동방 플라이트 ver 1.0

부제 : 이거 깨면 3만원 드림

팀 천호) 김민섭 이도규 하동호

동방 플라이트 ver 1.0

1. 게임 인터페이스

2. 게임 설명

3. 구현 클래스

4. 탄환 패턴 구조

5. Issue

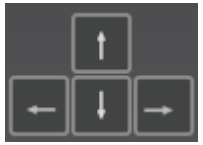
6. 타임 차트

1. 게임 인터페이스

화면 해상도

```
#define WINSIZE_X      1280
#define WINSIZE_Y      960
#define GAMESIZE_X     768
#define GAMESIZE_Y     896
```

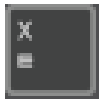
인터페이스



움직임 제어

플레이어 속도 감소

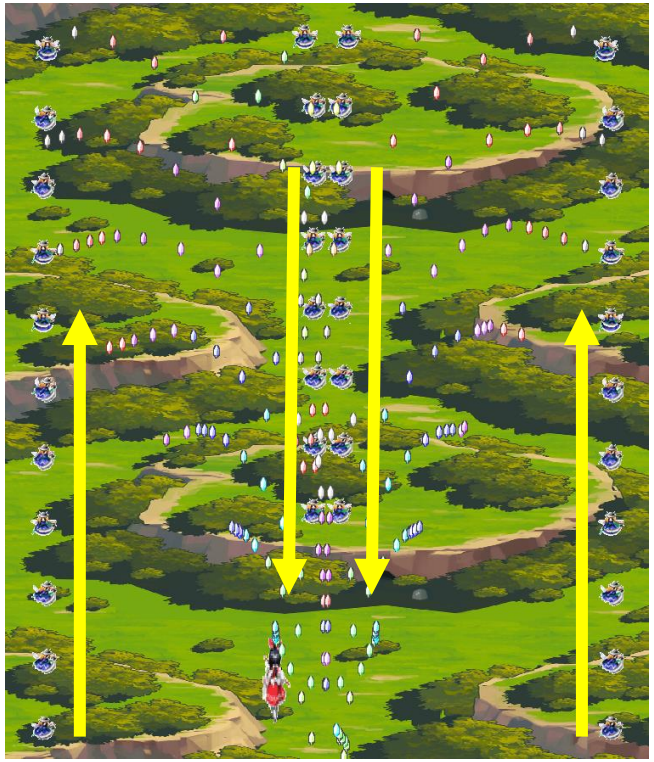
탄 발사



폭탄

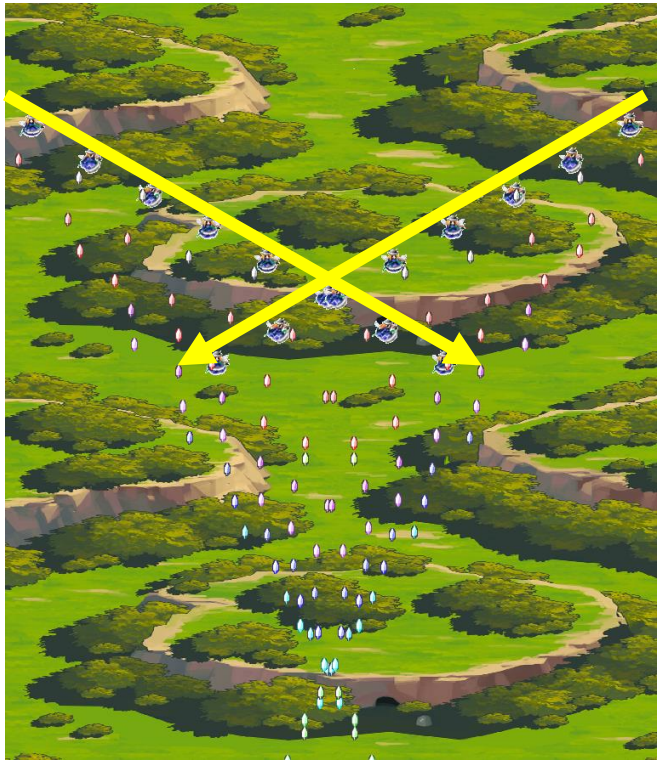
2. 게임 내용

게임 내용 → 게임 화면 및 로고 | 플레이어 탄 및 폭탄 | 몬스터 패턴 4개 | 보스 패턴 7개



몬스터 패턴 1

몬스터 패턴 2



몬스터 패턴 3

유도탄

2. 게임 내용

몬스터 패턴 4

좌우 발사

보스 패턴 1

물결 탄 / 산탄

보스 패턴2

