sangtae  //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]  const unsigned short \* Nanaim[] = {Nanaleft, Nanaright, Nanaup, Nanadown};  const unsigned short \* Heartim[] = {heartleft, heartright, heartup, heartdown};  const unsigned short \* Monim[] = {Mon1, Mon2, Mon3, Mon4};  const unsigned short \* levelim[] = {level1, level2, level3, level4, level5}; |
| 비고 |  |

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

|  |  |
| --- | --- |
| 기능 | 물체 움직임 위치 갱신 및 이미지 플래그 갱신  충돌 처리 (나나/몬스터, 하트/몬스터, 나나/코인)  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)  Void lcddrawbackgr (int x, int y, int x2, int y2, const unsigned short int \*fp) |
| 사용하는  외부변수 및 flag | int Display\_flag  int Game\_start\_flag  int Stage\_flag  int Stage\_Num  int score  int heartcnt  int monlife  Int coinlife  int sangtae  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]  const unsigned short \* Nanaim[] = {Nanaleft, Nanaright, Nanaup, Nanadown};  const unsigned short \* Heartim[] = {heartleft, heartright, heartup, heartdown};  const unsigned short \* Monim[] = {Mon1, Mon2, Mon3, Mon4};  const unsigned short \* levelim[] = {level1, level2, level3, level4, level5}; |
| 비고 |  |

# Sub Layer 함수 정의

## **void Init\_Game\_Data(void)**

|  |  |
| --- | --- |
| 기능 | 게임 초기화 (모든 객체/구조체 다) |
| 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 sangtae  int e  int heartcnt  int coinlife  int monlife  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) | Rand() |
| 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  int sangtae  //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) |  |
| 호출 함수  (디바이스 드라이버) |  |
| 사용하는  외부변수 및 flag | Struct Ob2 Fire[MAX\_FIRE\_NUM] |
| 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) |  |
| 호출 함수  (디바이스 드라이버) |  |
| 사용하는  외부변수 및 flag | Struct Ob2 Mon[MAX\_MON\_NUM] |
| 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] |
| 비고 | Button 5 shoots |

## **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 | 포인트 증가 |

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

|  |  |
| --- | --- |
| 기능 | 파이어볼 이미지 출력 |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) | Void lcddrawbackgr() |
| 호출 함수  (디바이스 드라이버) | Void LCD\_Draw\_BMP (void)  Void lcddrawbackgr (int x, int y, int x2, int y2, const unsigned short int \*fp)  Void Lcd\_Set\_Shape\_Mode(int, COLOR) |
| 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 lcddrawbg (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 lcddrawbackgr (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 lcddrawbg (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 lcddrawbg (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 lcddrawbg (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 | I don’t think the function goes through… |

## **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) |
| 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)  Timer3\_Buzzer\_Beep()  Timer4\_Delay() |
| 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)  Timer4\_Delay() |
| 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\_Printf() |
| 사용하는  외부변수 및 flag | Int score  Int Nana.life  Int Heart.life  Int Stage\_Num |
| 비고 |  |

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

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

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

|  |  |
| --- | --- |
| 기능 | 몬스터 행동 반경 + 스피드/색깔 플래그 변경 |
| 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]  Int Level\_Num |
| 비고 |  |

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

|  |  |
| --- | --- |
| 기능 | 파이어 행동반경 +스피드 변경 |
| Parameter | Void |
| Return Value | Void |
| 호출 함수  (Sub Layer 함수) |  |
| 호출 함수  (디바이스 드라이버) | Rand() |
| 사용하는  외부변수 및 flag | Struct Ob2 Fire[MAX\_FIRE\_NUM]  Int Level\_Num |
| 비고 |  |

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

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

# 전역변수(flag) 및 상수 정의

## **상수 정의**

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |
| 7 | #define START | 0 | Game start (default) |
| 8 | #define INPLAY | 1 | Game in play |
| 9 | #define NEWSTAGE | 2 | Game should be ready for new stage |

## **전역변수 정의**

|  |  |  |
| --- | --- | --- |
| 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 e | 배경 이미지 관리 |
|  | OM, DM | OM GM |
| 9 | Int heartcnt | 하트 슈터 불렛 카운트 |
|  | OM, GM, DM | OM, GM, |
| 10 | Int coinlife | 코인수 관리 (스코어 용) |
|  | OM, GM, DM | OM, GM |
| 11 | Int monlife | 몬스터 수 관리 (스코어 용) |
|  | OM, GM, DM | OM, GM |

## **flag 정의**

|  |  |  |
| --- | --- | --- |
| 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 |
| 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  OM |
| 5 | Int Win\_flag  GM, OM | 이겼는가 판단 플래그  Stage 5 에서 이길 경우  0 이기지 않음 (디폴트)  1 이김  OM |

## **5.2 구조체 정의**

STRUCT OB 1 NANA

|  |  |  |
| --- | --- | --- |
| No. | 이름 | 기능 |
| Int Timer | Timer |
|  | Int Moveflag | Did Nana move? Draw flag |
|  | Int Pos [2] | Position current |
|  | Int ini[2] | Initial position |
|  | Int pre[2] | Previous position |
|  | Int size[2] | Size of Nana |
|  | Int move\_step | Temp color |
|  | int life | Life |
|  | Int dir | Which direction is Nana moving |
|  | Unsigned col | Temp color |
|  | Int im | Unused |
|  | Int hit | Image of Nana |

STRUCT OB 2 MON and FIRE

|  |  |  |
| --- | --- | --- |
| No. | 이름 | 기능 |
| Int Timer | timer |
|  | Int moveflag | Did mon/fire move? Drawflag |
|  | Int pos[2] | Position current |
|  | Int ini[2] | Initial position |
|  | Int pre[2] | Previous position |
|  | Int size[2] | Size of Mon/Fire respecitively |
|  | Unsigned short color | Temp color |
|  | Int speed | Speed |
|  | Int movestep | movestep |
|  | Int movex | Which direction to move x axis |
|  | Int movey | Which direction to move y axis |
|  | Int life | Life |
|  | Int col | Color/im control |
|  | Int hit | Unused |

STRUCT OB 3 HEART

|  |  |
| --- | --- |
| 이름 | 기능 |
| Int Timer | Timer |
| Int moveflag | Did Heart move?? |
| Int pos[2] | Current position |
| Int pre[2] | Previous position |
| Int size[2] | size |
| Int ini[2] | Initial pos |
| Unsigned short color | Temp color |
| Int Speed | Speed |
| Int movestep | movestep |
| Int cnt | To check if bullet used or not |
| Int dir | Initial bullet direction |
| Int a | Controls image and direction of bullets |
| Int hit | unused |

STRUCT OB 4 COIN

|  |  |  |
| --- | --- | --- |
| No. | 이름 | 기능 |
| Pos[2] | position |
|  | Size[2] | Size |
|  | Col | unused |
|  | life | Life coin (draw flag, basically) |
|  | Unsigned short color | Temp color |
|  | hit | Unused |

There were separate entities of const unsigned short \* img[], but I used what was provided to me previously.

Enum used from given set.