

find game

0.0.1

다음에 의해 생성됨 : Doxygen 1.9.1



<b>1 README</b>	<b>1</b>
<b>2 README</b>	<b>3</b>
<b>3 파일 색인</b>	<b>5</b>
3.1 파일 목록	5
<b>4 파일 문서화</b>	<b>7</b>
4.1 almost2 (origin)/ending.c 파일 참조	7
4.1.1 함수 문서화	7
4.1.1.1 main()	8
4.2 almost2 (origin)/game.c 파일 참조	8
4.2.1 함수 문서화	9
4.2.1.1 checkMatch()	9
4.2.1.2 initializeBoard()	9
4.2.1.3 play_game()	10
4.2.1.4 printBoard()	11
4.2.1.5 processInput()	12
4.2.1.6 shuffleCards()	13
4.2.2 변수 문서화	13
4.2.2.1 board	13
4.2.2.2 images	14
4.2.2.3 revealedCards	14
4.2.2.4 score	14
4.3 almost2 (origin)/game.h 파일 참조	14
4.3.1 매크로 문서화	15
4.3.1.1 BOARD_AREA	15
4.3.1.2 BOARD_SIZE	15
4.3.1.3 IMG_HEIGHT	15
4.3.1.4 IMG_WIDTH	16
4.3.1.5 MAX_IMAGES	16
4.3.2 함수 문서화	16
4.3.2.1 checkMatch()	16
4.3.2.2 initializeBoard()	17
4.3.2.3 play_game()	17
4.3.2.4 printBoard()	18
4.3.2.5 processInput()	19
4.3.2.6 shuffleCards()	20
4.3.3 변수 문서화	20
4.3.3.1 board	21
4.3.3.2 images	21
4.3.3.3 revealedCards	21
4.3.3.4 score	21

4.4 almost2 (origin)/helpme.c 파일 참조	21
4.4.1 매크로 문서화	22
4.4.1.1 BOARD_AREA	22
4.4.1.2 BOARD_SIZE	22
4.4.1.3 IMG_HEIGHT	23
4.4.1.4 IMG_WIDTH	23
4.4.1.5 MAX_IMAGES	23
4.4.2 함수 문서화	23
4.4.2.1 checkMatch()	23
4.4.2.2 disableRawMode()	24
4.4.2.3 display_credits()	24
4.4.2.4 display_images()	25
4.4.2.5 display_menu()	25
4.4.2.6 enableRawMode()	26
4.4.2.7 initializeBoard()	26
4.4.2.8 main()	27
4.4.2.9 play_game()	28
4.4.2.10 printBoard()	29
4.4.2.11 processInput()	30
4.4.2.12 shuffleCards()	31
4.4.3 변수 문서화	31
4.4.3.1 board	31
4.4.3.2 images	32
4.4.3.3 orig_termios	32
4.4.3.4 revealedCards	32
4.4.3.5 score	32
4.5 helpme.c 파일 참조	32
4.5.1 매크로 문서화	33
4.5.1.1 BOARD_AREA	33
4.5.1.2 BOARD_SIZE	33
4.5.1.3 IMG_HEIGHT	34
4.5.1.4 IMG_WIDTH	34
4.5.1.5 MAX_IMAGES	34
4.5.2 함수 문서화	34
4.5.2.1 checkMatch()	34
4.5.2.2 disableRawMode()	35
4.5.2.3 display_credits()	35
4.5.2.4 display_images()	35
4.5.2.5 display_menu()	35
4.5.2.6 enableRawMode()	36
4.5.2.7 initializeBoard()	36
4.5.2.8 main()	37

4.5.2.9 play_game()	37
4.5.2.10 printBoard()	38
4.5.2.11 processInput()	39
4.5.2.12 shuffleCards()	39
4.5.3 변수 문서화	40
4.5.3.1 board	40
4.5.3.2 images	40
4.5.3.3 orig_termios	40
4.5.3.4 revealedCards	40
4.5.3.5 score	40
4.6 almost2 (origin)/main.c 파일 참조	41
4.6.1 함수 문서화	41
4.6.1.1 display_menu()	41
4.6.1.2 main()	42
4.7 almost2 (origin)/utility.c 파일 참조	43
4.7.1 함수 문서화	44
4.7.1.1 disableRawMode()	44
4.7.1.2 display_credits()	44
4.7.1.3 display_images()	45
4.7.1.4 enableRawMode()	45
4.7.2 변수 문서화	45
4.7.2.1 orig_termios	45
4.8 almost2 (origin)/utility.h 파일 참조	46
4.8.1 함수 문서화	46
4.8.1.1 disableRawMode()	46
4.8.1.2 display_credits()	47
4.8.1.3 display_images()	47
4.8.1.4 enableRawMode()	48
4.9 ed.c 파일 참조	48
4.9.1 함수 문서화	49
4.9.1.1 main()	49
4.10 loT3_ver_0.5.c 파일 참조	49
4.10.1 매크로 문서화	50
4.10.1.1 MAX_BOARD	50
4.10.1.2 MAX_IMAGES	50
4.10.2 함수 문서화	50
4.10.2.1 display_credits()	51
4.10.2.2 display_images()	51
4.10.2.3 display_menu()	52
4.10.2.4 initializeBoard()	53
4.10.2.5 main()	54
4.10.2.6 play_game()	54

4.10.2.7 shuffleCards()	55
4.10.3 변수 문서화	55
4.10.3.1 images	55
4.11 README.md 파일 참조	56
4.12 프로젝트 경과/README.md 파일 참조	56
4.13 프로젝트 경과/loT3_ver0.2.c 파일 참조	56
4.13.1 매크로 문서화	57
4.13.1.1 MAX_BOARD	57
4.13.1.2 MAX_IMAGES	57
4.13.2 함수 문서화	57
4.13.2.1 display_credits()	57
4.13.2.2 display_images()	58
4.13.2.3 display_menu()	58
4.13.2.4 initializeBoard()	59
4.13.2.5 main()	60
4.13.2.6 play_game()	61
4.13.2.7 shuffleCards()	62
4.13.3 변수 문서화	62
4.13.3.1 images	63
4.14 프로젝트 경과/loT3_ver_0.1.c 파일 참조	63
4.14.1 매크로 문서화	63
4.14.1.1 MAX_BOARD	64
4.14.1.2 MAX_IMAGES	64
4.14.2 함수 문서화	64
4.14.2.1 display_credits()	64
4.14.2.2 display_images()	65
4.14.2.3 display_menu()	65
4.14.2.4 initializeBoard()	66
4.14.2.5 main()	67
4.14.2.6 play_game()	67
4.14.2.7 shuffleCards()	68
4.14.3 변수 문서화	69
4.14.3.1 images	69
4.15 프로젝트 경과/loT3_ver_0.2.c 파일 참조	69
4.15.1 매크로 문서화	70
4.15.1.1 MAX_BOARD	70
4.15.1.2 MAX_IMAGES	70
4.15.2 함수 문서화	70
4.15.2.1 display_credits()	70
4.15.2.2 display_images()	71
4.15.2.3 display_menu()	71
4.15.2.4 initializeBoard()	72

4.15.2.5 main()	73
4.15.2.6 play_game()	74
4.15.2.7 shuffleCards()	75
4.15.3 변수 문서화	75
4.15.3.1 images	76
4.16 프로젝트 경과/loT3_ver_0.3.c 파일 참조	76
4.16.1 매크로 문서화	76
4.16.1.1 MAX_BOARD	77
4.16.1.2 MAX_IMAGES	77
4.16.2 함수 문서화	77
4.16.2.1 display_credits()	77
4.16.2.2 display_images()	78
4.16.2.3 display_menu()	78
4.16.2.4 initializeBoard()	80
4.16.2.5 main()	81
4.16.2.6 play_game()	82
4.16.2.7 shuffleCards()	82
4.16.3 변수 문서화	83
4.16.3.1 images	83
4.17 프로젝트 경과/loT3_ver_0.4.c 파일 참조	83
4.17.1 매크로 문서화	84
4.17.1.1 MAX_BOARD	84
4.17.1.2 MAX_IMAGES	84
4.17.2 함수 문서화	84
4.17.2.1 display_credits()	85
4.17.2.2 display_images()	85
4.17.2.3 display_menu()	86
4.17.2.4 initializeBoard()	87
4.17.2.5 main()	88
4.17.2.6 play_game()	89
4.17.2.7 shuffleCards()	90
4.17.3 변수 문서화	90
4.17.3.1 images	91
4.18 프로젝트 경과/살려조.c 파일 참조	91
4.18.1 매크로 문서화	92
4.18.1.1 MAX_BOARD	92
4.18.1.2 MAX_IMAGES	92
4.18.2 함수 문서화	92
4.18.2.1 display_credits()	93
4.18.2.2 display_images()	93
4.18.2.3 display_menu()	94
4.18.2.4 initializeBoard()	94

4.18.2.5 main()	95
4.18.2.6 play_game()	96
4.18.2.7 shuffleCards()	97
4.18.3 변수 문서화	97
4.18.3.1 images	97
4.19 프로젝트 경과/살려줘.c 파일 참조	97
4.19.1 매크로 문서화	98
4.19.1.1 MAX_BOARD	98
4.19.1.2 MAX_IMAGES	99
4.19.2 함수 문서화	99
4.19.2.1 display_credits()	99
4.19.2.2 display_images()	99
4.19.2.3 display_menu()	100
4.19.2.4 initializeBoard()	100
4.19.2.5 main()	101
4.19.2.6 play_game()	102
4.19.2.7 shuffleCards()	103
4.19.3 변수 문서화	103
4.19.3.1 images	103

<b>Index</b>	<b>105</b>
--------------	------------



# Chapter 1

## README

- 3조 프로젝트 계획서
  - 같은 그림 찾기 게임
    - \* 1라운드 2 x 2 >>> 총 4장의 카드 2쌍의 이미지
    - \* 2라운드 4 x 4 >>> 총 16장의 카드 8쌍의 이미지
    - \* 3라운드 6 x 6 >>> 총 36장의 카드 18쌍의 이미지
  - 매 라운드 무작위로 카드가 배치되며 5초간 보여주고 카드를 뒤집음
  - 방향키로 카드를 이동하며 이동시 선택되어있는 카드는 커서를 깜빡임
  - 스페이스바로 2장의 카드를 연속 선택시 앞면을 보이며 점수 카운트 1상승, 선택되지 않은 카드들은 계속 뒤집혀서 선택을 기다림
  - 서로 다른 두 장의 카드를 연속 선택시 점수 카운트 -1
  - 1라운드는 점수 카운트가 2가 되면 다음 라운드로 진출
  - 2라운드는 점수 카운트가 8이 되면 다음 라운드로 진출
  - 3라운드는 점수 카운트가 18이 되면 클리어
  - 점수 카운트가 0보다 작아지면 게임 실패
  - 게임 실패시 격려 문자 출력 후 메인 화면으로 이동
  - 각 라운드를 클리어해야 다음 라운드가 진행되며 3라운드 클리어시 축하 문자가 출력 후 메인 화면으로 이동
  - 메인 메뉴
- 1. 게임 시작
- 2. 사진 체크(게임에 나오는 이미지들 출력)
- 3. 크레딧 출력(위에서 아래로)후 메인 화면으로 이동
- 1. 게임 종료



## Chapter 2

# README

- 카드를 랜덤하게 섞는 알고리즘
  - 배열 요소를 무작위로 교환
    - \* Fisher-Yates 섞기 알고리즘
    - \* 2차원 배열을 랜덤하게 섞음
  - shuffle 함수 배열을 무작위로 섞는 역할

```
#include <time.h>
srand(time(NULL))
- 난수 생성기 초기화
```

```
#include <string.h>
strcpy를 사용하기 위한 헤더파일 인클루드
```



## Chapter 3

# 파일 색인

### 3.1 파일 목록

다음은 모든 파일에 대한 목록입니다. (간략한 설명만을 보여줍니다) :

<a href="#">ed.c</a>	48
<a href="#">helpme.c</a>	32
<a href="#">IoT3_ver_0.5.c</a>	49
almost2 (origin)/ <a href="#">ending.c</a>	7
almost2 (origin)/ <a href="#">game.c</a>	8
almost2 (origin)/ <a href="#">game.h</a>	14
almost2 (origin)/ <a href="#">helpme.c</a>	21
almost2 (origin)/ <a href="#">main.c</a>	41
almost2 (origin)/ <a href="#">utility.c</a>	43
almost2 (origin)/ <a href="#">utility.h</a>	46
프로젝트 경과/ <a href="#">IoT3_ver0.2.c</a>	56
프로젝트 경과/ <a href="#">IoT3_ver_0.1.c</a>	63
프로젝트 경과/ <a href="#">IoT3_ver_0.2.c</a>	69
프로젝트 경과/ <a href="#">IoT3_ver_0.3.c</a>	76
프로젝트 경과/ <a href="#">IoT3_ver_0.4.c</a>	83
프로젝트 경과/ <a href="#">살려조.c</a>	91
프로젝트 경과/ <a href="#">살려줘.c</a>	97



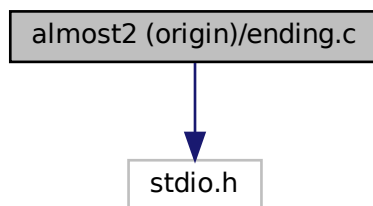
## Chapter 4

# 파일 문서화

### 4.1 almost2 (origin)/ending.c 파일 참조

```
#include <stdio.h>
```

ending.c에 대한 include 의존 그래프



함수

- int `main` ()

#### 4.1.1 함수 문서화





## 4.2.1 함수 문서화

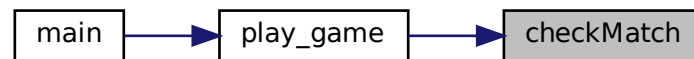
### 4.2.1.1 checkMatch()

```
void checkMatch (
    int * selectedCards,
    int * selectedCount )
```

game.c 파일의 183 번째 라인에서 정의되었습니다.

```
183                                     {
184     if (*selectedCount == 2) {
185         if (board[selectedCards[0]] == board[selectedCards[1]]) {
186             score++;
187             printf("found!");
188         } else {
189             revealedCards[selectedCards[0]] = 0;
190             revealedCards[selectedCards[1]] = 0;
191             printf("No match. Try again.\n");
192         }
193         *selectedCount = 0;
194         sleep(1);
195         // 사용자가 결과를 볼 수 있도록 잠시 대기
196     }
197 }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



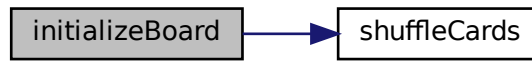
### 4.2.1.2 initializeBoard()

```
void initializeBoard ( )
```

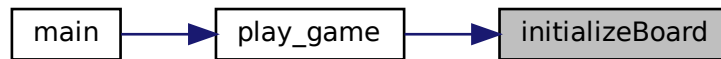
game.c 파일의 98 번째 라인에서 정의되었습니다.