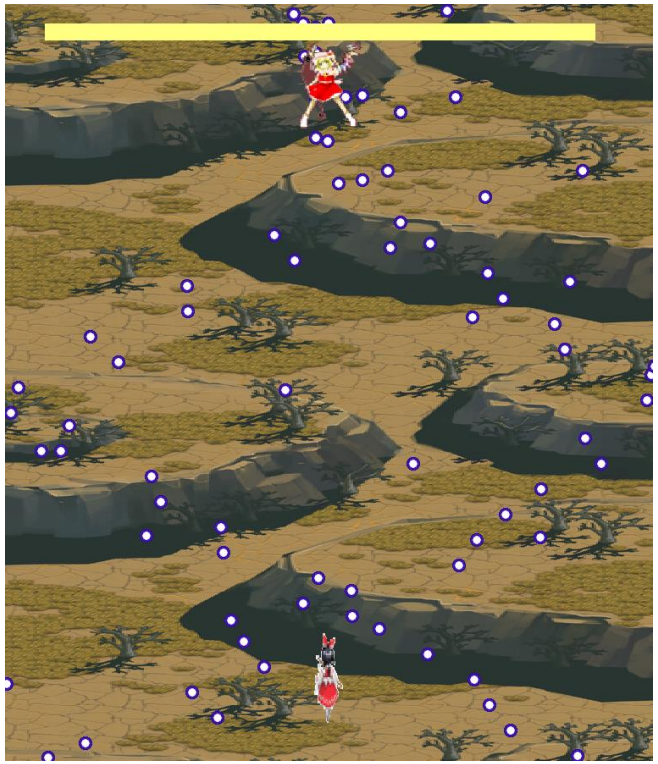
전방위 고속탄

랜덤 이동

2. 게임 내용

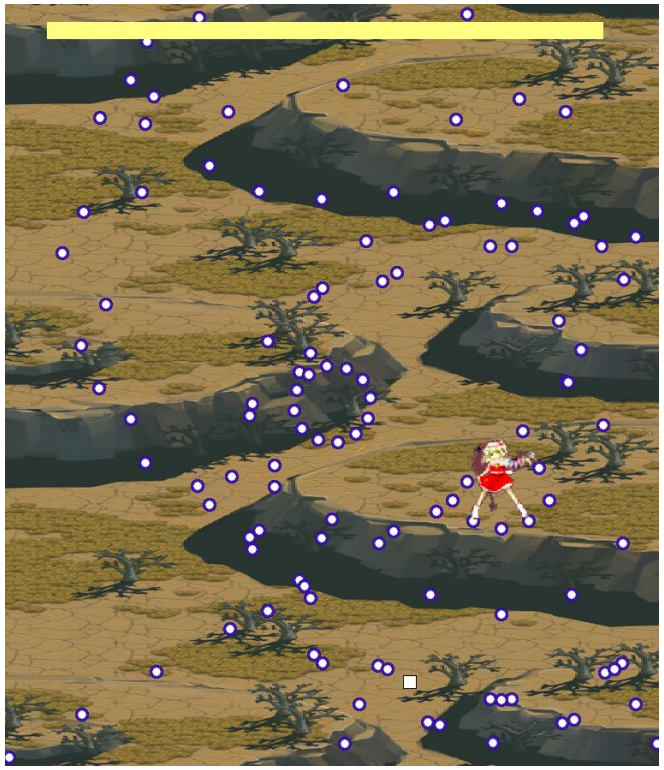
보스 패턴 3

대기 고속 유도탄



보스 패턴 4

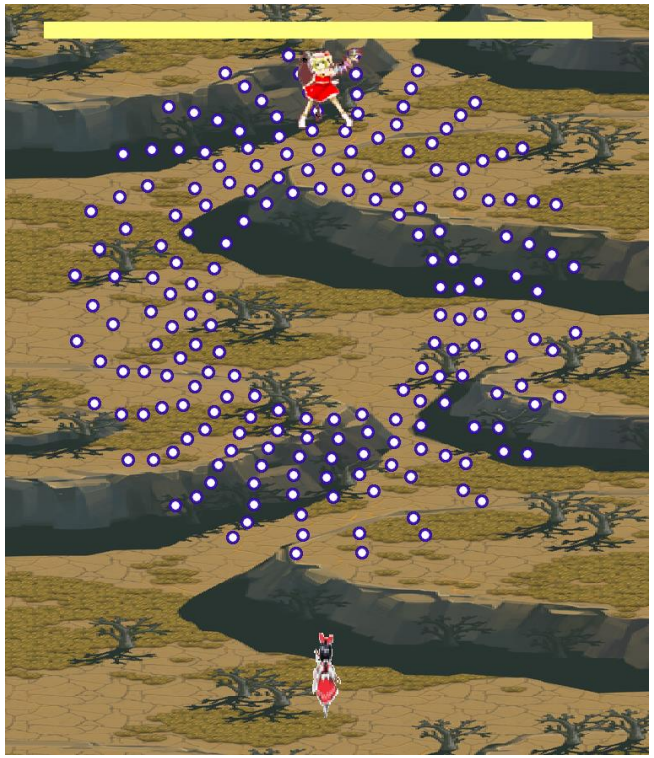
진눈깨비 탄



보스 패턴5

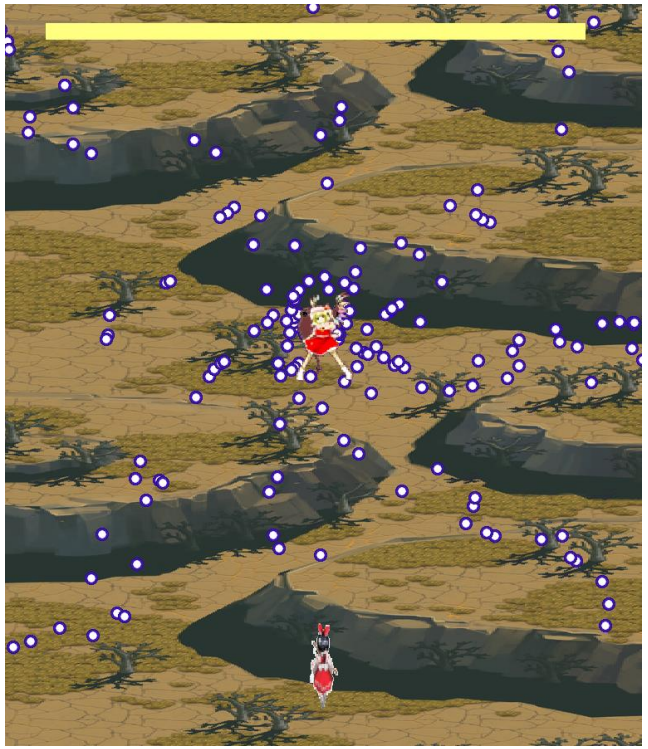
중력 자탄

2. 게임 내용



보스 패턴 6

확산탄



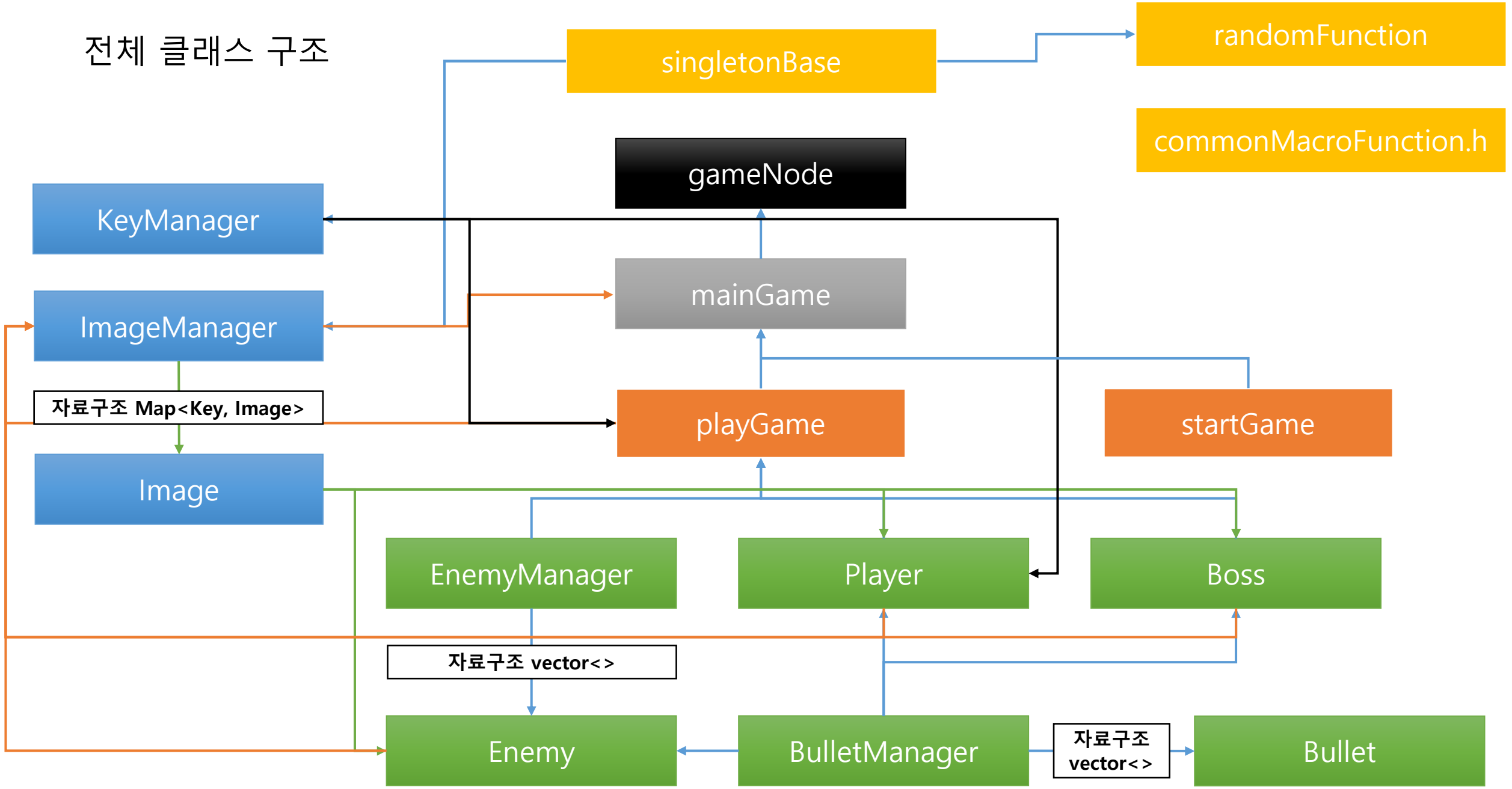
보스 패턴 7

쥐불놀이 탄

