‘BOMB Link’ returns

Object Oriented Programming
[CSED232-01]

20140231 정민욱
20180529 박진우
20180740 염재후

Contents

1. Introduction

주제 선정 동기

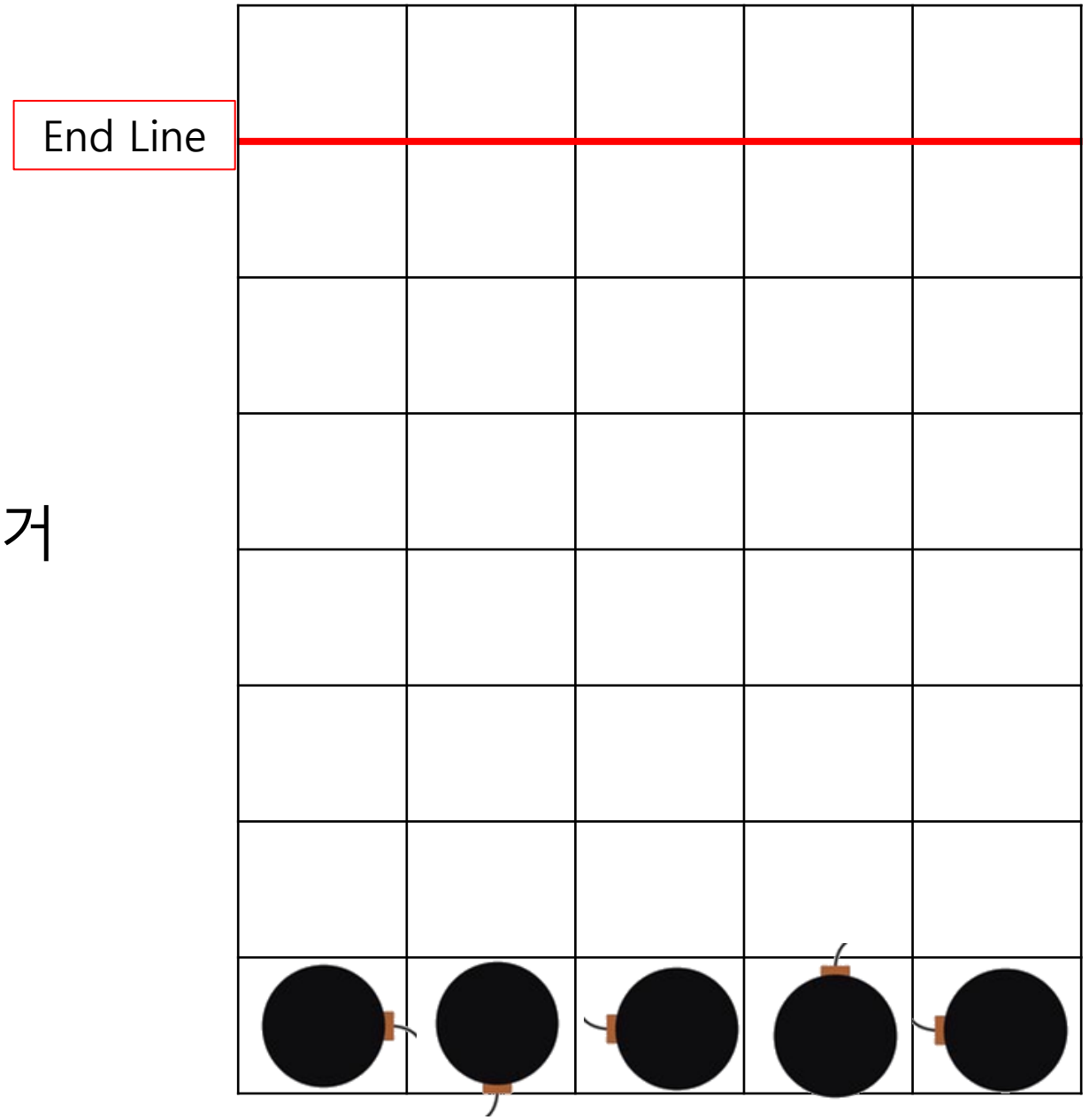
->2000년대 초반 핸드폰 내장 게임이다.
이제는 플레이 할 수 없기 때문에 게임을 만드는데
 큰 의미가 있다고 생각했다.

-> 폭탄, 불, 게임 판과 같이 다양한 객체가
 상호작용하는 이 게임을 객체지향 프로그래밍으로
잘 구현할 수 있겠다는 생각을 했다.