```
98                                     {
99     int placements[BOARD_AREA] = {0};
100
101     for (int i = 0; i < BOARD_AREA / 2; i++) {
102         placements[i] = i % (MAX_IMAGES / 2);
103         placements[i + BOARD_AREA / 2] = i % (MAX_IMAGES / 2);
104     }
105
106     shuffleCards(placements, BOARD_AREA);
107
108     for (int i = 0; i < BOARD_AREA; i++) {
109         board[i] = placements[i];
110     }
111 }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.2.1.3 play\_game()

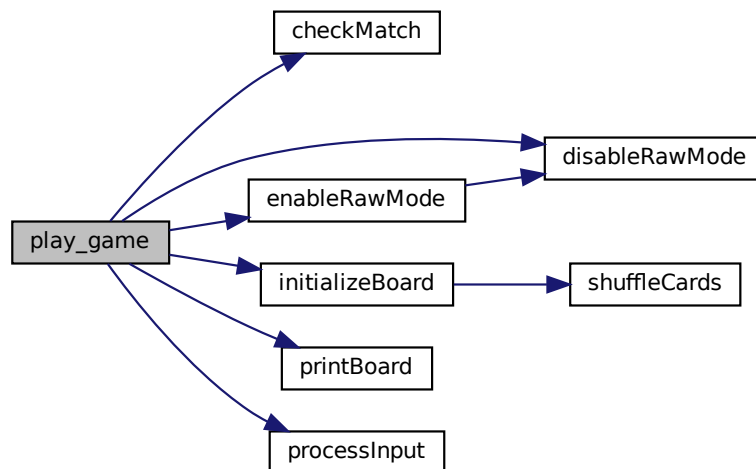
```
void play_game ( )
```

game.c 파일의 199 번째 라인에서 정의되었습니다.

```

199 {
200     int cursorPosition = 0; // 현재 커서 위치
201     int selectedCards[2]; // 선택된 카드의 인덱스
202     int selectedCount = 0; // 현재 선택된 카드의 수
203
204     enableRawMode();
205
206     initializeBoard(); // 게임 보드 초기화
207     while (score < (BOARD_AREA / 2)) { // 모든 쌍을 찾을 때까지 반복
208         system("clear");
209         printBoard(cursorPosition); // 현재 게임 보드 출력, 커서 위치 인자 전달
210         processInput(&cursorPosition, selectedCards, &selectedCount); // 사용자 입력 처리
211         if (selectedCount == 2) { // 두 카드가 모두 선택되면
212             checkMatch(selectedCards, &selectedCount); // 카드가 일치하는지 확인
213         }
214     }
215
216     disableRawMode();
217     printf("Congratulations! You've completed the game with a score of %d.\n", score);
218 }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.2.1.4 printBoard()

```
void printBoard (
    int cursorPosition )
```

game.c 파일의 114 번째 라인에서 정의되었습니다.

```

114 {
115     printf("Match Score: %d\n", score);
116     for (int i = 0; i < BOARD_SIZE; ++i) {
117         for (int j = 0; j < BOARD_SIZE; ++j) {
118             int cardIndex = i * BOARD_SIZE + j;
119             // 카드 상단 경계 출력
120             printf("-----");
121         }
122         printf("\n"); // 줄의 끝에서만 경계 닫기
123
124         for (int k = 0; k < IMG_HEIGHT; ++k) { // 이미지 높이 (7) 만큼 반복
125             for (int j = 0; j < BOARD_SIZE; ++j) {
126                 int cardIndex = i * BOARD_SIZE + j;
127                 if (revealedCards[cardIndex]) { // 카드가 공개된 경우
128                     // 실제 이미지 데이터가 있다면 여기서 이미지 라인을 출력합니다.
129                     // 예제에서는 간단한 텍스트로 대체합니다.

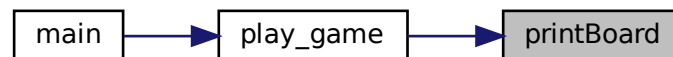
```

```

130         for (int m = 0; m < IMG_WIDTH+1; m++){
131             printf("%c", images[board[cardIndex]][k][m]);
132         }
133     } else if (cardIndex == cursorPosition && k == 0) { // 커서 위치의 첫 번째 라인에만 선택 표시
134         printf("|      선택됨      ");
135     } else {
136         printf("|      "); // 공개되지 않은 카드
137     }
138     // } else{
139     //     for (int m = 0; m < IMG_WIDTH+1; m++){
140     //         printf("%c", images[board[cardIndex]][k][m]);
141     //     }
142     // }
143     printf("\n");
144 }
145
146 for (int j = 0; j < BOARD_SIZE; ++j) {
147     // 카드 하단 경계 출력
148     printf("-----");
149 }
150 printf("\n");
151 }
152 }
153 }
154 }
155 }

```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.2.1.5 processInput()

```

void processInput (
    int * cursorPosition,
    int * selectedCards,
    int * selectedCount )

```

game.c 파일의 158 번째 라인에서 정의되었습니다.

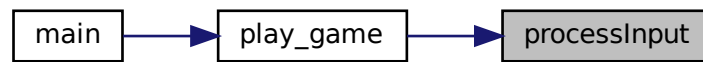
```

158 char c = getchar();
159 switch (c) {
160     case 'w':
161         if (*cursorPosition >= BOARD_SIZE) *cursorPosition -= BOARD_SIZE;
162         break;
163     case 's':
164         if (*cursorPosition < BOARD_AREA - BOARD_SIZE) *cursorPosition += BOARD_SIZE;
165         break;
166     case 'a':
167         if (*cursorPosition % BOARD_SIZE != 0) (*cursorPosition)--;
168         break;
169     case 'd':
170         if (*cursorPosition % BOARD_SIZE != BOARD_SIZE - 1) (*cursorPosition)++;
171         break;
172     case ' ':
173         if (*selectedCount < 2 && !revealedCards[*cursorPosition]) {
174             selectedCards[*selectedCount++] = *cursorPosition;
175             revealedCards[*cursorPosition] = 1;
176         }
177         break;
178 }
179 }

```

```
180 }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.2.1.6 shuffleCards()

```
void shuffleCards (
    int * array,
    int size )
```

game.c 파일의 88 번째 라인에서 정의되었습니다.

```
88 {
89     for (int i = 0; i < size; i++) {
90         int j = rand() % size;
91         int temp = array[i];
92         array[i] = array[j];
93         array[j] = temp;
94     }
95 }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



## 4.2.2 변수 문서화

### 4.2.2.1 board

```
int board[BOARD_AREA] = {0}
```

game.c 파일의 7 번째 라인에서 정의되었습니다.

#### 4.2.2.2 images

```
char images[MAX_IMAGES][IMG_HEIGHT][IMG_WIDTH+1]
```

game.c 파일의 12 번째 라인에서 정의되었습니다.

#### 4.2.2.3 revealedCards

```
int revealedCards[BOARD_AREA] = {0}
```

game.c 파일의 8 번째 라인에서 정의되었습니다.

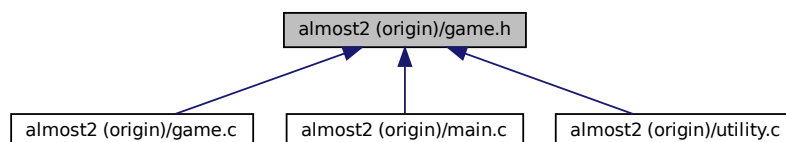
#### 4.2.2.4 score

```
int score = 0
```

game.c 파일의 9 번째 라인에서 정의되었습니다.

### 4.3 almost2 (origin)/game.h 파일 참조

이 그래프는 이 파일을 직/간접적으로 include 하는 파일들을 보여줍니다.:



#### 매크로

- #define IMG\_HEIGHT 7
- #define IMG\_WIDTH 18
- #define MAX\_IMAGES 16
- #define BOARD\_SIZE 4
- #define BOARD\_AREA (BOARD\_SIZE \* BOARD\_SIZE)

## 함수

- void `initializeBoard` ()
- void `shuffleCards` (int \*array, int size)
- void `printBoard` (int cursorPosition)
- void `processInput` (int \*cursorPosition, int \*selectedCards, int \*selectedCount)
- void `checkMatch` (int \*selectedCards, int \*selectedCount)
- void `play_game` ()

## 변수

- char `images` [MAX\_IMAGES][IMG\_HEIGHT][IMG\_WIDTH+1]
- int `board` [BOARD\_AREA]
- int `revealedCards` [BOARD\_AREA]
- int `score`

### 4.3.1 매크로 문서화

#### 4.3.1.1 BOARD\_AREA

```
#define BOARD_AREA (BOARD_SIZE * BOARD_SIZE)
```

game.h 파일의 8 번째 라인에서 정의되었습니다.

#### 4.3.1.2 BOARD\_SIZE

```
#define BOARD_SIZE 4
```

game.h 파일의 7 번째 라인에서 정의되었습니다.

#### 4.3.1.3 IMG\_HEIGHT

```
#define IMG_HEIGHT 7
```

game.h 파일의 4 번째 라인에서 정의되었습니다.

#### 4.3.1.4 IMG\_WIDTH

```
#define IMG_WIDTH 18
```

game.h 파일의 5 번째 라인에서 정의되었습니다.

#### 4.3.1.5 MAX\_IMAGES

```
#define MAX_IMAGES 16
```

game.h 파일의 6 번째 라인에서 정의되었습니다.

### 4.3.2 함수 문서화

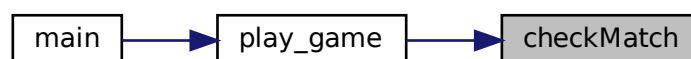
#### 4.3.2.1 checkMatch()

```
void checkMatch (
    int * selectedCards,
    int * selectedCount )
```

game.c 파일의 183 번째 라인에서 정의되었습니다.

```
183                                     {
184     if (*selectedCount == 2) {
185         if (board[selectedCards[0]] == board[selectedCards[1]]) {
186             score++;
187             printf("found!");
188         } else {
189             revealedCards[selectedCards[0]] = 0;
190             revealedCards[selectedCards[1]] = 0;
191             printf("No match. Try again.\n");
192         }
193         *selectedCount = 0;
194         sleep(1);
195         // 사용자가 결과를 볼 수 있도록 잠시 대기
196     }
197 }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:





## 4.3.2.2 initializeBoard()

```
void initializeBoard ( )
```

game.c 파일의 98 번째 라인에서 정의되었습니다.

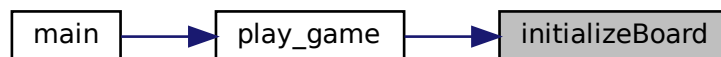
```

98     {
99     int placements[BOARD_AREA] = {0};
100
101     for (int i = 0; i < BOARD_AREA / 2; i++) {
102         placements[i] = i % (MAX_IMAGES / 2);
103         placements[i + BOARD_AREA / 2] = i % (MAX_IMAGES / 2);
104     }
105
106     shuffleCards(placements, BOARD_AREA);
107
108     for (int i = 0; i < BOARD_AREA; i++) {
109         board[i] = placements[i];
110     }
111 }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



## 4.3.2.3 play\_game()

```
void play_game ( )
```

game.c 파일의 199 번째 라인에서 정의되었습니다.

```

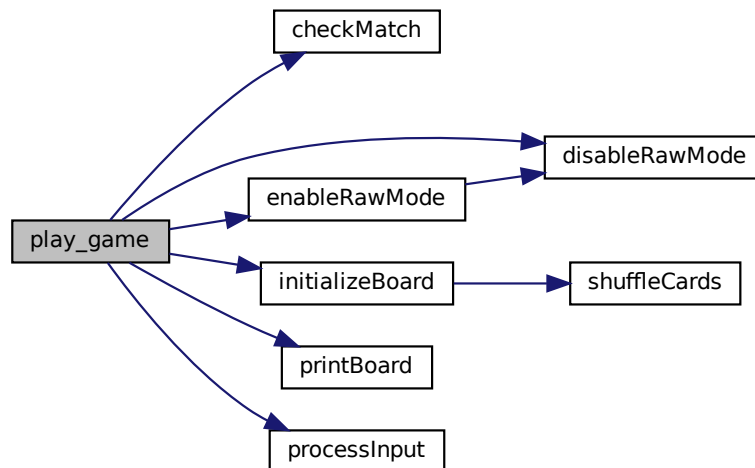
199     {
200     int cursorPosition = 0; // 현재 커서 위치
201     int selectedCards[2]; // 선택된 카드의 인덱스
202     int selectedCount = 0; // 현재 선택된 카드의 수
203
204     enableRawMode();
205
206     initializeBoard(); // 게임 보드 초기화
207     while (score < (BOARD_AREA / 2)) { // 모든 쌍을 찾을 때까지 반복
208         system("clear");
209         printBoard(cursorPosition); // 현재 게임 보드 출력, 커서 위치 인자 전달
  
```

```

210     processInput(&cursorPosition, selectedCards, &selectedCount); // 사용자 입력 처리
211     if (selectedCount == 2) { // 두 카드가 모두 선택되면
212         checkMatch(selectedCards, &selectedCount); // 카드가 일치하는지 확인
213     }
214 }
215
216 disableRawMode();
217 printf("Congratulations! You've completed the game with a score of %d.\n", score);
218 }

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.3.2.4 printBoard()

```

void printBoard (
    int cursorPosition )

```

game.c 파일의 114 번째 라인에서 정의되었습니다.

```

114     {
115     printf("Match Score: %d\n", score);
116     for (int i = 0; i < BOARD_SIZE; ++i) {
117         for (int j = 0; j < BOARD_SIZE; ++j) {
118             int cardIndex = i * BOARD_SIZE + j;

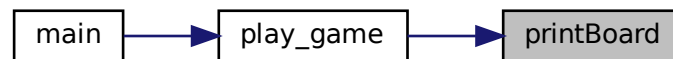
```

```

119         // 카드 상단 경계 출력
120         printf("-----");
121     }
122     printf("+\n"); // 줄의 끝에서만 경계 닫기
123
124     for (int k = 0; k < IMG_HEIGHT; ++k) { // 이미지 높이(7)만큼 반복
125         for (int j = 0; j < BOARD_SIZE; ++j) {
126             int cardIndex = i * BOARD_SIZE + j;
127             if (revealedCards[cardIndex]) { // 카드가 공개된 경우
128                 // 실제 이미지 데이터가 있다면 여기서 이미지 라인을 출력합니다.
129                 // 예제에서는 간단한 텍스트로 대체합니다.
130                 for (int m = 0; m < IMG_WIDTH+1; m++){
131                     printf("%c", images[board[cardIndex]][k][m]);
132                 }
133             } else if (cardIndex == cursorPosition && k == 0) { // 커서 위치의 첫 번째 라인에만 선택 표시
134                 printf("|   선택됨   ");
135             } else {
136                 printf("|               "); // 공개되지 않은 카드
137             }
138             // } else{
139             //     for (int m = 0; m < IMG_WIDTH+1; m++){
140             //         printf("%c", images[board[cardIndex]][k][m]);
141             //     }
142             // }
143         }
144         printf("\n");
145     }
146     printf("\n");
147 }
148
149     for (int j = 0; j < BOARD_SIZE; ++j) {
150         // 카드 하단 경계 출력
151         printf("-----");
152     }
153     printf("+\n");
154 }
155 }

```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.3.2.5 processInput()

```

void processInput (
    int * cursorPosition,
    int * selectedCards,
    int * selectedCount )

```

game.c 파일의 158 번째 라인에서 정의되었습니다.

```

158 {
159     char c = getchar();
160     switch (c) {
161         case 'w':
162             if (*cursorPosition >= BOARD_SIZE) *cursorPosition -= BOARD_SIZE;
163             break;
164         case 's':
165             if (*cursorPosition < BOARD_AREA - BOARD_SIZE) *cursorPosition += BOARD_SIZE;
166             break;
167         case 'a':
168             if (*cursorPosition % BOARD_SIZE != 0) (*cursorPosition)--;

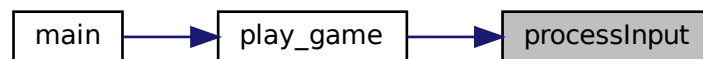
```

```

169         break;
170     case 'd':
171         if (*cursorPosition % BOARD_SIZE != BOARD_SIZE - 1) (*cursorPosition)++;
172         break;
173     case ' ':
174         if (*selectedCount < 2 && !revealedCards[*cursorPosition]) {
175             selectedCards[(*selectedCount)++] = *cursorPosition;
176             revealedCards[*cursorPosition] = 1;
177         }
178         break;
179     }
180 }

```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.3.2.6 shuffleCards()

```

void shuffleCards (
    int * array,
    int size )

```

game.c 파일의 88 번째 라인에서 정의되었습니다.

```

88     {
89         for (int i = 0; i < size; i++) {
90             int j = rand() % size;
91             int temp = array[i];
92             array[i] = array[j];
93             array[j] = temp;
94         }
95     }

```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



### 4.3.3 변수 문서화

#### 4.3.3.1 board

```
int board[BOARD_AREA] [extern]
```

game.c 파일의 7 번째 라인에서 정의되었습니다.