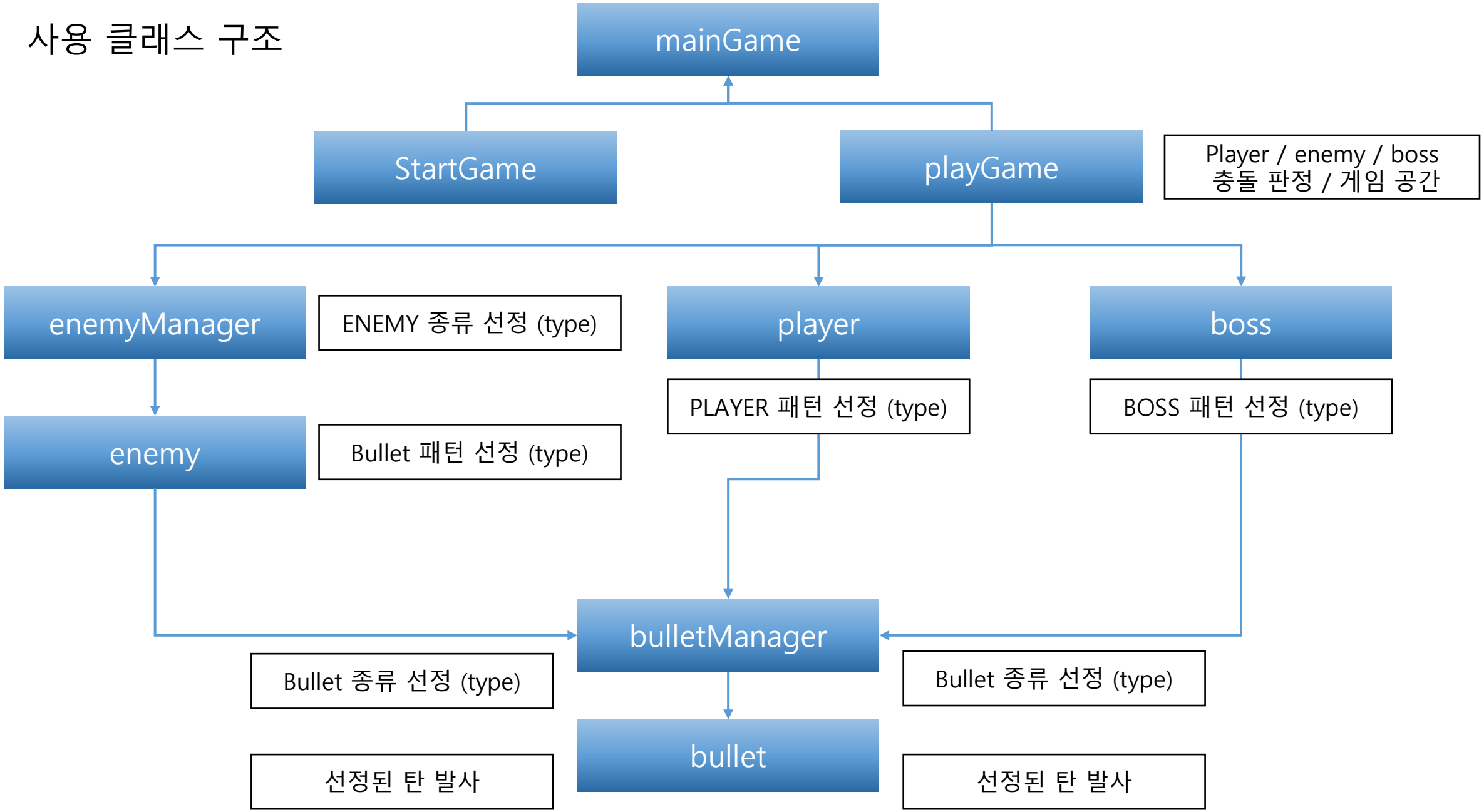
Ver 1 | Ver 2 | Ver 3

플레이어 폭탄

전체 클래스 구조

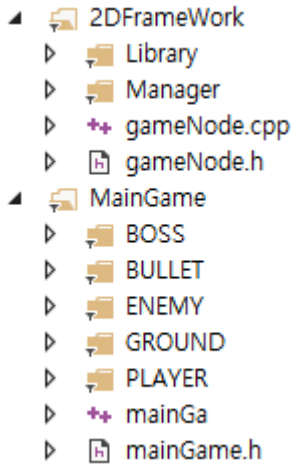

singletonBase
randomFunction
commonMacroFunction.h
gameNode
KeyManager
mainGame
ImageManager
자료구조 Map<Key, Image>
playGame
startGame
Image
EnemyManager
Player
Boss
자료구조 vector<>
Enemy
BulletManager
자료구조
vector<>
Bullet