<u>과제 목표 :</u>
<u>객체지향 프로그래밍을 이용하여 BombLink를 구현한다.</u>

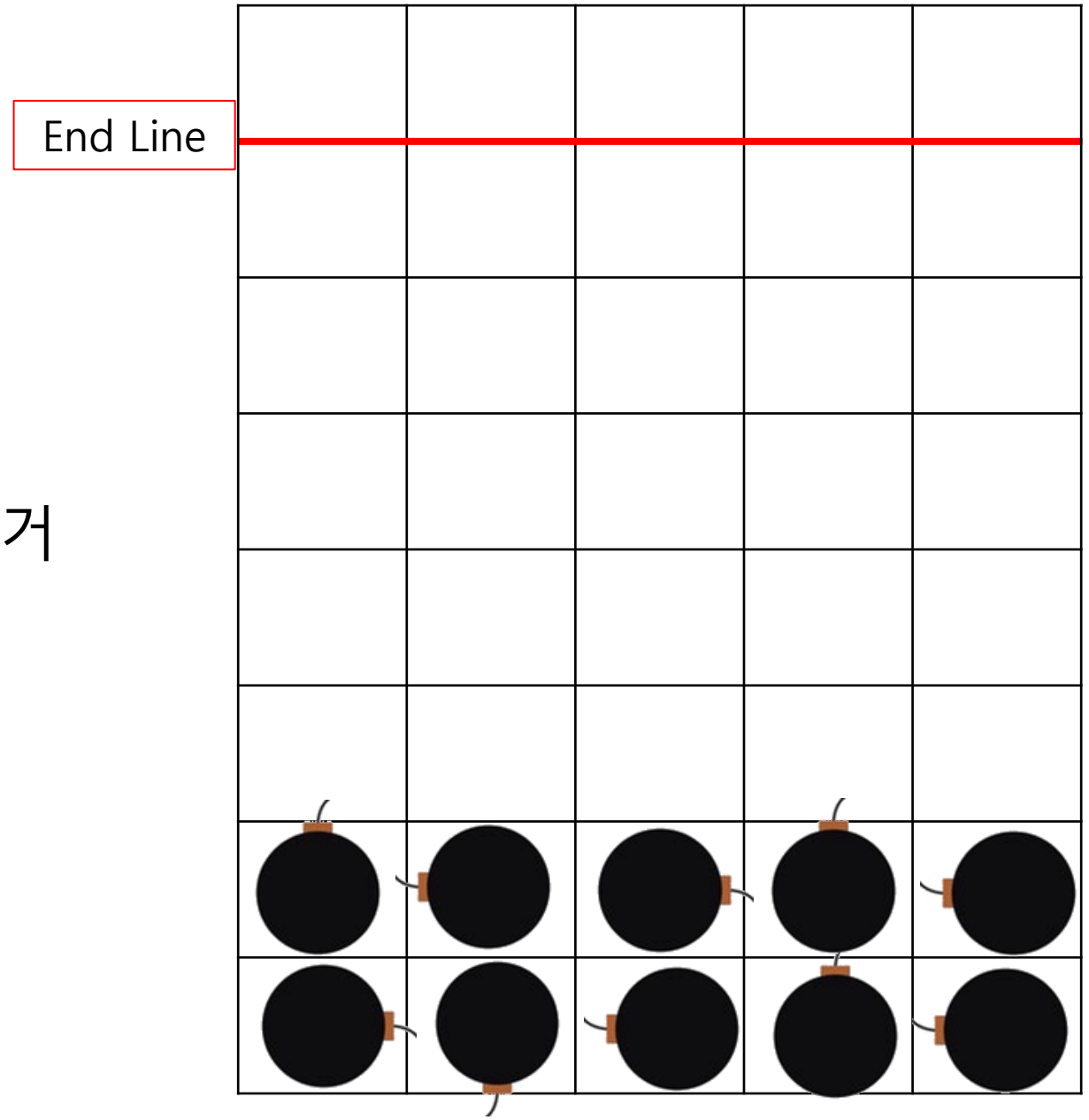
<게임 설명>

1. 폭탄 생성
2. 불 생성
3. 심지방향 조절
4. 폭탄제거
5. 연속적으로 폭탄 제거



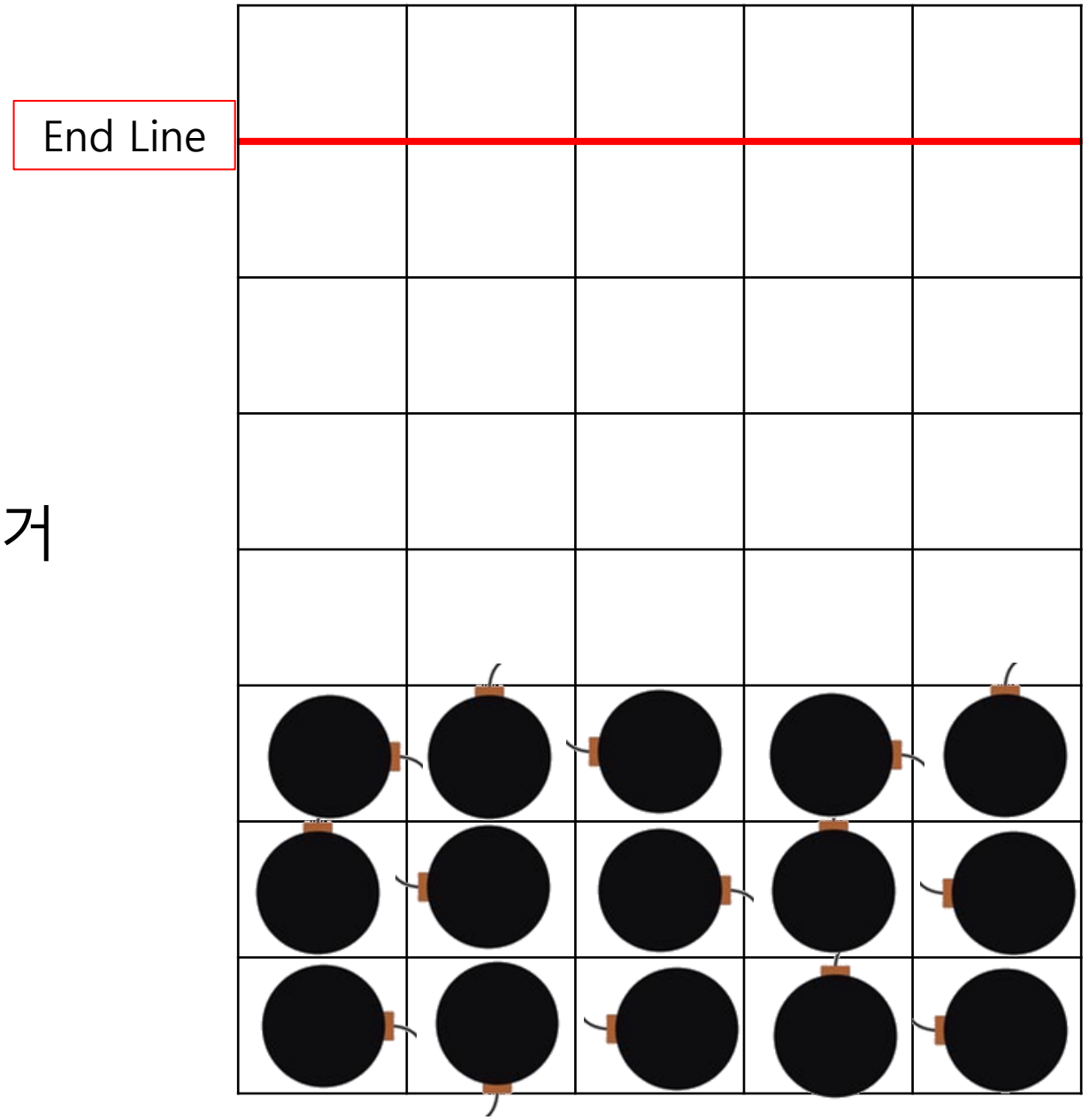
<게임 설명>

1. 폭탄 생성
2. 불 생성
3. 심지방향 조절
4. 폭탄제거
5. 연속적으로 폭탄 제거



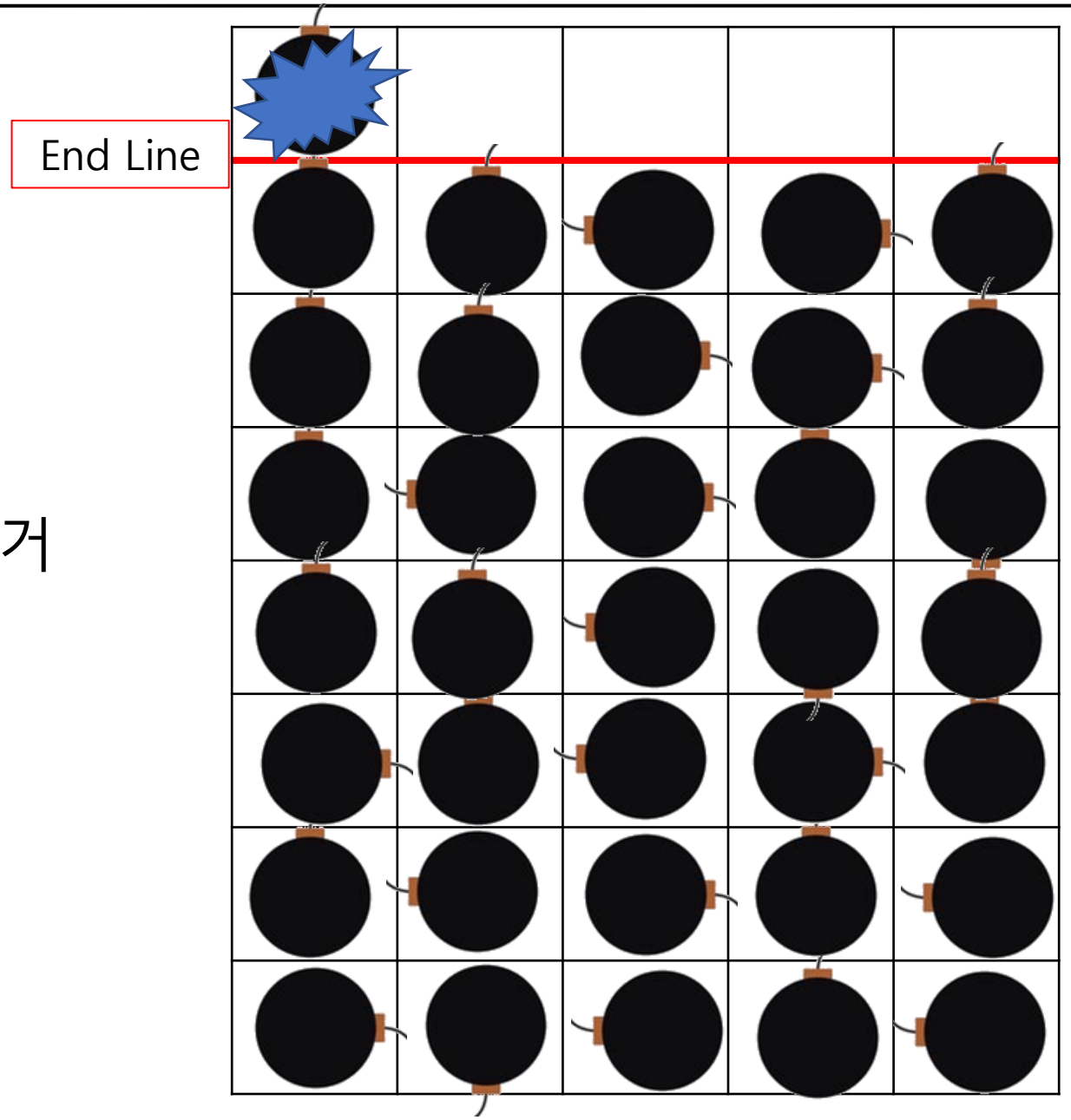
<게임 설명>

1. 폭탄 생성
2. 불 생성
3. 심지방향 조절
4. 폭탄제거
5. 연속적으로 폭탄 제거



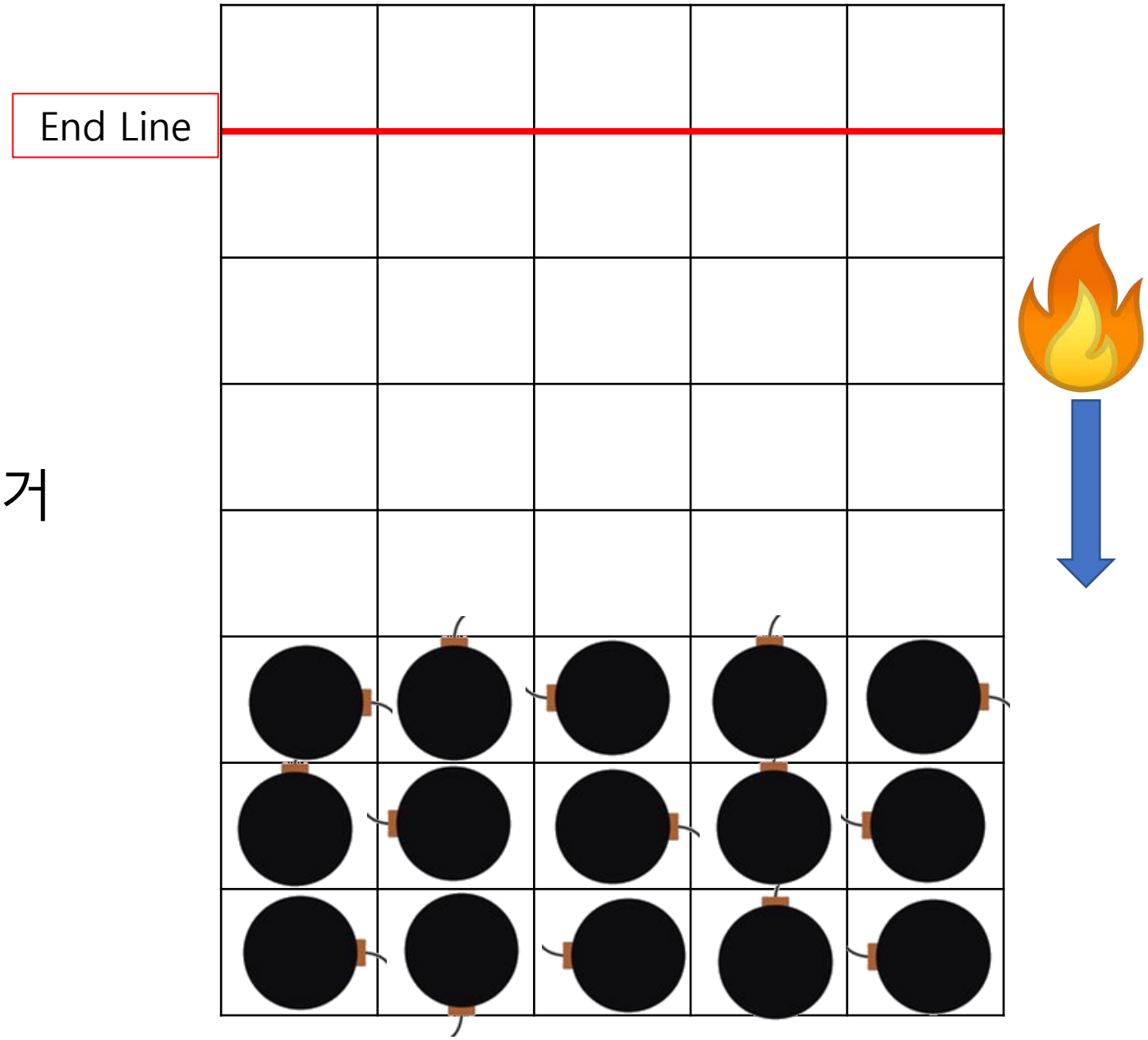
<게임 설명>

1. 폭탄 생성
2. 불 생성
3. 심지방향 조절
4. 폭탄제거
5. 연속적으로 폭탄 제거



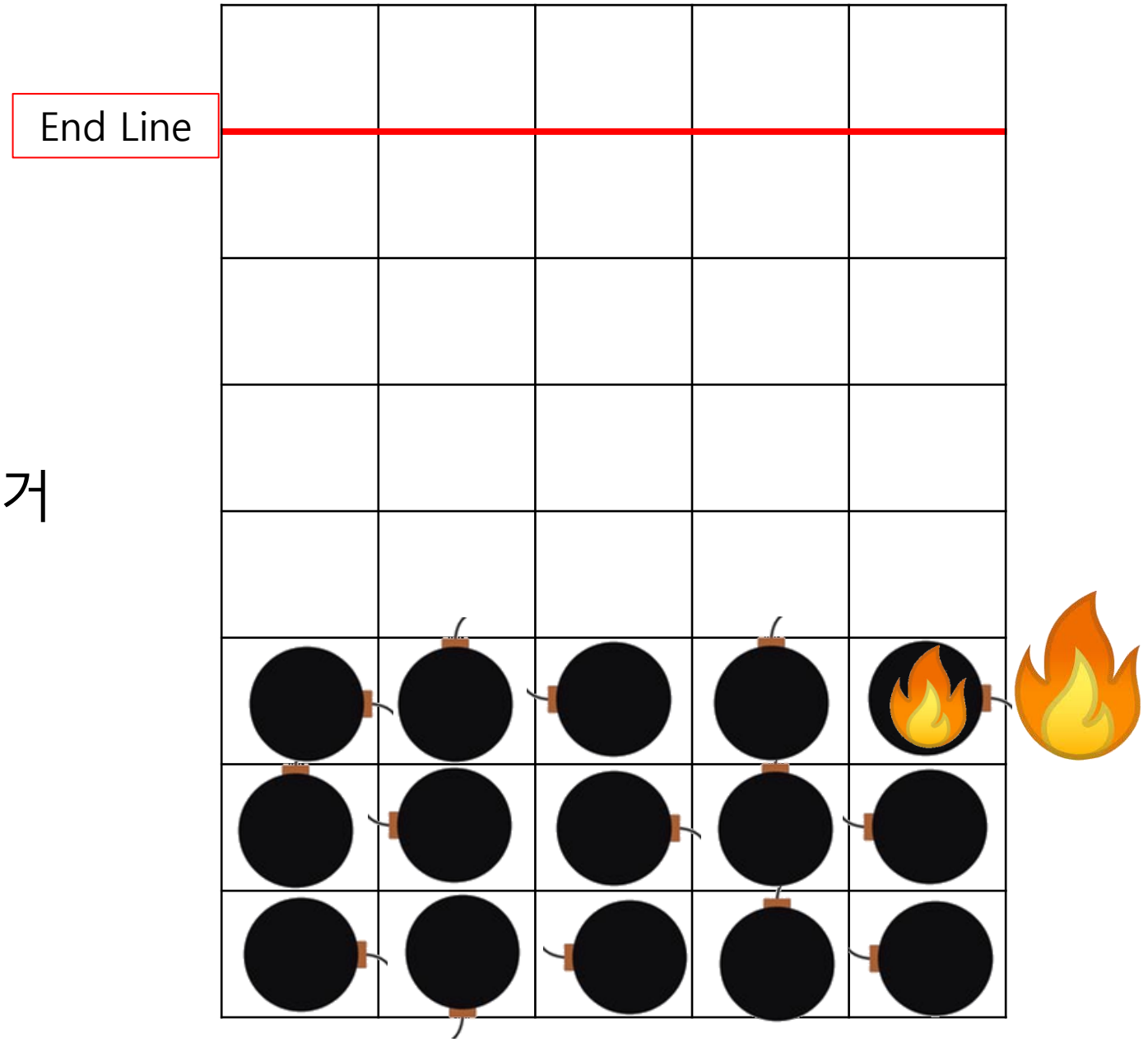
<게임 설명>

1. 폭탄 생성
2. 불 생성
3. 심지방향 조절
4. 폭탄제거
5. 연속적으로 폭탄 제거



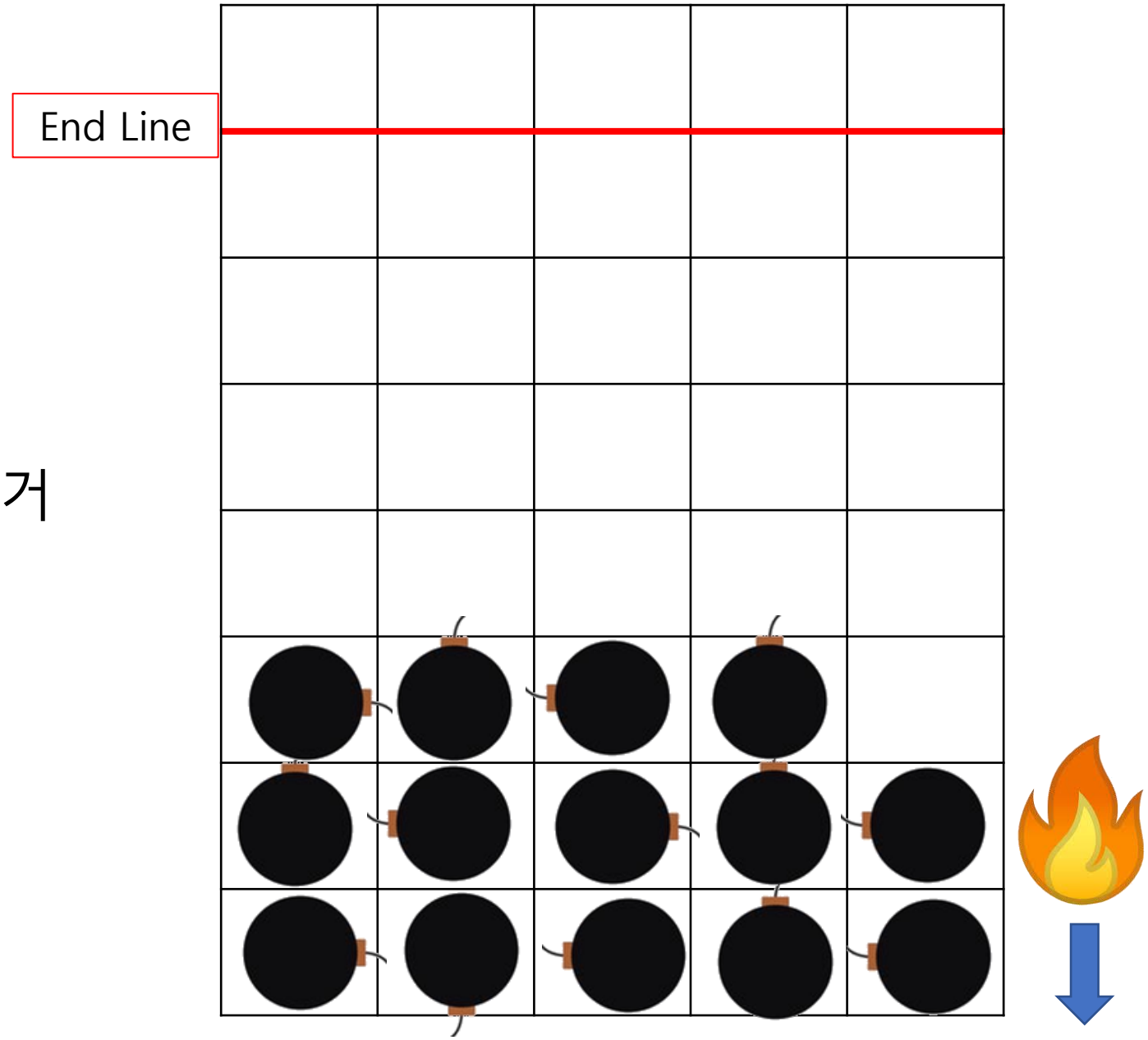