#### 4.3.3.2 images

```
char images[MAX_IMAGES][IMG_HEIGHT][IMG_WIDTH+1] [extern]
```

game.c 파일의 12 번째 라인에서 정의되었습니다.

#### 4.3.3.3 revealedCards

```
int revealedCards[BOARD_AREA] [extern]
```

game.c 파일의 8 번째 라인에서 정의되었습니다.

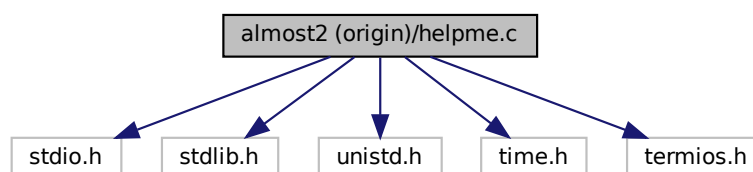
#### 4.3.3.4 score

```
int score [extern]
```

game.c 파일의 9 번째 라인에서 정의되었습니다.

## 4.4 almost2 (origin)/helpme.c 파일 참조

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <time.h>
#include <termios.h>
helpme.c에 대한 include 의존 그래프
```



## 매크로

- `#define IMG_HEIGHT 7`
- `#define IMG_WIDTH 17`
- `#define MAX_IMAGES 16`
- `#define BOARD_SIZE 4`
- `#define BOARD_AREA (BOARD_SIZE * BOARD_SIZE)`

## 함수

- `void disableRawMode ()`
- `void enableRawMode ()`
- `void shuffleCards (int *array, int size)`
- `void initializeBoard ()`
- `void printBoard (int cursorPosition)`
- `void processInput (int *cursorPosition, int *selectedCards, int *selectedCount)`
- `void checkMatch (int *selectedCards, int *selectedCount)`
- `int display_menu ()`
- `void display_credits ()`
- `void display_images ()`
- `void play_game ()`
- `int main ()`

## 변수

- `int board [BOARD_AREA] = {0}`
- `int revealedCards [BOARD_AREA] = {0}`
- `int score = 0`
- `struct termios orig_termios`
- `char images [MAX_IMAGES][IMG_HEIGHT][IMG_WIDTH+1]`

### 4.4.1 매크로 문서화

#### 4.4.1.1 BOARD\_AREA

```
#define BOARD_AREA (BOARD_SIZE * BOARD_SIZE)
```

helpme.c 파일의 12 번째 라인에서 정의되었습니다.

#### 4.4.1.2 BOARD\_SIZE

```
#define BOARD_SIZE 4
```

helpme.c 파일의 11 번째 라인에서 정의되었습니다.

#### 4.4.1.3 IMG\_HEIGHT

```
#define IMG_HEIGHT 7
```

helpme.c 파일의 7 번째 라인에서 정의되었습니다.

#### 4.4.1.4 IMG\_WIDTH

```
#define IMG_WIDTH 17
```

helpme.c 파일의 8 번째 라인에서 정의되었습니다.

#### 4.4.1.5 MAX\_IMAGES

```
#define MAX_IMAGES 16
```

helpme.c 파일의 9 번째 라인에서 정의되었습니다.

### 4.4.2 함수 문서화

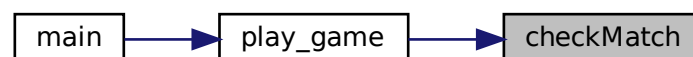
#### 4.4.2.1 checkMatch()

```
void checkMatch (
    int * selectedCards,
    int * selectedCount )
```

helpme.c 파일의 200 번째 라인에서 정의되었습니다.

```
200                                     {
201     if (*selectedCount == 2) {
202         if (board[selectedCards[0]] == board[selectedCards[1]]) {
203             score++;
204             printf("Match found! Score: %d\n", score);
205         } else {
206             revealedCards[selectedCards[0]] = 0;
207             revealedCards[selectedCards[1]] = 0;
208             printf("No match. Try again.\n");
209         }
210         *selectedCount = 0;
211         sleep(1);
212         // 사용자가 결과를 볼 수 있도록 잠시 대기
213     }
214 }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



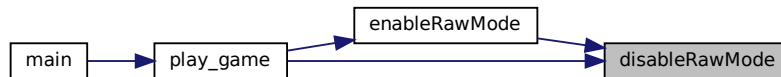
#### 4.4.2.2 disableRawMode()

```
void disableRawMode ( )
```

helpme.c 파일의 95 번째 라인에서 정의되었습니다.

```
95     {
96         tcsetattr(STDIN_FILENO, TCSAFLUSH, &orig_termios);
97     }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.4.2.3 display\_credits()

```
void display_credits ( )
```

helpme.c 파일의 234 번째 라인에서 정의되었습니다.

```
234     {
235         printf("\033[2J\033[H"); // 화면 지우기
236
237         printf("==\t윤현중\t\t==\n\n");
238         sleep(1);
239
240         printf("==\t홍솔아\t\t==\n\n");
241         sleep(1);
242
243         printf("==\t이진우\t\t==\n\n");
244         sleep(1);
245
246         printf("==\t송해든\t\t==\n\n");
247         sleep(3);
248     }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:





## 4.4.2.4 display\_images()

```
void display_images ( )
```

helpme.c 파일의 250 번째 라인에서 정의되었습니다.

```
250     {
251         printf("\033[2J\033[H"); // 화면 지우기 클리어 함수와 같음
252
253         for (int img = 0; img < MAX_IMAGES / 2; ++img) { // 각 이미지에 대해 반복
254             for (int row = 0; row < IMG_HEIGHT; ++row) { // 각 이미지의 모든 행에 대해 반복
255                 // 이미지의 현재 행을 출력
256                 printf("%s\n", images[img][row]);
257             }
258             printf("\n"); // 이미지 간 구분을 위해 빈 줄 추가
259         }
260         sleep(10); // 이미지를 잠시 표시한 후 메인 메뉴로 복귀
261     }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



## 4.4.2.5 display\_menu()

```
int display_menu ( )
```

helpme.c 파일의 216 번째 라인에서 정의되었습니다.

```
216     {
217         int menu;
218
219         system("clear");
220
221         printf("같은 그림 찾기 게임\n");
222         printf("=====\n");
223         printf("    게임 메뉴\n");
224         printf("=====\n");
225         printf("1. 게임 시작\n");
226         printf("2. 사진 체크\n");
227         printf("3. 크레딧\n");
228         printf("4. 게임 종료\n");
229
230         scanf("%d", &menu);
231         return menu;
232     }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.4.2.6 enableRawMode()

```
void enableRawMode ( )
```

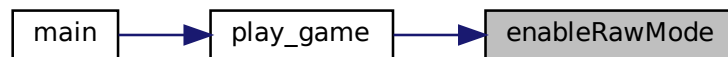
helpme.c 파일의 99 번째 라인에서 정의되었습니다.

```
99 {
100     tcgetattr(STDIN_FILENO, &orig_termios);
101     atexit(disableRawMode);
102
103     struct termios raw = orig_termios;
104     raw.c_lflag &= ~(ICANON | ECHO);
105
106     tcsetattr(STDIN_FILENO, TCSAFLUSH, &raw);
107 }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.4.2.7 initializeBoard()

```
void initializeBoard ( )
```

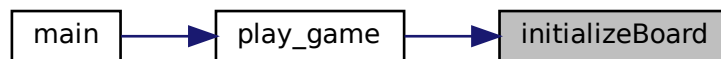
helpme.c 파일의 118 번째 라인에서 정의되었습니다.

```
118 {
119     int placements[BOARD_AREA] = {0};
120
121     // 카드를 쌍으로 배치하기 위한 초기화
122     for (int i = 0; i < BOARD_AREA / 2; i++) {
123         placements[i] = i % (MAX_IMAGES / 2);
124         placements[i + BOARD_AREA / 2] = i % (MAX_IMAGES / 2);
125     }
126
127     shuffleCards(placements, BOARD_AREA);
128
129     for (int i = 0; i < BOARD_AREA; i++) {
130         board[i] = placements[i];
131     }
132 }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.4.2.8 main()

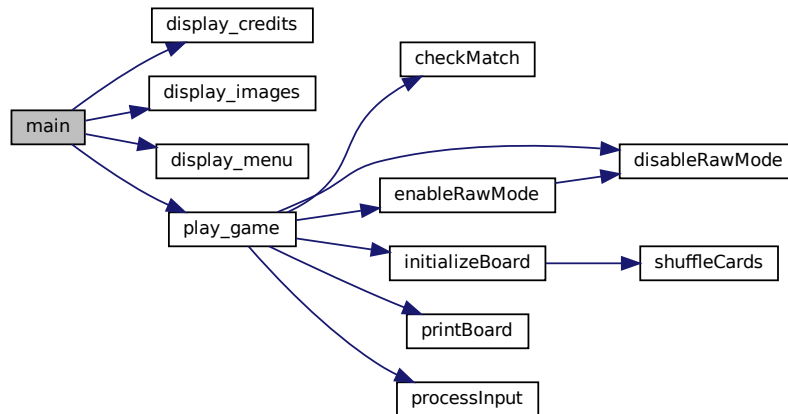
```
int main (
    void )
```

helpme.c 파일의 285 번째 라인에서 정의되었습니다.

```

285     {
286         int choice;
287         srand(time(NULL)); // 랜덤 시드 초기화
288
289         while (1) {
290             choice = display_menu();
291
292             switch(choice) {
293                 case 1:
294                     play_game();
295                     break;
296                 case 2:
297                     display_images();
298                     break;
299                 case 3:
300                     display_credits();
301                     // 크레딧 보여주고 메뉴로 돌아가야 하므로, 여기에 display_menu()를 호출하지 않음
302                     break;
303                 case 4:
304                     printf("\033[2J\033[H");
305                     printf("게임을 종료합니다.\n");
306                     exit(0); // 프로그램 종료
307             default:
308                 printf("잘못된 선택입니다.\n");
309                 sleep(1); // 잠시 대기 후 다시 메뉴를 보여줌
310             }
311         }
312
313         return 0; // 실제로는 여기에 도달하지 않음
314     }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



#### 4.4.2.9 play\_game()

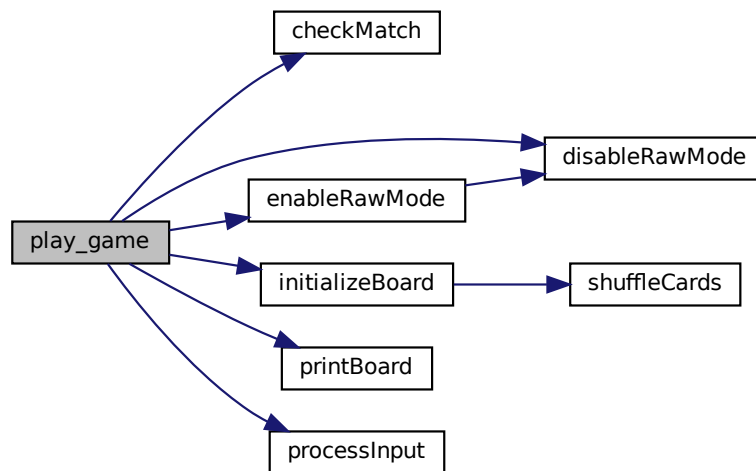
```
void play_game ( )
```

helpme.c 파일의 264 번째 라인에서 정의되었습니다.

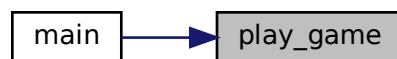
```

264     {
265         int cursorPosition = 0; // 현재 커서 위치
266         int selectedCards[2]; // 선택된 카드의 인덱스
267         int selectedCount = 0; // 현재 선택된 카드의 수
268
269         enableRawMode();
270
271         initializeBoard(); // 게임 보드 초기화
272         while (score < (BOARD_AREA / 2)) { // 모든 쌍을 찾을 때까지 반복
273             system("clear");
274             printBoard(cursorPosition); // 현재 게임 보드 출력, 커서 위치 인자 전달
275             processInput(&cursorPosition, selectedCards, &selectedCount); // 사용자 입력 처리
276             if (selectedCount == 2) { // 두 카드가 모두 선택되면
277                 checkMatch(selectedCards, &selectedCount); // 카드가 일치하는지 확인
278             }
279         }
280
281         disableRawMode();
282         printf("Congratulations! You've completed the game with a score of %d.\n", score);
283     }
  
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.4.2.10 printBoard()

```
void printBoard (
    int cursorPosition )
```

helpme.c 파일의 134 번째 라인에서 정의되었습니다.

```

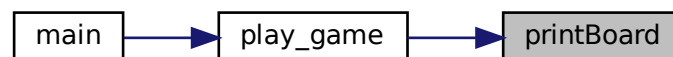
134     {
135     for (int i = 0; i < BOARD_SIZE; ++i) {
136     for (int j = 0; j < BOARD_SIZE; ++j) {
137     int cardIndex = i * BOARD_SIZE + j;
138     // 카드 상단 경계 출력
139     printf("-----");
140     }
141     printf("+\n"); // 줄의 끝에서만 경계 닫기
142
143     for (int k = 0; k < IMG_HEIGHT; ++k) { // 이미지 높이 (7) 만큼 반복
144     for (int j = 0; j < BOARD_SIZE; ++j) {
145     int cardIndex = i * BOARD_SIZE + j;
146     if (revealedCards[cardIndex]) { // 카드가 공개된 경우
147     // 실제 이미지 데이터가 있다면 여기서 이미지 라인을 출력합니다.
148     // 예제에서는 간단한 텍스트로 대체합니다.
149     for (int m = 0; m < IMG_WIDTH+1; m++){
```

```

150             printf("%c", images[board[cardIndex]][k][m]);
151         }
152     } else if (cardIndex == cursorPosition && k == 0) { // 커서 위치의 첫 번째 라인에만 선택 표시
153         printf("    선택됨    ");
154     } else {
155         // } else {
156         //     printf("|          "); // 공개되지 않은 카드
157         // }
158     } else{
159         for (int m = 0; m < IMG_WIDTH+1; m++){
160             printf("%c", images[board[cardIndex]][k][m]);
161         }
162     }
163 }
164 }
165 printf("|\\n");
166 }
167
168 for (int j = 0; j < BOARD_SIZE; ++j) {
169     // 카드 하단 경계 출력
170     printf("-----");
171 }
172 printf("+\\n");
173 }
174 }