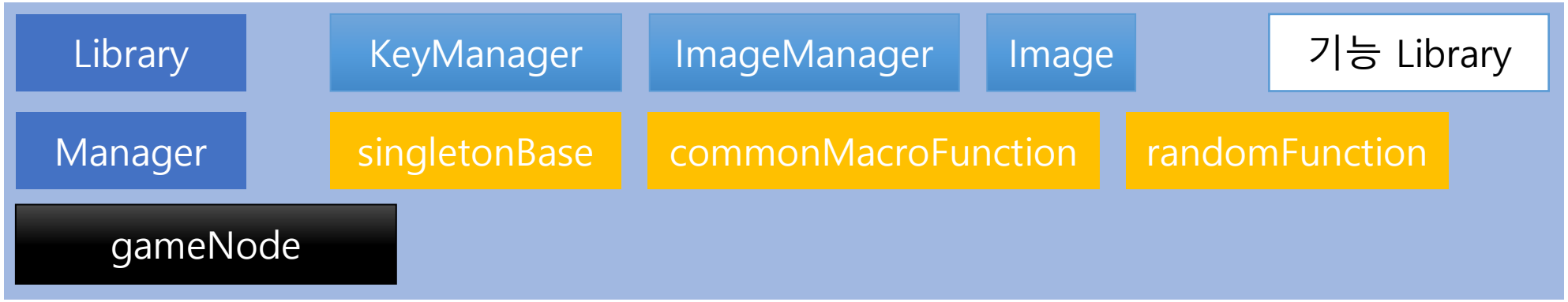
사용 클래스 구조

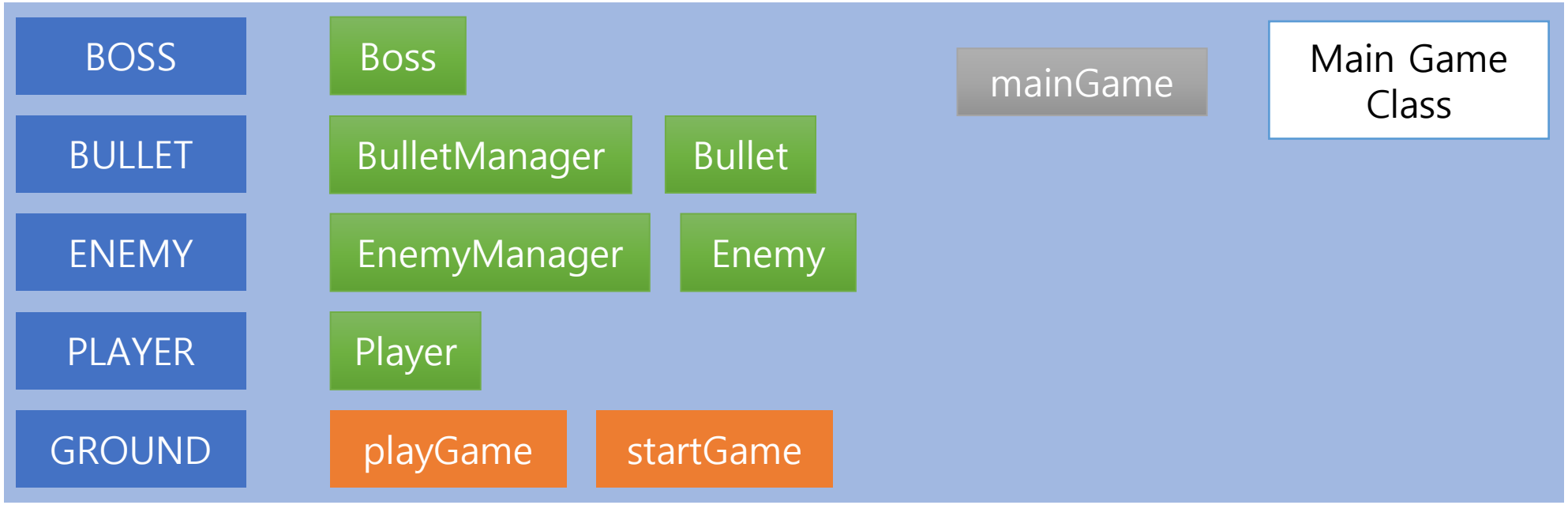

mainGame
StartGame
playGame
Player / enemy / boss
충돌 판정 / 게임 공간
enemyManager
ENEMY 종류 선정 (type)
player
boss
enemy
Bullet 패턴 선정 (type)
PLAYER 패턴 선정 (type)
BOSS 패턴 선정 (type)
bulletManager
Bullet 종류 선정 (type)
Bullet 종류 선정 (type)
bullet
선정된 탄 발사
선정된 탄 발사

클래스 세부 구조

사용 클래스 구조







4. 탄환 패턴 구조

Player / Monster / Boss

```
bool isAlive;               // 생명

float posX;              // X 좌표 위치
float posY;              // Y 좌표 위치
float sizeX;                // X 크기
float sizeY;                // Y 크기
float collisionX;           // 피격판정 X
float collisionY;           // 피격판정 Y
float moveSpeed;         // 플레이어 속도


RECT rc;            // 피격 판정 RECT

int time;           // 타이머
```

Player 중요 변수

```
bool isAlive;    // 생명

float posX;      // X 좌표
float posY;      // Y 좌표
float sizeX;     // X 크기
float sizeY;     // Y 크기

float angle;     // 움직이는 방향
float moveSpeed;// 움직이는 속도

// 유도형 변수
float targetX;  // 표적 X 좌표
float targetY;  // 표적 Y 좌표
// 총알 발사 변수
int time;        // 탄환 발사 시간
```

Monster 중요 변수

```