<게임 설명>

1. 폭탄 생성
2. 불 생성
3. 심지방향 조절
4. 폭탄제거
5. 연속적으로 폭탄 제거



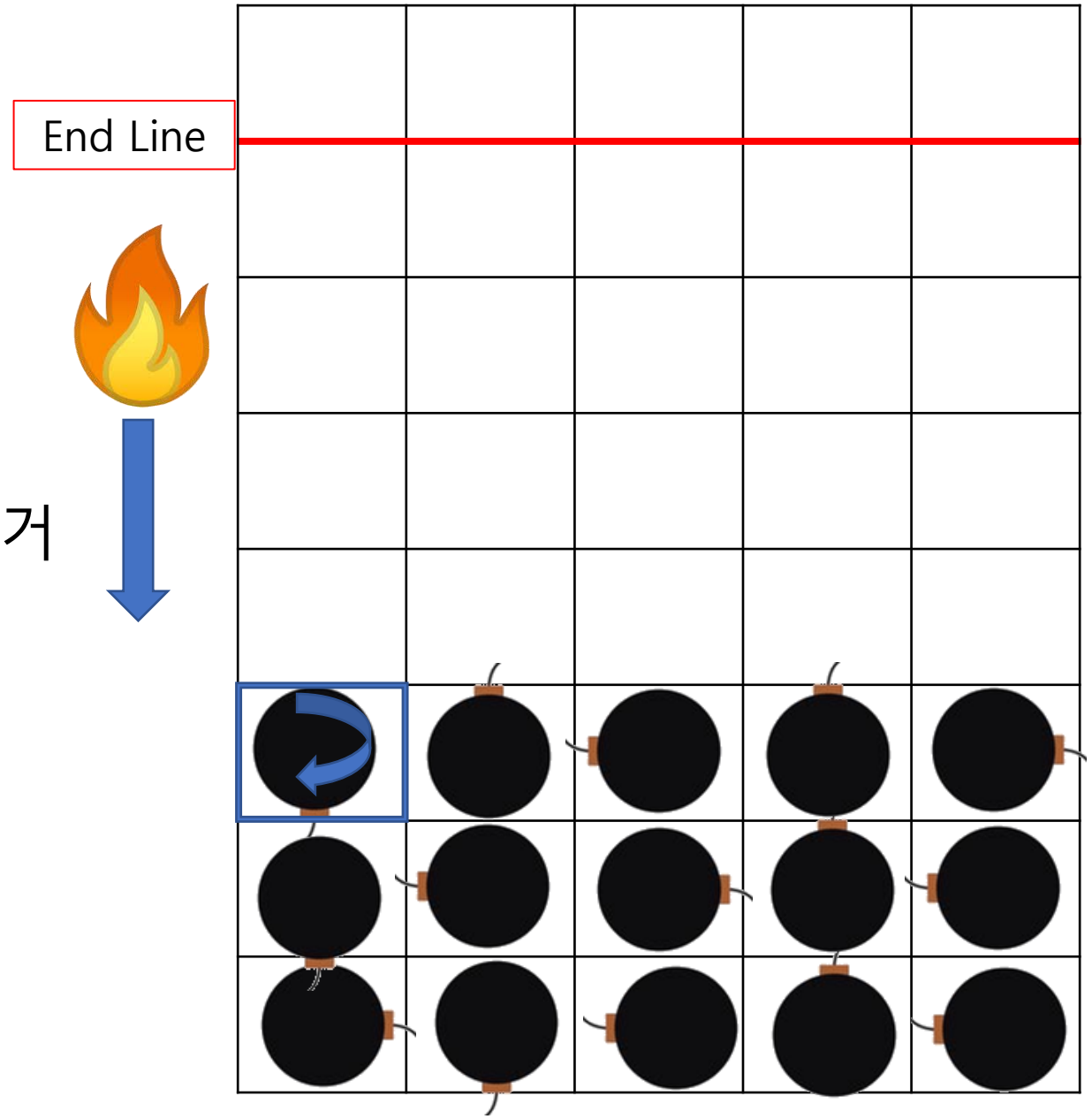
<게임 설명>

1. 폭탄 생성
2. 불 생성
3. 심지방향 조절
4. 폭탄제거
5. 연속적으로 폭탄 제거



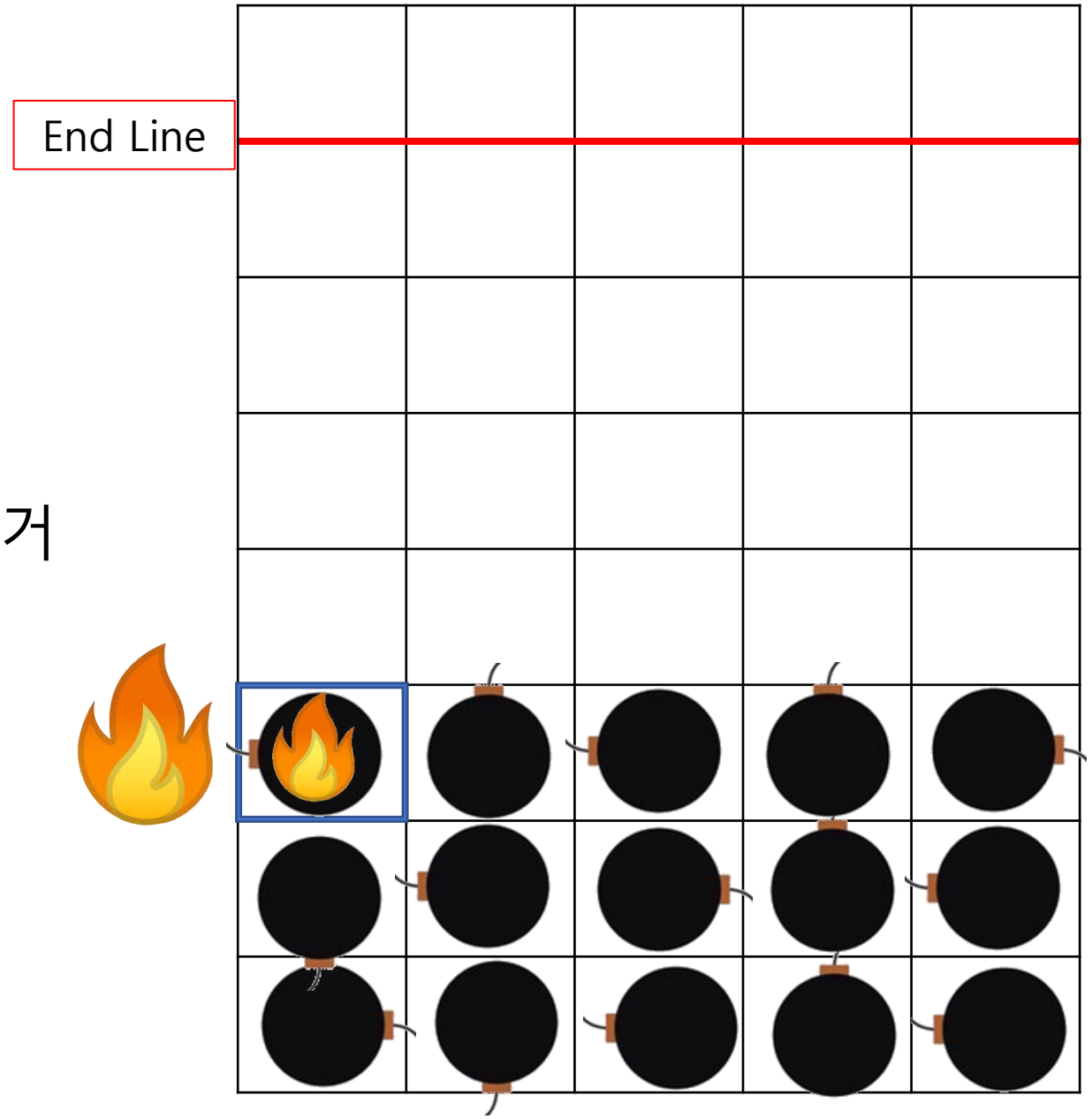
<게임 설명>

1. 폭탄 생성
2. 불 생성
3. 심지방향 조절
4. 폭탄제거
5. 연속적으로 폭탄 제거



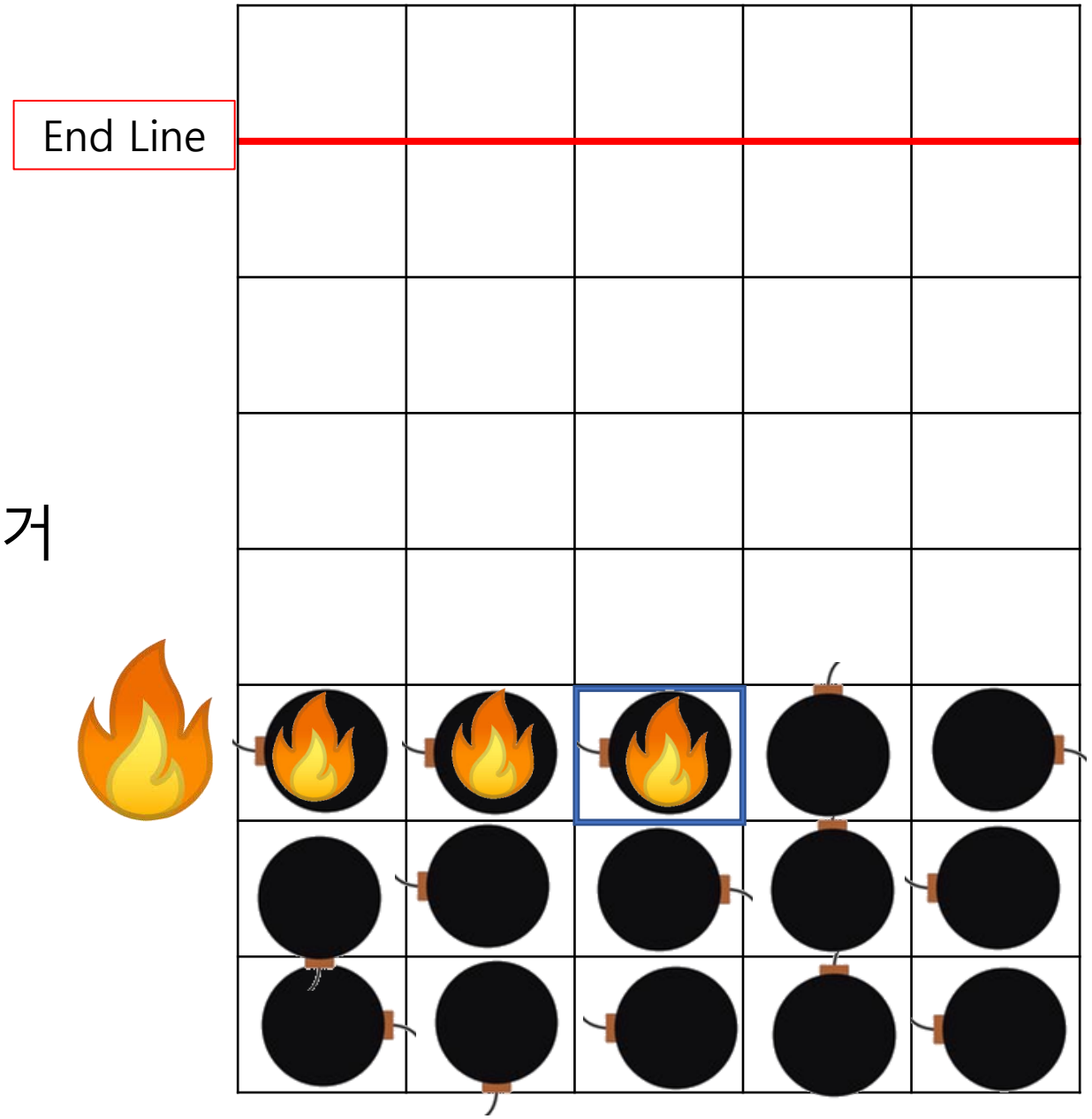
<게임 설명>

1. 폭탄 생성
2. 불 생성
3. 심지방향 조절
4. 폭탄제거
5. 연속적으로 폭탄 제거



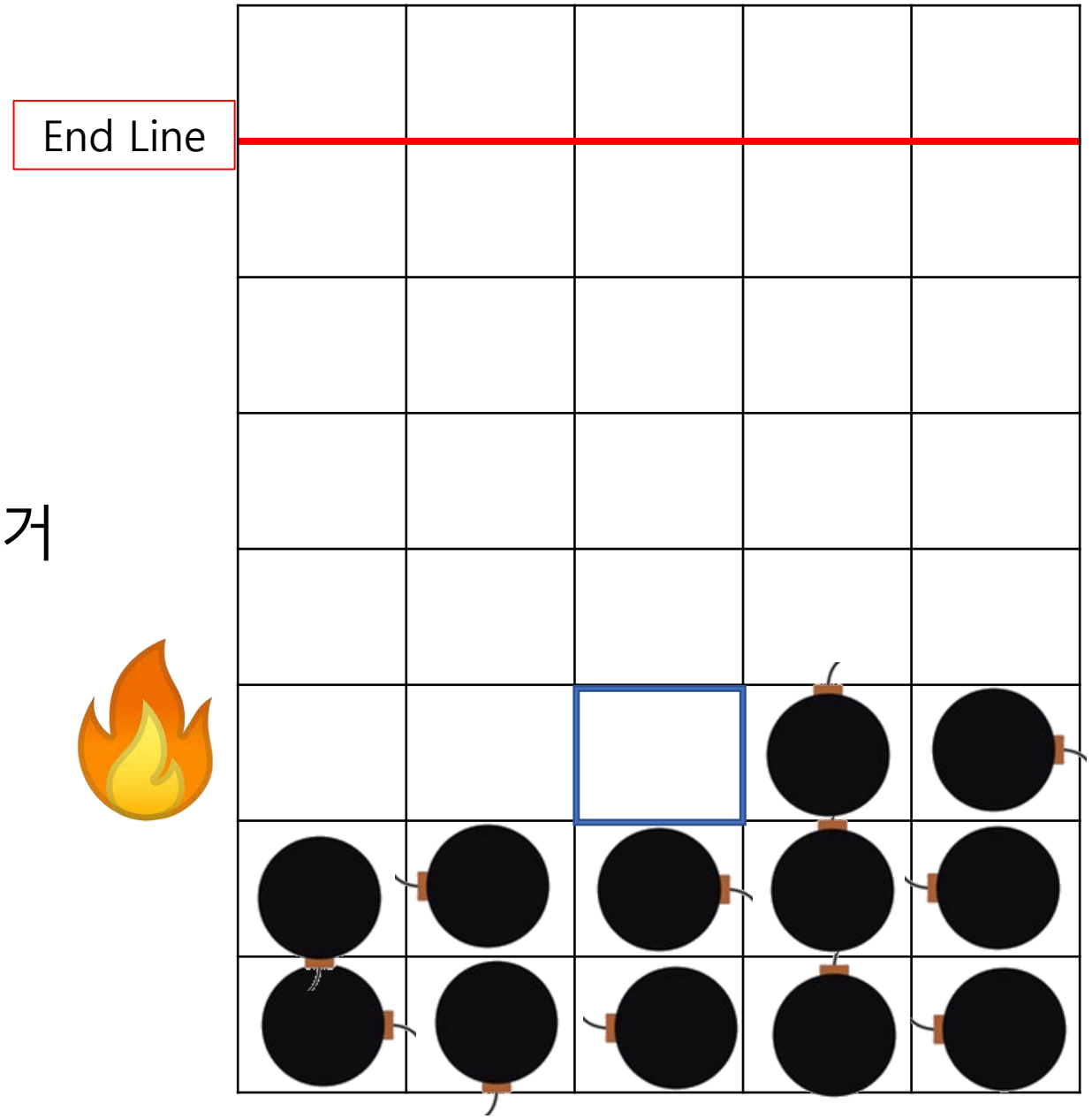
<게임 설명>

1. 폭탄 생성
2. 불 생성
3. 심지방향 조절
4. 폭탄제거
5. 연속적으로 폭탄 제거

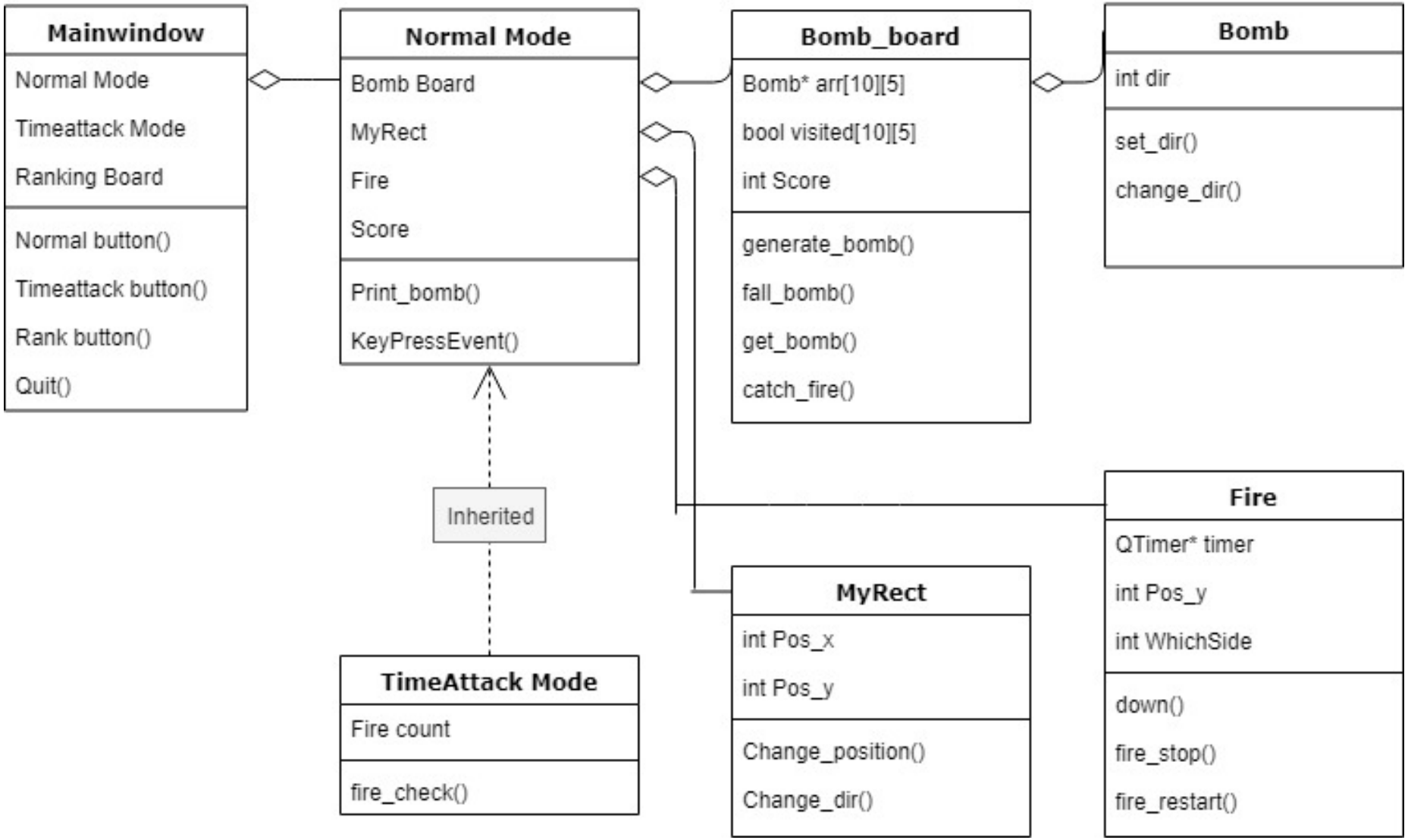


<게임 설명>

1. 폭탄 생성
2. 불 생성
3. 심지방향 조절
4. 폭탄제거
5. 연속적으로 폭탄 제거