```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.4.2.11 processInput()

```

void processInput (
    int * cursorPosition,
    int * selectedCards,
    int * selectedCount )

```

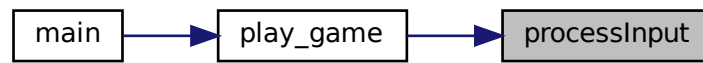
helpme.c 파일의 176 번째 라인에서 정의되었습니다.

```

176 {
177     char c = getchar();
178     switch (c) {
179         case 'w':
180             if (*cursorPosition >= BOARD_SIZE) *cursorPosition -= BOARD_SIZE;
181             break;
182         case 's':
183             if (*cursorPosition < BOARD_AREA - BOARD_SIZE) *cursorPosition += BOARD_SIZE;
184             break;
185         case 'a':
186             if (*cursorPosition % BOARD_SIZE != 0) (*cursorPosition)--;
187             break;
188         case 'd':
189             if (*cursorPosition % BOARD_SIZE != BOARD_SIZE - 1) (*cursorPosition)++;
190             break;
191         case ' ':
192             if (*selectedCount < 2 && !revealedCards[*cursorPosition]) {
193                 selectedCards[( *selectedCount)++] = *cursorPosition;
194                 revealedCards[*cursorPosition] = 1;
195             }
196             break;
197     }
198 }

```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.4.2.12 shuffleCards()

```

void shuffleCards (
    int * array,
    int size )
  
```

helpme.c 파일의 109 번째 라인에서 정의되었습니다.

```

109 {
110     for (int i = 0; i < size; i++) {
111         int j = rand() % size;
112         int temp = array[i];
113         array[i] = array[j];
114         array[j] = temp;
115     }
116 }
  
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



### 4.4.3 변수 문서화

#### 4.4.3.1 board

```
int board[BOARD_AREA] = {0}
```

helpme.c 파일의 15 번째 라인에서 정의되었습니다.

#### 4.4.3.2 images

```
char images[MAX_IMAGES][IMG_HEIGHT][IMG_WIDTH+1]
```

helpme.c 파일의 20 번째 라인에서 정의되었습니다.

#### 4.4.3.3 orig\_termios

```
struct termios orig_termios
```

helpme.c 파일의 17 번째 라인에서 정의되었습니다.

#### 4.4.3.4 revealedCards

```
int revealedCards[BOARD_AREA] = {0}
```

helpme.c 파일의 16 번째 라인에서 정의되었습니다.

#### 4.4.3.5 score

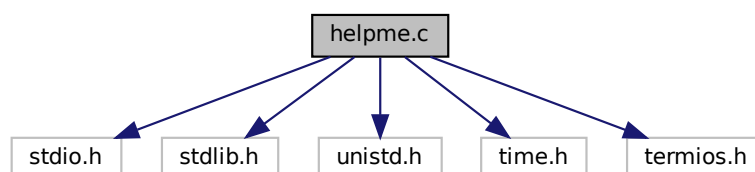
```
int score = 0
```

helpme.c 파일의 17 번째 라인에서 정의되었습니다.

## 4.5 helpme.c 파일 참조

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <time.h>
#include <termios.h>
```

helpme.c에 대한 include 의존 그래프





## 매크로

- `#define IMG_HEIGHT 7`
- `#define IMG_WIDTH 17`
- `#define MAX_IMAGES 16`
- `#define BOARD_SIZE 4`
- `#define BOARD_AREA (BOARD_SIZE * BOARD_SIZE)`

## 함수

- `void disableRawMode ()`
- `void enableRawMode ()`
- `void shuffleCards (int *array, int size)`
- `void initializeBoard ()`
- `void printBoard (int cursorPosition)`
- `void processInput (int *cursorPosition, int *selectedCards, int *selectedCount)`
- `void checkMatch (int *selectedCards, int *selectedCount)`
- `int display_menu ()`
- `void display_credits ()`
- `void display_images ()`
- `void play_game ()`
- `int main ()`

## 변수

- `int board [BOARD_AREA] = {0}`
- `int revealedCards [BOARD_AREA] = {0}`
- `int score = 0`
- `struct termios orig_termios`
- `char images [MAX_IMAGES][IMG_HEIGHT][IMG_WIDTH+1]`

### 4.5.1 매크로 문서화

#### 4.5.1.1 BOARD\_AREA

```
#define BOARD_AREA (BOARD_SIZE * BOARD_SIZE)
```

helpme.c 파일의 12 번째 라인에서 정의되었습니다.

#### 4.5.1.2 BOARD\_SIZE

```
#define BOARD_SIZE 4
```

helpme.c 파일의 11 번째 라인에서 정의되었습니다.

#### 4.5.1.3 IMG\_HEIGHT

```
#define IMG_HEIGHT 7
```

helpme.c 파일의 7 번째 라인에서 정의되었습니다.

#### 4.5.1.4 IMG\_WIDTH

```
#define IMG_WIDTH 17
```

helpme.c 파일의 8 번째 라인에서 정의되었습니다.

#### 4.5.1.5 MAX\_IMAGES

```
#define MAX_IMAGES 16
```

helpme.c 파일의 9 번째 라인에서 정의되었습니다.

### 4.5.2 함수 문서화

#### 4.5.2.1 checkMatch()

```
void checkMatch (
    int * selectedCards,
    int * selectedCount )
```

helpme.c 파일의 200 번째 라인에서 정의되었습니다.

```
200 {
201     if (*selectedCount == 2) {
202         if (board[selectedCards[0]] == board[selectedCards[1]]) {
203             score++;
204             printf("Match found! Score: %d\n", score);
205         } else {
206             revealedCards[selectedCards[0]] = 0;
207             revealedCards[selectedCards[1]] = 0;
208             printf("No match. Try again.\n");
209         }
210         *selectedCount = 0;
211         sleep(1);
212         // 사용자가 결과를 볼 수 있도록 잠시 대기
213     }
214 }
```

#### 4.5.2.2 disableRawMode()

```
void disableRawMode ( )
```

helpme.c 파일의 95 번째 라인에서 정의되었습니다.

```
95     {
96         tcsetattr(STDIN_FILENO, TCSAFLUSH, &orig_termios);
97     }
```

#### 4.5.2.3 display\_credits()

```
void display_credits ( )
```

helpme.c 파일의 234 번째 라인에서 정의되었습니다.

```
234     {
235         printf("\033[2J\033[H"); // 화면 지우기
236
237         printf("==\t윤현중\t\t==\n\n");
238         sleep(1);
239
240         printf("==\t홍술아\t\t==\n\n");
241         sleep(1);
242
243         printf("==\t이진우\t\t==\n\n");
244         sleep(1);
245
246         printf("==\t송해든\t\t==\n\n");
247         sleep(3);
248     }
```

#### 4.5.2.4 display\_images()

```
void display_images ( )
```

helpme.c 파일의 250 번째 라인에서 정의되었습니다.

```
250     {
251         printf("\033[2J\033[H"); // 화면 지우기 클리어 함수와 같음
252
253         for (int img = 0; img < MAX_IMAGES / 2; ++img) { // 각 이미지에 대해 반복
254             for (int row = 0; row < IMG_HEIGHT; ++row) { // 각 이미지의 모든 행에 대해 반복
255                 // 이미지의 현재 행을 출력
256                 printf("%s\n", images[img][row]);
257             }
258             printf("\n"); // 이미지 간 구분을 위해 빈 줄 추가
259         }
260         sleep(10); // 이미지를 잠시 표시한 후 메인 메뉴로 복귀
261     }
```

#### 4.5.2.5 display\_menu()

```
int display_menu ( )
```

helpme.c 파일의 216 번째 라인에서 정의되었습니다.

```
216     {
217         int menu;
218
219         system("clear");
220
221         printf("같은 그림 찾기 게임\n");
222         printf("=====\n");
223         printf("      게임 메뉴\n");
224         printf("=====\n");
225         printf("1. 게임 시작\n");
226         printf("2. 사진 체크\n");
227         printf("3. 크레딧\n");
228         printf("4. 게임 종료\n");
229
230         scanf("%d", &menu);
231         return menu;
232     }
```

#### 4.5.2.6 enableRawMode()

```
void enableRawMode ( )
```

helpme.c 파일의 99 번째 라인에서 정의되었습니다.

```
99      {
100      tcgetattr(STDIN_FILENO, &orig_termios);
101      atexit(disableRawMode);
102
103      struct termios raw = orig_termios;
104      raw.c_lflag &= ~(ICANON | ECHO);
105
106      tcsetattr(STDIN_FILENO, TCSAFLUSH, &raw);
107 }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



#### 4.5.2.7 initializeBoard()

```
void initializeBoard ( )
```

helpme.c 파일의 118 번째 라인에서 정의되었습니다.

```
118      {
119      int placements[BOARD_AREA] = {0};
120
121      // 카드를 쌍으로 배치하기 위한 초기화
122      for (int i = 0; i < BOARD_AREA / 2; i++) {
123          placements[i] = i % (MAX_IMAGES / 2);
124          placements[i + BOARD_AREA / 2] = i % (MAX_IMAGES / 2);
125      }
126
127      shuffleCards(placements, BOARD_AREA);
128
129      for (int i = 0; i < BOARD_AREA; i++) {
130          board[i] = placements[i];
131      }
132 }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



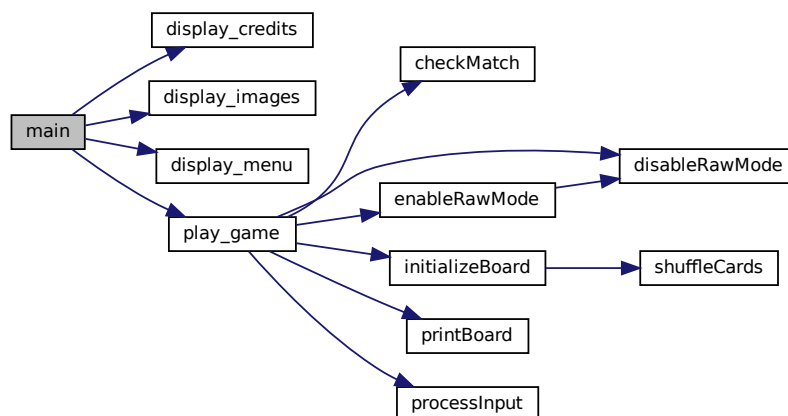
## 4.5.2.8 main()

```
int main (
    void )
```

helpme.c 파일의 285 번째 라인에서 정의되었습니다.

```
285 {
286     int choice;
287     srand(time(NULL)); // 랜덤 시드 초기화
288
289     while (1) {
290         choice = display_menu();
291
292         switch(choice) {
293             case 1:
294                 play_game();
295                 break;
296             case 2:
297                 display_images();
298                 break;
299             case 3:
300                 display_credits();
301                 // 크레딧 보여주고 메뉴로 돌아가야 하므로, 여기에 display_menu()를 호출하지 않음
302                 break;
303             case 4:
304                 printf("\033[2J\033[H");
305                 printf("게임을 종료합니다.\n");
306                 exit(0); // 프로그램 종료
307             default:
308                 printf("잘못된 선택입니다.\n");
309                 sleep(1); // 잠시 대기 후 다시 메뉴를 보여줌
310         }
311     }
312
313     return 0; // 실제로는 여기에 도달하지 않음
314 }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



## 4.5.2.9 play\_game()

```
void play_game ( )
```

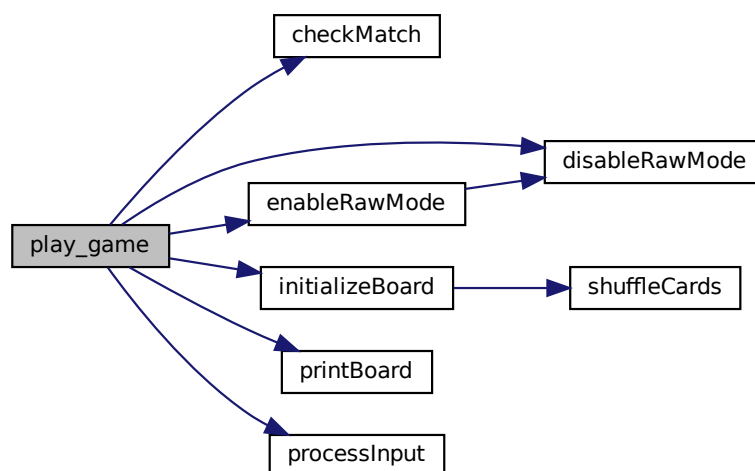
helpme.c 파일의 264 번째 라인에서 정의되었습니다.

```

264         {
265             int cursorPosition = 0; // 현재 커서 위치
266             int selectedCards[2]; // 선택된 카드의 인덱스
267             int selectedCount = 0; // 현재 선택된 카드의 수
268
269             enableRawMode();
270
271             initializeBoard(); // 게임 보드 초기화
272             while (score < (BOARD_AREA / 2)) { // 모든 쌍을 찾을 때까지 반복
273                 system("clear");
274                 printBoard(cursorPosition); // 현재 게임 보드 출력, 커서 위치 인자 전달
275                 processInput(&cursorPosition, selectedCards, &selectedCount); // 사용자 입력 처리
276                 if (selectedCount == 2) { // 두 카드가 모두 선택되면
277                     checkMatch(selectedCards, &selectedCount); // 카드가 일치하는지 확인
278                 }
279             }
280
281             disableRawMode();
282             printf("Congratulations! You've completed the game with a score of %d.\n", score);
283 }

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



#### 4.5.2.10 printBoard()

```

void printBoard (
    int cursorPosition )

```

helpme.c 파일의 134 번째 라인에서 정의되었습니다.

```

134         {
135             for (int i = 0; i < BOARD_SIZE; ++i) {
136                 for (int j = 0; j < BOARD_SIZE; ++j) {
137                     int cardIndex = i * BOARD_SIZE + j;
138                     // 카드 상단 경계 출력
139                     printf("-----");
140                 }
141                 printf("+\n"); // 줄의 끝에서만 경계 닫기
142
143                 for (int k = 0; k < IMG_HEIGHT; ++k) { // 이미지 높이(7)만큼 반복
144                     for (int j = 0; j < BOARD_SIZE; ++j) {
145                         int cardIndex = i * BOARD_SIZE + j;
146                         if (revealedCards[cardIndex]) { // 카드가 공개된 경우
147                             // 실제 이미지 데이터가 있다면 여기서 이미지 라인을 출력합니다.