int hp;             // 보스 체력

bool isAlive;       // 생명

float posX;         // X 좌표
float posY;         // Y 좌표
float sizeX;        // X 크기
float sizeY;        // Y 크기

float angle;        // 움직이는 각도
float moveSpeed;    // 움직이는 속도


float targetX;      // 표적 X 좌표
float targetY;      // 표적 Y 좌표

int time;           // 총알 발사 시간
```

Boss 중요 변수

4. 탄환 패턴 구조

Player / Monster / Boss

```
EnemyFire(MONSTER1, MONSTERPATTEN101, 50, GAMESIZE_Y + 100, 90, 5);
```

몬스터 이름 | 움직임 패턴 | 생성 좌표 | 각도, 속도

Boss

보스의 움직임을 담당한다
.

```
void BossMove1();
void BossMove2();
void BossMove3();
void BossMove4();
void BossMove5();
void BossMove6();
void BossMove7();
void BossMove8();
void BossMove9();
void BossMove10();
```

EnemyManager

움직이는 방식 설정

FireMgr(BOSSPATTEN1);

움직이면서 탄 발사

```
// 몬스터 패턴 : ENEMY
#define MONSTERPATTEN_SIZE 6
#define MONSTERPATTEN101 101
#define MONSTERPATTEN102 102
#define MONSTERPATTEN103 103
#define MONSTERPATTEN103_R 1031
#define MONSTERPATTEN103_L 1032
#define MONSTERPATTEN104 104
```

Enemy

몬스터의 움직임을
담당한다

```
void MonsterMove101();
void MonsterMove102();
void MonsterMove103();
void MonsterMove104();
```