3. Class Design



Mode Class : 게임 모드에 대한 Class로,

게임 진행에 필요한 멤버변수와 함수들이 있다.

-멤버 변수

- Gameboard : 게임 판에 관련된 변수(폭탄 배열)
- MyRect : 사용자 입력을 받기 위한 커서
- Fire : 랜덤하게 불을 생성하기 위한 변수
- Score : 사용자의 현재 점수를 저장

- 멤버 함수

- PrintBomb() : Bomb_arr와 fire를 입력받아 화면 출력
- KeyPressEvent(): 사용자의 입력에 따라 이동, 폭탄 방향 회전, 폭탄 생성 등의 기능 수행

```
class Mode: public QGraphicsScene{
    Q_OBJECT
public:
    Mode();
    void printBomb(bomb* arr[][5], fire* fire);
    virtual void keyPressEvent(QKeyEvent * event);
    gameboard* r_board() {return Board;}
    gameboard* Board;
private:
    int init_x,init_y;
    int arr[10][5];
    QTimer* firetimer;
    int count;
    int n;
    int score;
    bomb* bomb_arr [10][5];
    QTimer* curser_timer;
    QGraphicsPixmapItem* image[51];
    QGraphicsRectItem* Rect[5];
    Score * cur_score;

protected:
    fire* Fire;
     MyRect *rect;//커서 표시(사각형)

public slots:
    void restartfire();
    void change_bomb_dir();
    void printprint();
    void oversignal();
    void plus_score(int a);
signals:
     void bomb_reprint();
     void print_again();
```

gameboard Class : 게임 판에 대한 Class로, 폭탄 배열과 폭탄을 생성하고, 터트리는 함수가 있다.

-멤버 변수

- bomb[10][5] : 폭탄을 담아두는 배열
- visited[10][5] : BFS 알고리즘으로 폭탄 터트릴 때 사용