```

```

148             // 예제에서는 간단한 텍스트로 대체합니다.
149             for (int m = 0; m < IMG_WIDTH+1; m++){
150                 printf("%c", images[board[cardIndex]][k][m]);
151             }
152         } else if (cardIndex == cursorPosition && k == 0) { // 커서 위치의 첫 번째 라인에만 선택 표시
153             printf("      선택됨      ");
154         } // } else {
155         //     printf("|                "); // 공개되지 않은 카드
156         // }
157     } else{
158         for (int m = 0; m < IMG_WIDTH+1; m++){
159             printf("%c", images[board[cardIndex]][k][m]);
160         }
161     }
162 }
163 }
164 }
165 printf("|\\n");
166 }
167
168 for (int j = 0; j < BOARD_SIZE; ++j) {
169     // 카드 하단 경계 출력
170     printf("-----");
171 }
172 printf("+\\n");
173 }
174 }

```

#### 4.5.2.11 processInput()

```

void processInput (
    int * cursorPosition,
    int * selectedCards,
    int * selectedCount )

```

helpme.c 파일의 176 번째 라인에서 정의되었습니다.

```

176
177 char c = getchar();
178 switch (c) {
179     case 'w':
180         if (*cursorPosition >= BOARD_SIZE) *cursorPosition -= BOARD_SIZE;
181         break;
182     case 's':
183         if (*cursorPosition < BOARD_AREA - BOARD_SIZE) *cursorPosition += BOARD_SIZE;
184         break;
185     case 'a':
186         if (*cursorPosition % BOARD_SIZE != 0) (*cursorPosition)--;
187         break;
188     case 'd':
189         if (*cursorPosition % BOARD_SIZE != BOARD_SIZE - 1) (*cursorPosition)++;
190         break;
191     case ' ':
192         if (*selectedCount < 2 && !revealedCards[*cursorPosition]) {
193             selectedCards[*selectedCount++] = *cursorPosition;
194             revealedCards[*cursorPosition] = 1;
195         }
196         break;
197 }
198 }

```

#### 4.5.2.12 shuffleCards()

```

void shuffleCards (
    int * array,
    int size )

```

helpme.c 파일의 109 번째 라인에서 정의되었습니다.

```

109
110 for (int i = 0; i < size; i++) {
111     int j = rand() % size;
112     int temp = array[i];
113     array[i] = array[j];
114     array[j] = temp;
115 }
116 }

```

### 4.5.3 변수 문서화

#### 4.5.3.1 board

```
int board[BOARD_AREA] = {0}
```

helpme.c 파일의 15 번째 라인에서 정의되었습니다.

#### 4.5.3.2 images

```
char images[MAX_IMAGES][IMG_HEIGHT][IMG_WIDTH+1]
```

helpme.c 파일의 20 번째 라인에서 정의되었습니다.

#### 4.5.3.3 orig\_termios

```
struct termios orig_termios
```

helpme.c 파일의 17 번째 라인에서 정의되었습니다.

#### 4.5.3.4 revealedCards

```
int revealedCards[BOARD_AREA] = {0}
```

helpme.c 파일의 16 번째 라인에서 정의되었습니다.

#### 4.5.3.5 score

```
int score = 0
```

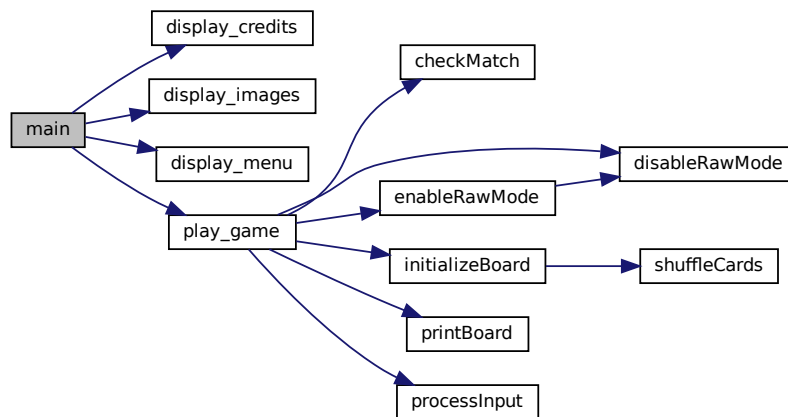
helpme.c 파일의 17 번째 라인에서 정의되었습니다.







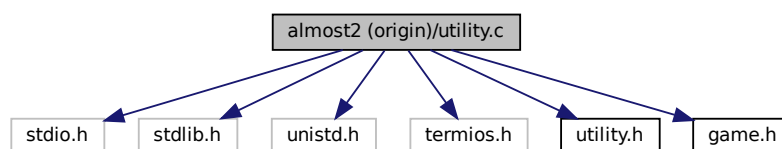
이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



## 4.7 almost2 (origin)/utility.c 파일 참조

```

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <termios.h>
#include "utility.h"
#include "game.h"
utility.c에 대한 include 의존 그래프
  
```



### 함수

- void [disableRawMode](#) ()
- void [enableRawMode](#) ()
- void [display\\_credits](#) ()
- void [display\\_images](#) ()

### 변수

- struct termios [orig\\_termios](#)

### 4.7.1 함수 문서화

#### 4.7.1.1 disableRawMode()

void disableRawMode ( )

utility.c 파일의 10 번째 라인에서 정의되었습니다.

```
11 {
12     tcsetattr(STDIN_FILENO, TCSAFLUSH, &orig_termios);
13 }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.7.1.2 display\_credits()

void display\_credits ( )

utility.c 파일의 27 번째 라인에서 정의되었습니다.

```
28 {
29     printf("\033[2J\033[H"); // Clear screen
30
31     printf("\n");
32     printf("                Sola HONG  Data Manager\n");
33     sleep(1);
34     printf("                Subtitle/Translate\n");
35     sleep(1);
36     printf("                Line Producer\n");
37     sleep(1);
38     printf("\n");
39     printf("                Haedeun SONG  Production Assistant\n");
40     sleep(1);
41     printf("                Art Design\n");
42     sleep(1);
43     printf("                Tech Engineer\n");
44     sleep(1);
45     printf("\n");
46     printf("                Jinwoo LEE   Producer & Director\n");
47     sleep(1);
48     printf("                Motion Graphic Design\n");
49     sleep(1);
50     printf("                Supervisor\n");
51     sleep(1);
52     printf("\n");
53     printf("                Hyunjoong YOON Director\n");
54     sleep(1);
55     printf("                Art Design\n");
56     sleep(1);
57     printf("                Coordinating\n");
58     printf("\n");
59     sleep(1);
60     printf("                Present By Team IoT3, Thank you!\n");
61     sleep(1);
62     printf("\n");
63
64     sleep(600);
65 }
```

### 4.7.1.3 display\_images()

```
void display_images ( )
```

utility.c 파일의 68 번째 라인에서 정의되었습니다.

```
69 {
70     printf("\033[2J\033[H"); // 화면 지우기 클리어 함수와 같음
71
72     for (int img = 0; img < MAX_IMAGES / 2; ++img)
73     { // 각 이미지에 대해 반복
74         for (int row = 0; row < IMG_HEIGHT; ++row)
75         { // 각 이미지의 모든 행에 대해 반복
76             // 이미지의 현재 행을 출력
77             printf("%s\n", images[img][row]);
78         }
79         printf("\n"); // 이미지 간 구분을 위해 빈 줄 추가
80     }
81     sleep(30); // 이미지를 잠시 표시한 후 메인 메뉴로 복귀
82 }
```

### 4.7.1.4 enableRawMode()

```
void enableRawMode ( )
```

utility.c 파일의 15 번째 라인에서 정의되었습니다.

```
16 {
17     tcgetattr(STDIN_FILENO, &orig_termios);
18     atexit(disableRawMode);
19
20     struct termios raw = orig_termios;
21     raw.c_lflag &= ~(ICANON | ECHO);
22
23     tcsetattr(STDIN_FILENO, TCSAFLUSH, &raw);
24 }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



## 4.7.2 변수 문서화

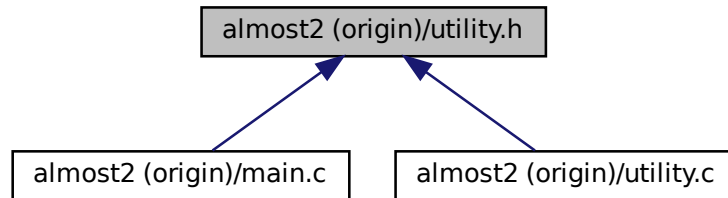
### 4.7.2.1 orig\_termios

```
struct termios orig_termios
```

utility.c 파일의 1 번째 라인에서 정의되었습니다.

## 4.8 almost2 (origin)/utility.h 파일 참조

이 그래프는 이 파일을 직/간접적으로 include 하는 파일들을 보여줍니다.:



### 함수

- void `enableRawMode` ()
- void `disableRawMode` ()
- void `display_credits` ()
- void `display_images` ()

### 4.8.1 함수 문서화

#### 4.8.1.1 `disableRawMode()`

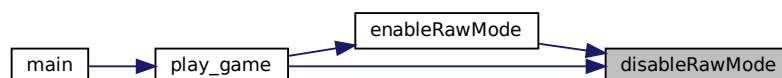
```
void disableRawMode ( )
```

helpme.c 파일의 95 번째 라인에서 정의되었습니다.

```

95     {
96         tcsetattr(STDIN_FILENO, TCSAFLUSH, &orig_termios);
97     }
  
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



## 4.8.1.2 display\_credits()

```
void display_credits ( )
```

helpme.c 파일의 234 번째 라인에서 정의되었습니다.

```
234     {
235         printf("\033[2J\033[H"); // 화면 지우기
236
237         printf("==\t윤현중\t\t==\n\n");
238         sleep(1);
239
240         printf("==\t홍솔아\t\t==\n\n");
241         sleep(1);
242
243         printf("==\t이진우\t\t==\n\n");
244         sleep(1);
245
246         printf("==\t송해든\t\t==\n\n");
247         sleep(3);
248     }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



## 4.8.1.3 display\_images()

```
void display_images ( )
```

helpme.c 파일의 250 번째 라인에서 정의되었습니다.

```
250     {
251         printf("\033[2J\033[H"); // 화면 지우기 클리어 함수와 같음
252
253         for (int img = 0; img < MAX_IMAGES / 2; ++img) { // 각 이미지에 대해 반복
254             for (int row = 0; row < IMG_HEIGHT; ++row) { // 각 이미지의 모든 행에 대해 반복
255                 // 이미지의 현재 행을 출력
256                 printf("%s\n", images[img][row]);
257             }
258             printf("\n"); // 이미지 간 구분을 위해 빈 줄 추가
259         }
260         sleep(10); // 이미지를 잠시 표시한 후 메인 메뉴로 복귀
261     }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.8.1.4 enableRawMode()

```
void enableRawMode ( )
```

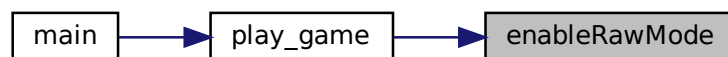
helpme.c 파일의 99 번째 라인에서 정의되었습니다.

```
99      {
100      tcgetattr(STDIN_FILENO, &orig_termios);
101      atexit(disableRawMode);
102
103      struct termios raw = orig_termios;
104      raw.c_lflag &= ~(ICANON | ECHO);
105
106      tcsetattr(STDIN_FILENO, TCSAFLUSH, &raw);
107 }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



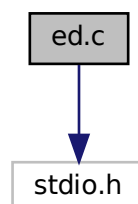
이 함수를 호출하는 함수들에 대한 그래프입니다.:



## 4.9 ed.c 파일 참조

```
#include <stdio.h>
```

ed.c에 대한 include 의존 그래프







## 함수

- int `display_menu` ()
- void `display_credits` ()
- int `main` (void)
- void `display_images` ()
- void `shuffleCards` (int \*array, int size)
- void `initializeBoard` (int boardSize)
- void `play_game` ()

## 변수

- const char \* `images` [`MAX_IMAGES`]

### 4.10.1 매크로 문서화

#### 4.10.1.1 MAX\_BOARD

```
#define MAX_BOARD 6
```

loT3\_ver\_0.5.c 파일의 8 번째 라인에서 정의되었습니다.

#### 4.10.1.2 MAX\_IMAGES

```
#define MAX_IMAGES 20
```

loT3\_ver\_0.5.c 파일의 7 번째 라인에서 정의되었습니다.

### 4.10.2 함수 문서화

### 4.10.2.1 display\_credits()

void display\_credits ( )

IoT3\_ver\_0.5.c 파일의 276 번째 라인에서 정의되었습니다.

```

277 {
278     printf("\033[2J\033[H"); // Clear screen
279
280     printf("\n");
281     printf("                Sola HONG Data Manager\n");
282     sleep(1);
283     printf("                Subtitle/Translate\n");
284     sleep(1);
285     printf("                Line Producer\n");
286     sleep(1);
287     printf("\n");
288
289     printf("                Haedeun SONG Production Assistant\n");
290     sleep(1);
291     printf("                Art Design\n");
292     sleep(1);
293     printf("                Tech Engineer\n");
294     sleep(1);
295     printf("\n");
296
297     printf("                Jinwoo LEE Producer & Director\n");
298     sleep(0.8);
299     printf("                Motion Graphic Design\n");
300     sleep(1);
301     printf("                Supervisor\n");
302     sleep(1);
303     printf("\n");
304
305     printf("                Hyunjoong YOON Director\n");
306     sleep(1);
307     printf("                Art Design\n");
308     sleep(1);
309     printf("                Coordinating\n");
310     printf("\n");
311     sleep(1);
312     printf("                Present By Team IoT3...Thank you...\n");
313     sleep(1);
314     printf("\n");
315
316     sleep(600);
317 }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



### 4.10.2.2 display\_images()

void display\_images ( )

IoT3\_ver\_0.5.c 파일의 326 번째 라인에서 정의되었습니다.

```

327 {
328     printf("\033[2J\033[H"); // 화면 지우기 클리어 함수와 같음
329
330     for (int i = 0; i < MAX_IMAGES; ++i)
331     {
332         printf("%s\n", images[i]); // 이미지 출력
333     }
334     sleep(10); // 이미지를 잠시 표시한 후 메인 메뉴로 복귀
335 }
```



```

261     printf("                Find the Same Picture\n");
262     printf("                -----\n");
263     printf("                M e n u\n");
264     printf("                -----\n");
265     printf("                1.   Game Start\n");
266     printf("                2.   Card Check\n");
267     printf("                3.   Credit\n");
268     printf("                4.   End\n");
269
270     scanf("%d", &menu);
271     return menu;
272 }

```

#### 4.10.2.4 initializeBoard()

```

void initializeBoard (
    int boardSize )

```

loT3\_ver\_0.5.c 파일의 348 번째 라인에서 정의되었습니다.

```

349 {
350     int board[MAX_BOARD][MAX_BOARD] = {0};
351     int cardIndexes[MAX_IMAGES]; // 이 배열은 사용 가능한 카드 인덱스를 저장합니다.
352
353     // 각 카드 인덱스를 초기화
354     for (int i = 0; i < MAX_IMAGES; i++)
355     {
356         cardIndexes[i] = i;
357     }
358
359     // 카드 인덱스를 셔플
360     shuffleCards(cardIndexes, MAX_IMAGES);
361
362     // 보드에 카드 배치
363     int cardCount = 0;
364     for (int i = 0; i < boardSize; i++)
365     {
366         for (int j = 0; j < boardSize; j += 2)
367         {
368             // 같은 카드를 2개씩 배치
369             board[i][j] = board[i][j + 1] = cardIndexes[cardCount++];
370             if (cardCount >= boardSize * boardSize / 2)
371             {
372                 // 필요한 카드 수에 도달하면 중단
373                 break;
374             }
375         }
376         if (cardCount >= boardSize * boardSize / 2)
377         {
378             break;
379         }
380     }
381
382     // 카드 출력
383     for (int i = 0; i < boardSize; i++)
384     {
385         for (int j = 0; j < boardSize; j++)
386         {
387             printf("%s ", images[board[i][j]]);
388         }
389         printf("\n");
390     }
391 }

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.10.2.5 main()

```
int main (
    void )
```

IoT3\_ver\_0.5.c 파일의 319 번째 라인에서 정의되었습니다.

```
320 {
321     display_credits();
322
323     return 0;
324 }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



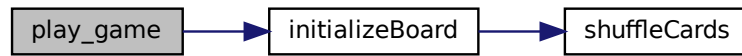
#### 4.10.2.6 play\_game()

```
void play_game ( )
```

IoT3\_ver\_0.5.c 파일의 393 번째 라인에서 정의되었습니다.