4. 탄환 패턴 구조

BulletManager

탄 발사 함수 → 몬스터가 BULLET1 총알을 260도의 각도에 8의 속도로 발사한다.

```
EnemyFire(BULLET1, MONSTER_BULLET, 260, 8);
```

- BULLET1: 발사할 총알
- MONSTER_BULLET: 총알을 쏘는 대상
- 260, 8: 각도 / 속도

BulletManager → Bullet

```
// 플레이어 패턴 : PLAYER
#define PLAYERPATTEN1 1
// 몬스터 패턴 : ENEMY
#define MONSTERPATTEN_SIZE 6
#define MONSTERPATTEN101 101
#define MONSTERPATTEN102 102
#define MONSTERPATTEN103 103
#define MONSTERPATTEN103_R 1031
#define MONSTERPATTEN103_L 1032
#define MONSTERPATTEN104 104
```

```
// 보스 패턴 : BOSS
#define BOSSPATTEN1 201
#define BOSSPATTEN2 202
#define BOSSPATTEN3 203
#define BOSSPATTEN4 204
#define BOSSPATTEN5 205
#define BOSSPATTEN6 206
#define BOSSPATTEN7 207
#define BOSSPATTEN8 208
#define BOSSPATTEN9 209
```

플레이어 / 몬스터 / 보스가
총알을 쏘는 패턴에 관한 함수 선정

```
// 총알의 종류
#define BULLET1 1          // 방향탄
#define BULLET2 2          // 유도탄
#define BULLET3 3          // y 400값에 멈추는탄
#define BULLET4 4          // 회전탄
#define BULLET5 5          // 유도탄// 고속

#define BULLET52 52        // 확산탄
#define BULLET53 53        // 특수탄
#define BULLE54 54         // 진눈깨비탄
#define BULLET55 55        // 중력자탄
#define BULLET56 56        // 쥐불놀이
```

총알이 움직이는 함수 선정

4. 탄환 패턴 구조

BulletManager

BulletManager

```
switch (bulletType)
{
case MONSTERPATTEN101:
    MonsterPatten101();
    break;
case MONSTERPATTEN102:
    MonsterPatten102();
    break;
```

```
void BulletManager::MonsterPatten01()          // 몬스터 아래 직진
{
    if (time % 5 == 0)
    {
        EnemyFire(BULLET1, MONSTER_BULLET, 270);
        time = 0;
    }
}

void BulletManager::MonsterPatten101()
{
    if (posY < 300)
    {
        if (time % 10 == 0)
        {
            EnemyFire(BULLET2, MONSTER_BULLET, GetAngle(posX, posY, targetX, targetY),8);
            time = 0;
        }
    }
}
```

Bullet

```
switch (bulletType)
{
case BULLET1:
    BulletType1();
    break;
case BULLET2:
    BulletType2();
    break;
```

```
void Bullet::BulletType1()  // 방향탄
{
    posX += moveSpeed * cosf((float)(DEGREE_TO_RADIAN(angle)));
    posY -= moveSpeed * sinf((float)(DEGREE_TO_RADIAN(angle)));
}

void Bullet::BulletType2()  // 유도탄
{
    int targetAngle = GetAngle(posX, posY, targetX, targetY);

    posX += moveSpeed * cosf((float)((targetAngle)));
    posY += moveSpeed * sinf((float)((targetAngle)));
}
```

함수 사용 예제

BulletType에 따라 Bullet을 발사한다.

Bullet이 BulletType에 따라 움직인다

5. Issue

1. 구조 동기화

1.1 몬스터 / 플레이어 / 보스 에 대한 세분화 ▶ **움직임(Object) / 총알 장전(Mgr) / 총알 발사(Bullet) 3가지 종류**