- score : 폭탄을 한번 터트렸을 때 점수

- 멤버 함수

- generate_bomb() : 아랫줄에서 폭탄 5개 생성
- catch_fire() : 심지에 불이 붙으면 폭탄을 제거
- get_bomb() : 폭탄의 주소를 반환
- fall_bomb() : 폭탄 제거 이후 남은 폭탄 정렬

```
class gameboard : public QObject
{
private:
    Q_OBJECT
    QTimer* timer;
    bool visited[10][5];
    bomb* board[10][5];
    int firecount;

public:
    explicit gameboard(QObject *parent = nullptr);
    void fallbomb();
    void change_direction(const int x,const int y);
    bomb* get_bomb(const int x, const int y);
    friend void printBomb();
    bomb* (*get_all_bomb())[5];
    int get_bomb_dir(const int x, const int y);

signals:
    void stop();
    void start();
    void board_change();
    void gameover();
    void score(int sc);

public slots:
    void catchfire(int x, int y);
    void generatebomb();
```

Bomb Class : 폭탄에 관련된 정보 저장

-멤버 변수

•dir : 폭탄의 방향 지정 (위쪽부터 시계방향으로 0~3)

- 멤버 함수

• set_dir() : 폭탄의 방향 지정

• change_dir() : 폭탄의 방향 변경

```
class bomb :public QObject{
    Q_OBJECT
private:
    int dir;
public:
    bomb();
    bomb(int direction);
    virtual ~bomb(){}
    int direction();
    void setdir(){dir=rand()%4;}//방향을 리턴해주는 함수

public slots:
    void changedir(); //시계방향으로 돌리는 함수

};
```