```
394 {
395     printf("\033[2J\033[H"); // 화면 지우기 및 커서 이동
396     printf("게임을 시작합니다...\n\n");
397
398     // 게임 보드 초기화 및 카드 배치
399     initializeBoard(MAX_BOARD);
400
401     // 게임 로직 구현 (사용자 입력 받기, 선택된 카드 비교 등)
402     // ...
403 }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



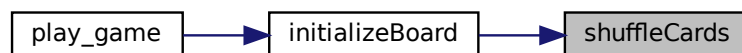
#### 4.10.2.7 shuffleCards()

```
void shuffleCards (
    int * array,
    int size )
```

IoT3\_ver\_0.5.c 파일의 337 번째 라인에서 정의되었습니다.

```
338 {
339     for (int i = 0; i < size; i++)
340     {
341         int j = rand() % size;
342         int temp = array[i];
343         array[i] = array[j];
344         array[j] = temp;
345     }
346 }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



### 4.10.3 변수 문서화

#### 4.10.3.1 images

```
const char* images[MAX_IMAGES]
```

초기값:

```
= {
    (" |-----| \n"),
    (" |         | \n"),
    (" |      (\\_\\_) | \n"),
    (" |      (>'.'<) | \n"),
    (" |      (\\_)_ (\\_) | \n"),
}
```

```

(" |          | \n" ),
(" |-----| \n" ),
(" |-----| \n" ),
(" |          | \n" ),
(" |      (\\_/_/ ) | \n" ),
(" |      (>' , '<) | \n" ),
(" |      (\" ) _ (\" ) | \n" ),
(" |          | \n" ),
(" |-----| \n" ),
}

```

IoT3\_ver\_0.5.c 파일의 177 번째 라인에서 정의되었습니다.

## 4.11 README.md 파일 참조

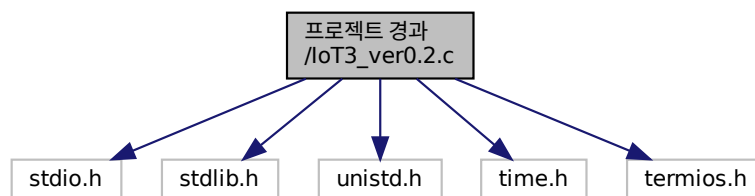
## 4.12 프로젝트 경과/README.md 파일 참조

## 4.13 프로젝트 경과/IoT3\_ver0.2.c 파일 참조

```

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <time.h>
#include <termios.h>
IoT3_ver0.2.c에 대한 include 의존 그래프

```



## 매크로

- #define MAX\_IMAGES 20
- #define MAX\_BOARD 6

## 함수

- int display\_menu ()
- void display\_credits ()
- void display\_images ()
- void shuffleCards (int \*array, int size)
- void initializeBoard (int boardSize)
- void play\_game ()
- int main ()



## 변수

- `const char * images [MAX_IMAGES]`

### 4.13.1 매크로 문서화

#### 4.13.1.1 MAX\_BOARD

```
#define MAX_BOARD 6
```

loT3\_ver0.2.c 파일의 8 번째 라인에서 정의되었습니다.

#### 4.13.1.2 MAX\_IMAGES

```
#define MAX_IMAGES 20
```

loT3\_ver0.2.c 파일의 7 번째 라인에서 정의되었습니다.

### 4.13.2 함수 문서화

#### 4.13.2.1 display\_credits()

```
void display_credits ( )
```

loT3\_ver0.2.c 파일의 254 번째 라인에서 정의되었습니다.

```
254 {
255     printf("\033[2J\033[H"); // 화면 지우기
256
257     printf("==\t윤현중\t\t==\n\n");
258     sleep(1);
259
260     printf("==\t홍솔아\t\t==\n\n");
261     sleep(1);
262
263     printf("==\t이진우\t\t==\n\n");
264     sleep(1);
265
266     printf("==\t송해든\t\t==\n\n");
267     sleep(3);
268 }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:





```

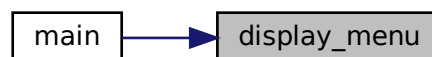
203     printf("\033[0m");
204     printf("\n");
205     printf("\n");
206     printf("\n");
207     printf("
208     printf("
209     printf("
210     printf("
211     printf("
212     printf("
213     printf("
214     printf("
215
216
217
218     printf("\n");
219     printf("\033[94m
220     printf("\033[94m
221     printf("\n");
222     printf("
223     printf("\033[94m
224     printf("
225     printf("\033[94m
226     printf("
227     printf("\033[94m
228     printf("
229     printf("\033[94m
230     printf("
231     printf("\033[94m
232     printf("\033[94m
233 */
234
235
236
237     printf("\033[0m");
238     printf("\n");
239     printf("\n");
240     printf("\n");
241     printf("
242     printf("
243     printf("
244     printf("
245     printf("
246     printf("
247     printf("
248     printf("
249
250     scanf("%d", &menu);
251     return menu;
252 }

```

같은 그림 찾기 게임\n");  
===== \n");  
게임 메뉴\n");  
===== \n");  
1. 게임 시작\n");  
2. 사진 체크\n");  
3. 크레딧\n");  
4. 게임 종료\n");

Find the Same Picture\n");  
----- \n");  
M e n u\n");  
----- \n");  
1. Game Start\n");  
2. Card Check\n");  
3. Credit\n");  
4. End\n");

이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.13.2.4 initializeBoard()

```

void initializeBoard (
    int boardSize )

```

IoT3\_ver0.2.c 파일의 288 번째 라인에서 정의되었습니다.

```

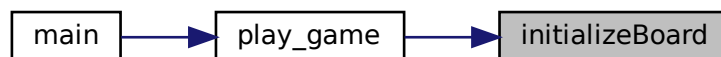
288     {
289         int board[MAX_BOARD][MAX_BOARD] = {0};
290         int cardIndexes[MAX_IMAGES]; // 이 배열은 사용 가능한 카드 인덱스를 저장합니다.
291
292         // 각 카드 인덱스를 초기화
293         for (int i = 0; i < MAX_IMAGES; i++) {
294             cardIndexes[i] = i;
295         }
296
297         // 카드 인덱스를 셔플
298         shuffleCards(cardIndexes, MAX_IMAGES);
299
300         // 보드에 카드 배치
301         int cardCount = 0;
302         for (int i = 0; i < boardSize; i++) {
303             for (int j = 0; j < boardSize; j += 2) {
304                 // 같은 카드를 2개씩 배치
305                 board[i][j] = board[i][j + 1] = cardIndexes[cardCount++];
306                 if (cardCount >= boardSize * boardSize / 2) {
307                     // 필요한 카드 수에 도달하면 중단
308                     break;
309                 }
310             }
311             if (cardCount >= boardSize * boardSize / 2) {
312                 break;
313             }
314         }
315
316         // 카드 출력
317         for (int i = 0; i < boardSize; i++) {
318             for (int j = 0; j < boardSize; j++) {
319                 printf("%s ", images[board[i][j]]);
320             }
321             printf("\n");
322         }
323     }

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.13.2.5 main()

```

int main (
    void )

```

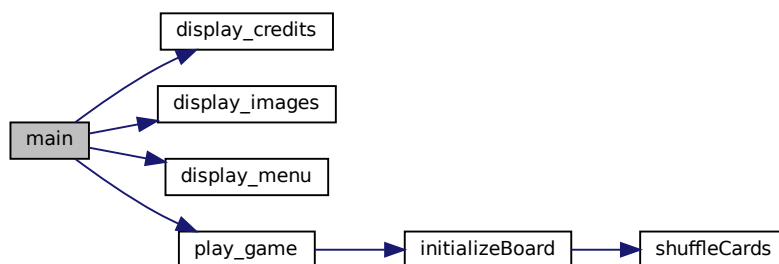
loT3\_ver0.2.c 파일의 337 번째 라인에서 정의되었습니다.

```

337 {
338     int choice;
339     srand(time(NULL)); // 랜덤 시드 초기화
340
341     while (1) {
342         choice = display_menu();
343
344         switch(choice) {
345             case 1:
346                 play_game();
347                 break;
348             case 2:
349                 display_images();
350                 break;
351             case 3:
352                 display_credits();
353                 // 크레딧 보여주고 메뉴로 돌아가야 하므로, 여기에 display_menu()를 호출하지 않음.
354                 break;
355             case 4:
356                 printf("\033[2J\033[H");
357                 printf("게임을 종료합니다.\n");
358                 exit(0); // 프로그램 종료
359             default:
360                 printf("잘못된 선택입니다.\n");
361                 sleep(1); // 잠시 대기 후 다시 메뉴를 보여줌
362         }
363     }
364
365     return 0; // 실제로는 여기에 도달하지 않음
366
367
368
369 }

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



#### 4.13.2.6 play\_game()

```
void play_game ( )
```

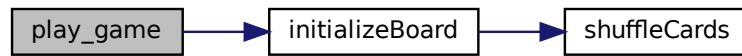
loT3\_ver0.2.c 파일의 325 번째 라인에서 정의되었습니다.

```

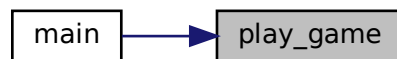
325 {
326     printf("\033[2J\033[H"); // 화면 지우기 및 커서 이동
327     printf("게임을 시작합니다...\n\n");
328
329     // 게임 보드 초기화 및 카드 배치
330     initializeBoard(MAX_BOARD);
331
332     // 게임 로직 구현 (사용자 입력 받기, 선택된 카드 비교 등)
333     // ...
334 }

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.13.2.7 shuffleCards()

```
void shuffleCards (
    int * array,
    int size )
```

IoT3\_ver0.2.c 파일의 279 번째 라인에서 정의되었습니다.

```
279 {
280     for (int i = 0; i < size; i++) {
281         int j = rand() % size;
282         int temp = array[i];
283         array[i] = array[j];
284         array[j] = temp;
285     }
286 }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.13.3 변수 문서화

#### 4.13.3.1 images

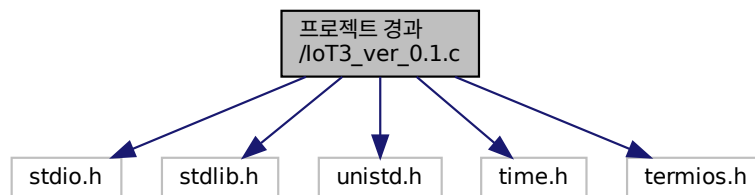
```
const char* images[MAX_IMAGES]
```

loT3\_ver0.2.c 파일의 10 번째 라인에서 정의되었습니다.

### 4.14 프로젝트 경과/loT3\_ver\_0.1.c 파일 참조

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <time.h>
#include <termios.h>
```

loT3\_ver\_0.1.c에 대한 include 의존 그래프



#### 매크로

- #define MAX\_IMAGES 20
- #define MAX\_BOARD 6

#### 함수

- int display\_menu ()
- void display\_credits ()
- void display\_images ()
- void shuffleCards (int \*array, int size)
- void initializeBoard (int boardSize)
- void play\_game ()
- int main ()

#### 변수

- const char \* images [MAX\_IMAGES]

#### 4.14.1 매크로 문서화

#### 4.14.1.1 MAX\_BOARD

```
#define MAX_BOARD 6
```

IoT3\_ver\_0.1.c 파일의 8 번째 라인에서 정의되었습니다.

#### 4.14.1.2 MAX\_IMAGES

```
#define MAX_IMAGES 20
```

IoT3\_ver\_0.1.c 파일의 7 번째 라인에서 정의되었습니다.

### 4.14.2 함수 문서화

#### 4.14.2.1 display\_credits()

```
void display_credits ( )
```

IoT3\_ver\_0.1.c 파일의 201 번째 라인에서 정의되었습니다.

```
201     {
202     printf("\033[2J\033[H"); // 화면 지우기
203
204     printf("==\t윤현중\t\t==\n\n");
205     sleep(1);
206
207     printf("==\t홍술아\t\t==\n\n");
208     sleep(1);
209
210     printf("==\t이진우\t\t==\n\n");
211     sleep(1);
212
213     printf("==\t송해든\t\t==\n\n");
214     sleep(3);
215 }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:





## 4.14.2.2 display\_images()

```
void display_images ( )
```

loT3\_ver\_0.1.c 파일의 217 번째 라인에서 정의되었습니다.

```
217     {
218     printf("\033[2J\033[H"); // 화면 지우기 클리어 함수와 같음
219
220     for (int i = 0; i < MAX_IMAGES; ++i) {
221     printf("%s\n", images[i]); // 이미지 출력
222     }
223     sleep(10); // 이미지를 잠시 표시한 후 메인 메뉴로 복귀
224 }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



## 4.14.2.3 display\_menu()

```
int display_menu ( )
```

loT3\_ver\_0.1.c 파일의 174 번째 라인에서 정의되었습니다.

```
174     {
175     int menu;
176
177     system("clear");
178
179     printf("\n");
180     printf("
181     printf("
182     printf("
183     printf("
184     printf("
185
186     printf("\n");
187     printf("\n");
188     printf("
189     printf("
190     printf("
191     printf("
192     printf("
193     printf("
194     printf("
195     printf("
196
197     scanf("%d", &menu);
198     return menu;
199 }
```

같은 그림 찾기 게임\n");  
=====\\n");  
게임 메뉴\\n");  
=====\\n");  
1. 게임 시작\\n");  
2. 사진 체크\\n");  
3. 크레딧\\n");  
4. 게임 종료\\n");

이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.14.2.4 initializeBoard()

```
void initializeBoard (
    int boardSize )
```

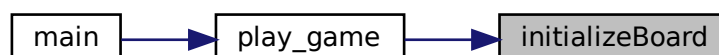
IoT3\_ver\_0.1.c 파일의 235 번째 라인에서 정의되었습니다.