1.2 Bullet 을 각 객체에 전달하기 위한 방법 ▶ **Vector를 사용한 Object 전달, Get의 경우 매개변수로 Main에 전달**

2. 충돌 체크

```
Enemy* GetEnemy(int i) { return vecEnemy[i]; }
```

2.1 충돌 체크 이후 잔류 문제 ▶ **충돌 이후 객체를 멀리 보낸 이후 False**

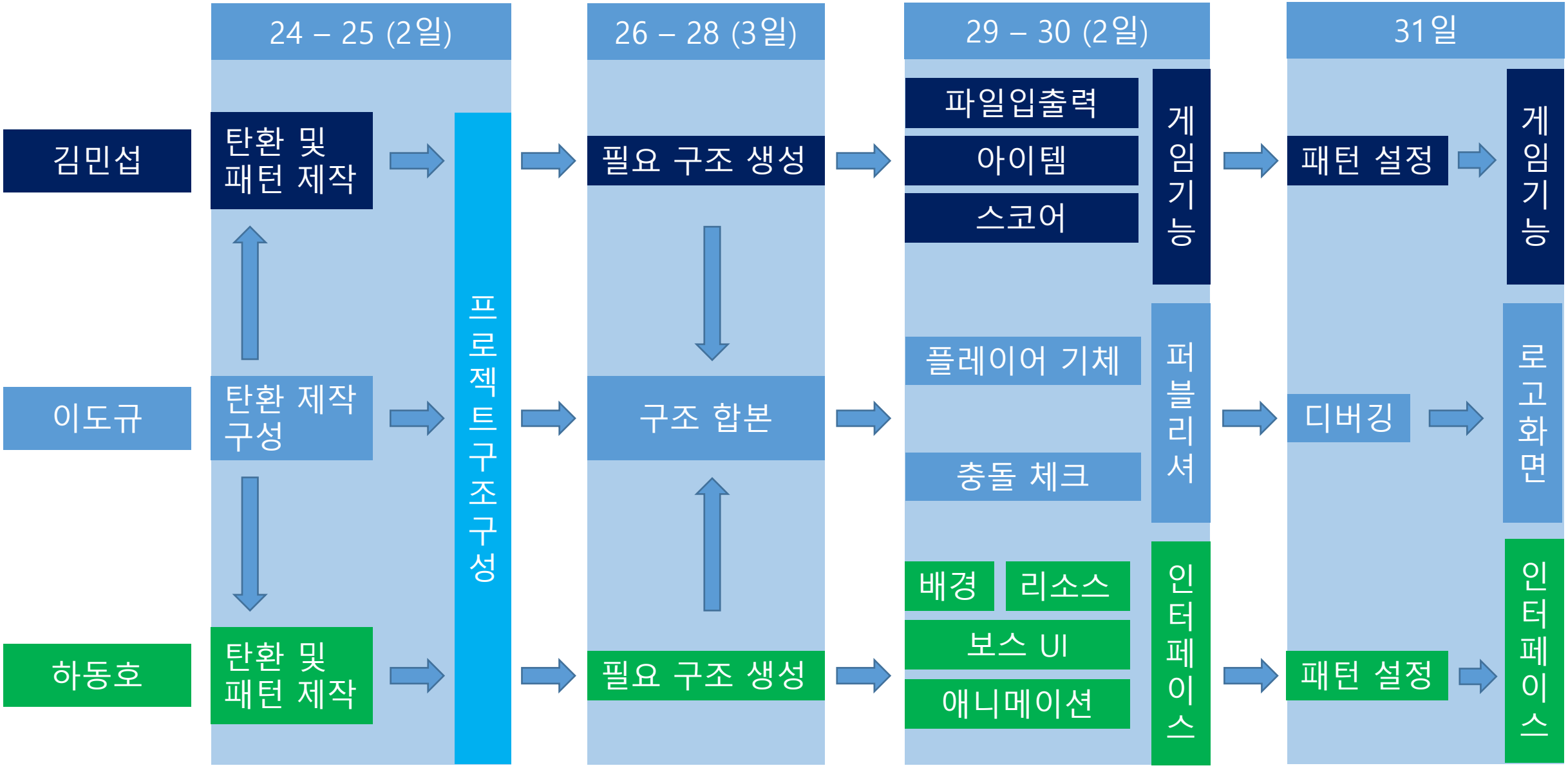
2.2 충돌에 따라 Manager의 총알 위치 값 오류 ▶ **Mgr 값이 객체값을 계속 따라다니도록 설정**

3. Time 동기화

3. 각 객체별로 Time의 의미를 치환

- ▶ **playGame** -> 저장한 패턴 함수들을 시간에 맞게 부르기 위한 Timer
- ▶ **Player** ▶ **Enemy** ▶ **Boss** -> 움직임이는 패턴을 조정하기 위한 Timer
- ▶ **EnemyMgr** ▶ **BulletMgr** ▶ -> 생성 또는 발사 타이밍을 조정하기 위한 Timer

6. 타임 차트


24 – 25 (2일)
26 – 28 (3일)
29 – 30 (2일)
31일
김민섭
이도규
하동호
탄환 및 패턴 제작
탄환 제작 구성
탄환 및 패턴 제작
프로젝트 구조 구성
필요 구조 생성
구조 합본
필요 구조 생성
파일입출력
아이템
스코어
플레이어 기체
충돌 체크
배경
리소스
보스 UI
애니메이션
게임 기능
퍼블리셔
인터페이스
패턴 설정
디버깅
패턴 설정
게임 기능
로고 화면
인터페이스

Q/A