MyRect Class : 폭탄에 관련된 정보 저장

-멤버 변수

• pos_x, pos_y : 사용자의 입력을 받는 커서 위치

- 멤버 함수

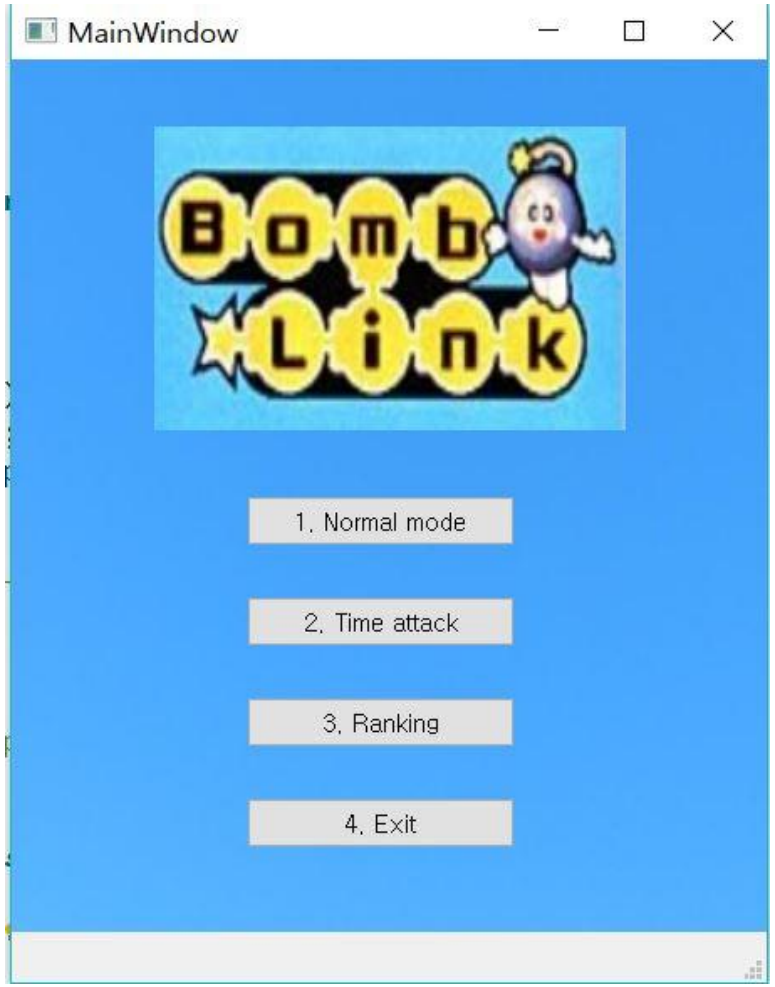
• change_pos() : 폭탄의 방향 지정

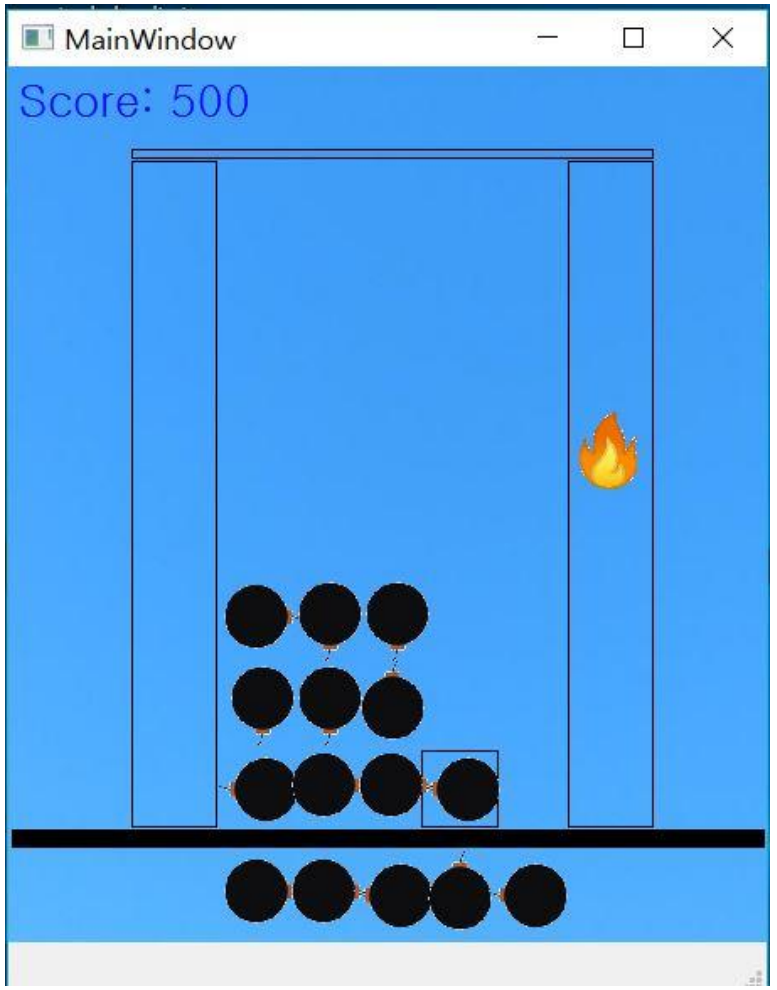
• change_dir() : 폭탄의 방향 변경

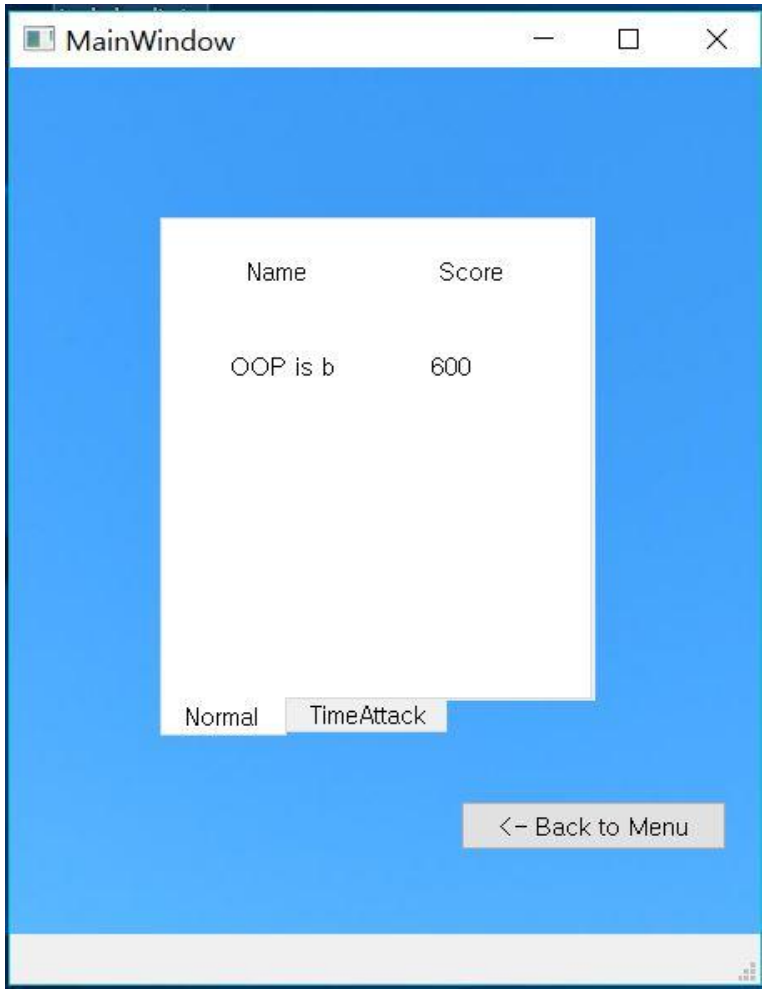
```
class MyRect : public QObject ,public QGraphicsRectItem{
    Q_OBJECT
public:
    MyRect(int ,int );
    void change_position(int x, int y);
    void keyPressEvent(QKeyEvent *event);
    int get_x(){return pos_x;}
    int get_y(){return pos_y;}
private:
    int pos_x,pos_y;

signals:
    void change_dir();
    void up_bomb();
};
```

4. DEMO


MainWindow
Bomb Link
1. Normal mode
2. Time attack
3. Ranking
4. Exit


MainWindow
Score: 500


MainWindow
Name
Score
OOP is b
600
Normal
TimeAttack
<- Back to Menu

5. Improvement direction

1. TimeAttack_mode

일반 Mode class를 상속받아, 필요한 변수, 함수를 추가하면 다양한 모드의 게임을 구현할 수 있을 것이다.

```
class TimeAttack_mode: public Mode{
public:
    TimeAttack_mode();
    virtual void keyPressEvent(QKeyEvent * event);
private:
    Score * num_fire=0;
};
```

2. Fixed_bomb

일반 Bomb class를 상속받아, 방향을 움직일 수 없는 폭탄, 길이가 긴 폭탄 등 여러 폭탄을 구현할 수 있을 것이다.

Thank you