```
235     {
236         int board[MAX_BOARD][MAX_BOARD] = {0};
237         int cardIndexes[MAX_IMAGES]; // 이 배열은 사용 가능한 카드 인덱스를 저장합니다.
238
239         // 각 카드 인덱스를 초기화
240         for (int i = 0; i < MAX_IMAGES; i++) {
241             cardIndexes[i] = i;
242         }
243
244         // 카드 인덱스를 셔플
245         shuffleCards(cardIndexes, MAX_IMAGES);
246
247         // 보드에 카드 배치
248         int cardCount = 0;
249         for (int i = 0; i < boardSize; i++) {
250             for (int j = 0; j < boardSize; j += 2) {
251                 // 같은 카드를 2개씩 배치
252                 board[i][j] = board[i][j + 1] = cardIndexes[cardCount++];
253                 if (cardCount >= boardSize * boardSize / 2) {
254                     // 필요한 카드 수에 도달하면 중단
255                     break;
256                 }
257             }
258             if (cardCount >= boardSize * boardSize / 2) {
259                 break;
260             }
261         }
262
263         // 카드 출력
264         for (int i = 0; i < boardSize; i++) {
265             for (int j = 0; j < boardSize; j++) {
266                 printf("%s ", images[board[i][j]]);
267             }
268             printf("\n");
269         }
270     }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



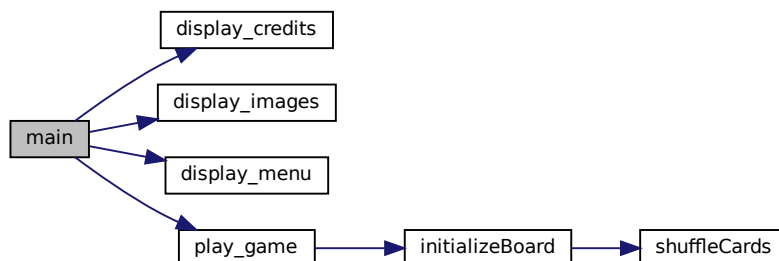
## 4.14.2.5 main()

```
int main (
    void )
```

loT3\_ver\_0.1.c 파일의 284 번째 라인에서 정의되었습니다.

```
284 {
285     int choice;
286     srand(time(NULL)); // 랜덤 시드 초기화
287
288     while (1) {
289         choice = display_menu();
290
291         switch(choice) {
292             case 1:
293                 play_game();
294                 break;
295             case 2:
296                 display_images();
297                 break;
298             case 3:
299                 display_credits();
300                 // 크레딧 보여주고 메뉴로 돌아가야 하므로, 여기에 display_menu()를 호출하지 않음.
301                 break;
302             case 4:
303                 printf("\033[2J\033[H");
304                 printf("게임을 종료합니다.\n");
305                 exit(0); // 프로그램 종료
306             default:
307                 printf("잘못된 선택입니다.\n");
308                 sleep(1); // 잠시 대기 후 다시 메뉴를 보여줌
309         }
310     }
311
312     return 0; // 실제로는 여기에 도달하지 않음
313 }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



## 4.14.2.6 play\_game()

```
void play_game ( )
```

loT3\_ver\_0.1.c 파일의 272 번째 라인에서 정의되었습니다.

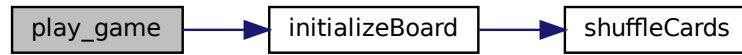
```
272 {
273     printf("\033[2J\033[H"); // 화면 지우기 및 커서 이동
274     printf("게임을 시작합니다...\n\n");
275
276     // 게임 보드 초기화 및 카드 배치
277     initializeBoard(MAX_BOARD);
278 }
```

```

279 // 게임 로직 구현 (사용자 입력 받기, 선택된 카드 비교 등)
280 // ...
281 }

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.14.2.7 shuffleCards()

```

void shuffleCards (
    int * array,
    int size )

```

IoT3\_ver\_0.1.c 파일의 226 번째 라인에서 정의되었습니다.

```

226 {
227     for (int i = 0; i < size; i++) {
228         int j = rand() % size;
229         int temp = array[i];
230         array[i] = array[j];
231         array[j] = temp;
232     }
233 }

```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



### 4.14.3 변수 문서화

#### 4.14.3.1 images

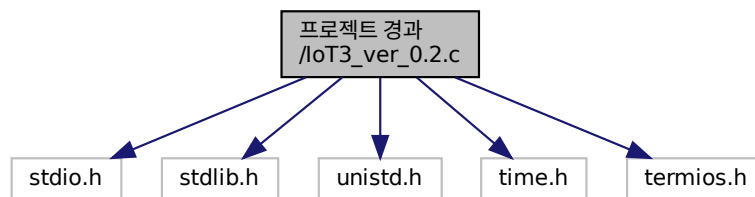
```
const char* images[MAX_IMAGES]
```

loT3\_ver\_0.1.c 파일의 10 번째 라인에서 정의되었습니다.

## 4.15 프로젝트 경과/loT3\_ver\_0.2.c 파일 참조

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <time.h>
#include <termios.h>
```

loT3\_ver\_0.2.c에 대한 include 의존 그래프



### 매크로

- #define MAX\_IMAGES 20
- #define MAX\_BOARD 6

### 함수

- int display\_menu ()
- void display\_credits ()
- void display\_images ()
- void shuffleCards (int \*array, int size)
- void initializeBoard (int boardSize)
- void play\_game ()
- int main ()

### 변수

- const char \* images [MAX\_IMAGES]

### 4.15.1 매크로 문서화

#### 4.15.1.1 MAX\_BOARD

```
#define MAX_BOARD 6
```

IoT3\_ver\_0.2.c 파일의 8 번째 라인에서 정의되었습니다.

#### 4.15.1.2 MAX\_IMAGES

```
#define MAX_IMAGES 20
```

IoT3\_ver\_0.2.c 파일의 7 번째 라인에서 정의되었습니다.

### 4.15.2 함수 문서화

#### 4.15.2.1 display\_credits()

```
void display_credits ( )
```

IoT3\_ver\_0.2.c 파일의 251 번째 라인에서 정의되었습니다.

```
251 {
252     printf("\033[2J\033[H"); // 화면 지우기
253
254     printf("==\t윤현중\t\t==\n\n");
255     sleep(1);
256
257     printf("==\t홍술아\t\t==\n\n");
258     sleep(1);
259
260     printf("==\t이진우\t\t==\n\n");
261     sleep(1);
262
263     printf("==\t송해든\t\t==\n\n");
264     sleep(3);
265 }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



## 4.15.2.2 display\_images()

```
void display_images ( )
```

loT3\_ver\_0.2.c 파일의 267 번째 라인에서 정의되었습니다.

```
267 {
268     printf("\033[2J\033[H"); // 화면 지우기 클리어 함수와 같음
269
270     for (int i = 0; i < MAX_IMAGES; ++i) {
271         printf("%s\n", images[i]); // 이미지 출력
272     }
273     sleep(10); // 이미지를 잠시 표시한 후 메인 메뉴로 복귀
274 }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



## 4.15.2.3 display\_menu()

```
int display_menu ( )
```

loT3\_ver\_0.2.c 파일의 174 번째 라인에서 정의되었습니다.

```
174 {
175     int menu;
176
177     system("clear");
178
179     /*printf("\n");
180     printf("
181     printf("
182     printf("
183     printf("
184     printf("
185
186
187     printf("\n");
188     printf("\033[32m
189     printf("\033[32m
190     printf("\033[32m
191     printf("
192     printf("\033[32m
193     printf("
194     printf("\033[32m
195     printf("
196     printf("\033[32m
197     printf("
198     printf("\033[32m
199     printf("
200     printf("\033[32m
201     printf("\033[32m
202
```





loT3\_ver\_0.2.c 파일의 285 번째 라인에서 정의되었습니다.

```

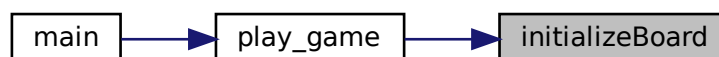
285     {
286         int board[MAX_BOARD][MAX_BOARD] = {0};
287         int cardIndexes[MAX_IMAGES]; // 이 배열은 사용 가능한 카드 인덱스를 저장합니다.
288
289         // 각 카드 인덱스를 초기화
290         for (int i = 0; i < MAX_IMAGES; i++) {
291             cardIndexes[i] = i;
292         }
293
294         // 카드 인덱스를 셔플
295         shuffleCards(cardIndexes, MAX_IMAGES);
296
297         // 보드에 카드 배치
298         int cardCount = 0;
299         for (int i = 0; i < boardSize; i++) {
300             for (int j = 0; j < boardSize; j += 2) {
301                 // 같은 카드를 2개씩 배치
302                 board[i][j] = board[i][j + 1] = cardIndexes[cardCount++];
303                 if (cardCount >= boardSize * boardSize / 2) {
304                     // 필요한 카드 수에 도달하면 중단
305                     break;
306                 }
307             }
308             if (cardCount >= boardSize * boardSize / 2) {
309                 break;
310             }
311         }
312
313         // 카드 출력
314         for (int i = 0; i < boardSize; i++) {
315             for (int j = 0; j < boardSize; j++) {
316                 printf("%s ", images[board[i][j]]);
317             }
318             printf("\n");
319         }
320     }

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.15.2.5 main()

```

int main (
    void )

```

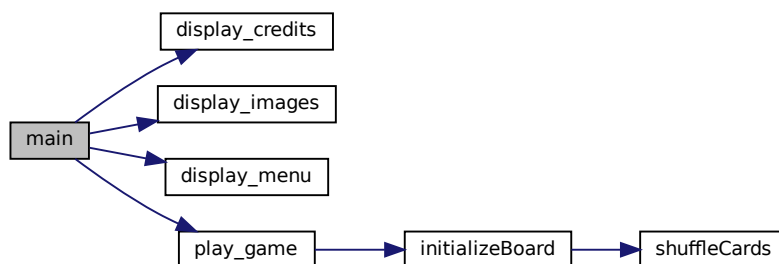
loT3\_ver\_0.2.c 파일의 334 번째 라인에서 정의되었습니다.

```

334 {
335     int choice;
336     srand(time(NULL)); // 랜덤 시드 초기화
337
338     while (1) {
339         choice = display_menu();
340
341         switch(choice) {
342             case 1:
343                 play_game();
344                 break;
345             case 2:
346                 display_images();
347                 break;
348             case 3:
349                 display_credits();
350                 // 크레딧 보여주고 메뉴로 돌아가야 하므로, 여기에 display_menu()를 호출하지 않음.
351                 break;
352             case 4:
353                 printf("\033[2J\033[H");
354                 printf("게임을 종료합니다.\n");
355                 exit(0); // 프로그램 종료
356             default:
357                 printf("잘못된 선택입니다.\n");
358                 sleep(1); // 잠시 대기 후 다시 메뉴를 보여줌
359         }
360     }
361
362     return 0; // 실제로는 여기에 도달하지 않음
363
364
365
366 }

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



#### 4.15.2.6 play\_game()

```
void play_game ( )
```

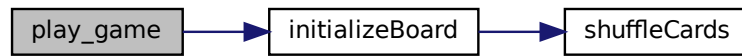
loT3\_ver\_0.2.c 파일의 322 번째 라인에서 정의되었습니다.

```

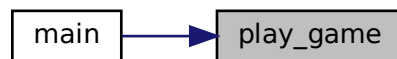
322 {
323     printf("\033[2J\033[H"); // 화면 지우기 및 커서 이동
324     printf("게임을 시작합니다...\n\n");
325
326     // 게임 보드 초기화 및 카드 배치
327     initializeBoard(MAX_BOARD);
328
329     // 게임 로직 구현 (사용자 입력 받기, 선택된 카드 비교 등)
330     // ...
331 }

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.15.2.7 shuffleCards()

```

void shuffleCards (
    int * array,
    int size )
  
```

loT3\_ver\_0.2.c 파일의 276 번째 라인에서 정의되었습니다.

```

276 {
277     for (int i = 0; i < size; i++) {
278         int j = rand() % size;
279         int temp = array[i];
280         array[i] = array[j];
281         array[j] = temp;
282     }
283 }
  
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



### 4.15.3 변수 문서화

#### 4.15.3.1 images

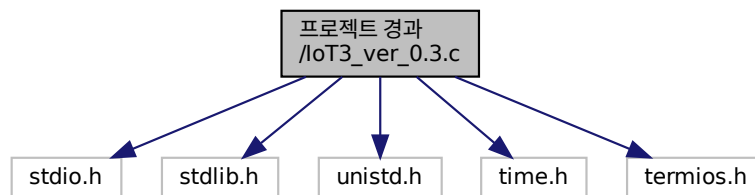
```
const char* images[MAX_IMAGES]
```

IoT3\_ver\_0.2.c 파일의 10 번째 라인에서 정의되었습니다.

### 4.16 프로젝트 경과/IoT3\_ver\_0.3.c 파일 참조

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <time.h>
#include <termios.h>
```

IoT3\_ver\_0.3.c에 대한 include 의존 그래프



#### 매크로

- #define MAX\_IMAGES 20
- #define MAX\_BOARD 6

#### 함수

- int display\_menu ()
- void display\_credits ()
- void display\_images ()
- void shuffleCards (int \*array, int size)
- void initializeBoard (int boardSize)
- void play\_game ()
- int main ()

#### 변수

- const char \* images [MAX\_IMAGES]

#### 4.16.1 매크로 문서화

#### 4.16.1.1 MAX\_BOARD

```
#define MAX_BOARD 6
```

loT3\_ver\_0.3.c 파일의 8 번째 라인에서 정의되었습니다.

#### 4.16.1.2 MAX\_IMAGES

```
#define MAX_IMAGES 20
```

loT3\_ver\_0.3.c 파일의 7 번째 라인에서 정의되었습니다.

### 4.16.2 함수 문서화

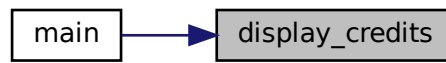
#### 4.16.2.1 display\_credits()

```
void display_credits ( )
```

loT3\_ver\_0.3.c 파일의 251 번째 라인에서 정의되었습니다.

```
251     {
252         printf("\033[2J\033[H"); // Clear screen
253
254         printf("\n");
255         printf("\n");
256         printf("                Sola HONG  Data Manager\n");
257         sleep(1);
258         printf("                Subtitle/Translate\n");
259         sleep(1);
260         printf("                Line Producer\n");
261         sleep(1);
262         printf("\n");
263
264         printf("                Haedeun SONG  Production Assistant\n");
265         sleep(1);
266         printf("                Art Design\n");
267         sleep(1);
268         printf("                Tech Engineer\n");
269         sleep(1);
270         printf("\n");
271
272         printf("                Jinwoo LEE   Producer & Director\n");
273         sleep(1);
274         printf("                Motion Graphic Design\n");
275         sleep(1);
276         printf("                Supervisor\n");
277         sleep(1);
278         printf("\n");
279
280         printf("                Hyunjoong YOON Director\n");
281         sleep(1);
282         printf("                Art Design\n");
283         sleep(1);
284         printf("                Coordinating\n");
285         printf("\n");
286         printf("\n");
287         sleep(1);
288         printf("                Present By Team IoT3...Thank you...\n");
289         sleep(1);
290         sleep(600);
291     }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.16.2.2 display\_images()

```
void display_images ( )
```

loT3\_ver\_0.3.c 파일의 293 번째 라인에서 정의되었습니다.

```

293     {
294         printf("\033[2J\033[H"); // 화면 지우기 클리어 함수와 같음
295
296         for (int i = 0; i < MAX_IMAGES; ++i) {
297             printf("%s\n", images[i]); // 이미지 출력
298         }
299         sleep(10); // 이미지를 잠시 표시한 후 메인 메뉴로 복귀
300     }
  
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.16.2.3 display\_menu()

```
int display_menu ( )
```

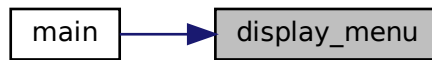
loT3\_ver\_0.3.c 파일의 174 번째 라인에서 정의되었습니다.

```

174     {
175         int menu;
176
177         system("clear");
178
179         /*printf("\n");
180         printf("
181         printf("
182         printf("
183         printf("
184         printf("
185 */
  
```

[illegible]

이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.16.2.4 initializeBoard()

```
void initializeBoard (
    int boardSize )
```

IoT3\_ver\_0.3.c 파일의 311 번째 라인에서 정의되었습니다.

```

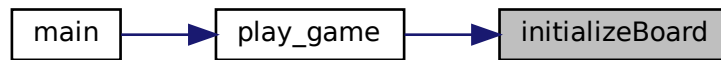
311     {
312     int board[MAX_BOARD][MAX_BOARD] = {0};
313     int cardIndexes[MAX_IMAGES]; // 이 배열은 사용 가능한 카드 인덱스를 저장합니다.
314
315     // 각 카드 인덱스를 초기화
316     for (int i = 0; i < MAX_IMAGES; i++) {
317         cardIndexes[i] = i;
318     }
319
320     // 카드 인덱스를 셔플
321     shuffleCards(cardIndexes, MAX_IMAGES);
322
323     // 보드에 카드 배치
324     int cardCount = 0;
325     for (int i = 0; i < boardSize; i++) {
326         for (int j = 0; j < boardSize; j += 2) {
327             // 같은 카드를 2개씩 배치
328             board[i][j] = board[i][j + 1] = cardIndexes[cardCount++];
329             if (cardCount >= boardSize * boardSize / 2) {
330                 // 필요한 카드 수에 도달하면 중단
331                 break;
332             }
333         }
334         if (cardCount >= boardSize * boardSize / 2) {
335             break;
336         }
337     }
338
339     // 카드 출력
340     for (int i = 0; i < boardSize; i++) {
341         for (int j = 0; j < boardSize; j++) {
342             printf("%s ", images[board[i][j]]);
343         }
344         printf("\n");
345     }
346 }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:





이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.16.2.5 main()

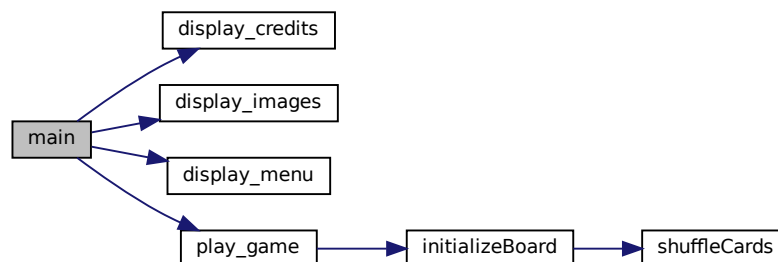
```
int main (
    void )
```

loT3\_ver\_0.3.c 파일의 360 번째 라인에서 정의되었습니다.

```

360 {
361     int choice;
362     srand(time(NULL)); // 랜덤 시드 초기화
363
364     while (1) {
365         choice = display_menu();
366
367         switch(choice) {
368             case 1:
369                 play_game();
370                 break;
371             case 2:
372                 display_images();
373                 break;
374             case 3:
375                 display_credits();
376                 // 크레딧 보여주고 메뉴로 돌아가야 하므로, 여기에 display_menu()를 호출하지 않음.
377                 break;
378             case 4:
379                 printf("\033[2J\033[H");
380                 printf("게임을 종료합니다.\n");
381                 exit(0); // 프로그램 종료
382             default:
383                 printf("잘못된 선택입니다.\n");
384                 sleep(1); // 잠시 대기 후 다시 메뉴를 보여줌
385         }
386     }
387
388     return 0; // 실제로는 여기에 도달하지 않음
389
390
391
392 }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



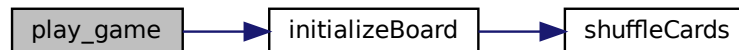
#### 4.16.2.6 play\_game()

```
void play_game ( )
```

IoT3\_ver\_0.3.c 파일의 348 번째 라인에서 정의되었습니다.

```
348 {
349     printf("\033[2J\033[H"); // 화면 지우기 및 커서 이동
350     printf("게임을 시작합니다...\n\n");
351
352     // 게임 보드 초기화 및 카드 배치
353     initializeBoard(MAX_BOARD);
354
355     // 게임 로직 구현 (사용자 입력 받기, 선택된 카드 비교 등)
356     // ...
357 }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.16.2.7 shuffleCards()

```
void shuffleCards (
    int * array,
    int size )
```

IoT3\_ver\_0.3.c 파일의 302 번째 라인에서 정의되었습니다.

```
302 {
303     for (int i = 0; i < size; i++) {
304         int j = rand() % size;
305         int temp = array[i];
306         array[i] = array[j];
307         array[j] = temp;
308     }
309 }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



### 4.16.3 변수 문서화

#### 4.16.3.1 images

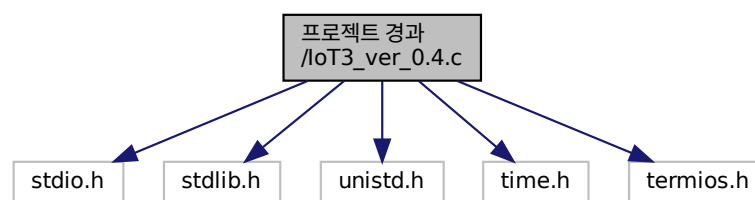
```
const char* images[MAX_IMAGES]
```

loT3\_ver\_0.3.c 파일의 10 번째 라인에서 정의되었습니다.

## 4.17 프로젝트 경과/loT3\_ver\_0.4.c 파일 참조

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <time.h>
#include <termios.h>
```

loT3\_ver\_0.4.c에 대한 include 의존 그래프



### 매크로

- #define MAX\_IMAGES 20
- #define MAX\_BOARD 6

## 함수

- int `display_menu` ()
- void `display_credits` ()
- void `display_images` ()
- void `shuffleCards` (int \*array, int size)
- void `initializeBoard` (int boardSize)
- void `play_game` ()
- int `main` ()

## 변수

- const char \* `images` [`MAX_IMAGES`]

### 4.17.1 매크로 문서화

#### 4.17.1.1 MAX\_BOARD

```
#define MAX_BOARD 6
```

loT3\_ver\_0.4.c 파일의 8 번째 라인에서 정의되었습니다.

#### 4.17.1.2 MAX\_IMAGES

```
#define MAX_IMAGES 20
```

loT3\_ver\_0.4.c 파일의 7 번째 라인에서 정의되었습니다.

### 4.17.2 함수 문서화

## 4.17.2.1 display\_credits()

```
void display_credits ( )
```

IoT3\_ver\_0.4.c 파일의 276 번째 라인에서 정의되었습니다.

```
276     {
277         printf("\033[2J\033[H"); // Clear screen
278
279         printf("\n");
280         printf("\n");
281         printf("                Sola HONG  Data Manager\n");
282         sleep(1);
283         printf("                Subtitle/Translate\n");
284         sleep(1);
285         printf("                Line Producer\n");
286         sleep(1);
287         printf("\n");
288
289         printf("                Haedeun SONG  Production Assistant\n");
290         sleep(1);
291         printf("                Art Design\n");
292         sleep(1);
293         printf("                Tech Engineer\n");
294         sleep(1);
295         printf("\n");
296
297         printf("                Jinwoo LEE   Producer & Director\n");
298         sleep(1);
299         printf("                Motion Graphic Design\n");
300         sleep(1);
301         printf("                Supervisor\n");
302         sleep(1);
303         printf("\n");
304
305         printf("                Hyunjoong YOON Director\n");
306         sleep(1);
307         printf("                Art Design\n");
308         sleep(1);
309         printf("                Coordinating\n");
310         printf("\n");
311         printf("\n");
312         sleep(1);
313         printf("                Present By Team IoT3...Thank you...\n");
314         sleep(1);
315         sleep(600);
316     }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



## 4.17.2.2 display\_images()

```
void display_images ( )
```

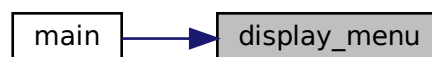
IoT3\_ver\_0.4.c 파일의 318 번째 라인에서 정의되었습니다.

```
318     {
319         printf("\033[2J\033[H"); // 화면 지우기 클리어 함수와 같음
320
321         for (int i = 0; i < MAX_IMAGES; ++i) {
```



[illegible]

이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.17.2.4 initializeBoard()

```
void initializeBoard (
    int boardSize )
```

IoT3\_ver\_0.4.c 파일의 336 번째 라인에서 정의되었습니다.

```

336     {
337         int board[MAX_BOARD][MAX_BOARD] = {0};
338         int cardIndexes[MAX_IMAGES]; // 이 배열은 사용 가능한 카드 인덱스를 저장합니다.
339
340         // 각 카드 인덱스를 초기화
341         for (int i = 0; i < MAX_IMAGES; i++) {
342             cardIndexes[i] = i;
343         }

```

```

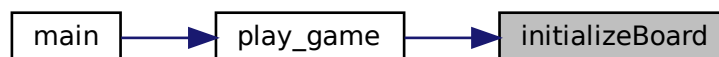
344
345 // 카드 인덱스를 셔플
346 shuffleCards(cardIndexes, MAX_IMAGES);
347
348 // 보드에 카드 배치
349 int cardCount = 0;
350 for (int i = 0; i < boardSize; i++) {
351     for (int j = 0; j < boardSize; j += 2) {
352         // 같은 카드를 2개씩 배치
353         board[i][j] = board[i][j + 1] = cardIndexes[cardCount++];
354         if (cardCount >= boardSize * boardSize / 2) {
355             // 필요한 카드 수에 도달하면 중단
356             break;
357         }
358     }
359     if (cardCount >= boardSize * boardSize / 2) {
360         break;
361     }
362 }
363
364 // 카드 출력
365 for (int i = 0; i < boardSize; i++) {
366     for (int j = 0; j < boardSize; j++) {
367         printf("%s ", images[board[i][j]]);
368     }
369     printf("\n");
370 }
371 }

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.17.2.5 main()

```

int main (
    void )

```

IoT3\_ver\_0.4.c 파일의 385 번째 라인에서 정의되었습니다.

```

385 {
386     int choice;
387     srand(time(NULL)); // 랜덤 시드 초기화

```

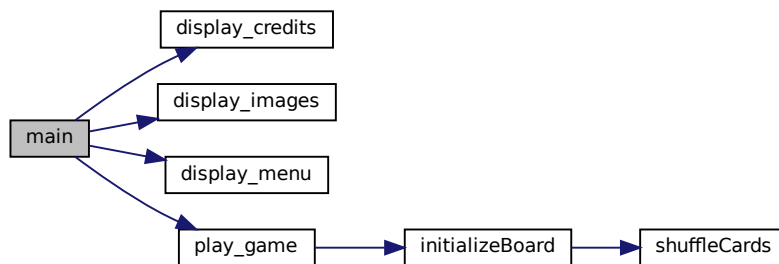


```

388
389     while (1) {
390         choice = display_menu();
391
392         switch(choice) {
393             case 1:
394                 play_game();
395                 break;
396             case 2:
397                 display_images();
398                 break;
399             case 3:
400                 display_credits();
401                 // 크레딧 보여주고 메뉴로 돌아가야 하므로, 여기에 display_menu()를 호출하지 않음.
402                 break;
403             case 4:
404                 printf("\033[2J\033[H");
405                 printf("게임을 종료합니다.\n");
406                 exit(0); // 프로그램 종료
407             default:
408                 printf("잘못된 선택입니다.\n");
409                 sleep(1); // 잠시 대기 후 다시 메뉴를 보여줌
410         }
411     }
412
413     return 0; // 실제로는 여기에 도달하지 않음
414
415
416
417 }

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



#### 4.17.2.6 play\_game()

```
void play_game ( )
```

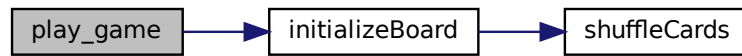
loT3\_ver\_0.4.c 파일의 373 번째 라인에서 정의되었습니다.

```

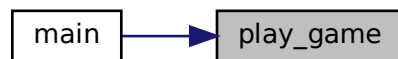
373     {
374         printf("\033[2J\033[H"); // 화면 지우기 및 커서 이동
375         printf("게임을 시작합니다...\n\n");
376
377         // 게임 보드 초기화 및 카드 배치
378         initializeBoard(MAX_BOARD);
379
380         // 게임 로직 구현 (사용자 입력 받기, 선택된 카드 비교 등)
381         // ...
382     }

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.17.2.7 shuffleCards()

```

void shuffleCards (
    int * array,
    int size )
  
```

loT3\_ver\_0.4.c 파일의 327 번째 라인에서 정의되었습니다.

```

327 {
328     for (int i = 0; i < size; i++) {
329         int j = rand() % size;
330         int temp = array[i];
331         array[i] = array[j];
332         array[j] = temp;
333     }
334 }
  
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.17.3 변수 문서화

#### 4.17.3.1 images

```
const char* images[MAX_IMAGES]
```

초기값:

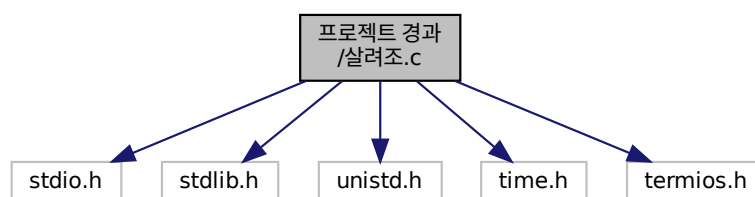
$$= \{$$
$$\begin{array}{l}
( " | \text{-----} | \backslash n " ) , \\
( " | \hspace{10em} | \backslash n " ) , \\
( " | \hspace{4em} ( \backslash \backslash \_ / ) \hspace{4em} | \backslash n " ) , \\
( " | \hspace{4em} ( > ' . ' < ) \hspace{4em} | \backslash n " ) , \\
( " | \hspace{4em} ( \backslash " ) \_ ( \backslash " ) \hspace{4em} | \backslash n " ) , \\
( " | \hspace{10em} | \backslash n " ) , \\
( " | \text{-----} | \backslash n " ) , \\
\\
( " | \text{-----} | \backslash n " ) , \\
( " | \hspace{10em} | \backslash n " ) , \\
( " | \hspace{4em} ( \backslash \backslash \_ / ) \hspace{4em} | \backslash n " ) , \\
( " | \hspace{4em} ( > ' . ' < ) \hspace{4em} | \backslash n " ) , \\
( " | \hspace{4em} ( \backslash " ) \_ ( \backslash " ) \hspace{4em} | \backslash n " ) , \\
( " | \hspace{10em} | \backslash n " ) , \\
( " | \text{-----} | \backslash n " ) , \\
\\
\}
\end{array}$$

IoT3\_ver\_0.4.c 파일의 177 번째 라인에서 정의되었습니다.

#### 4.18 프로젝트 경과/살려조.c 파일 참조

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <time.h>
#include <termios.h>
```

살려조.c에 대한 include 의존 그래프



매크로

- #define MAX\_IMAGES 20
- #define MAX\_BOARD 6

## 함수

- int `display_menu` ()
- void `display_credits` ()
- void `display_images` ()
- void `shuffleCards` (int \*array, int size)
- void `initializeBoard` (int boardSize)
- void `play_game` ()
- int `main` ()

## 변수

- const char \* `images` [`MAX_IMAGES`]

### 4.18.1 매크로 문서화

#### 4.18.1.1 MAX\_BOARD

```
#define MAX_BOARD 6
```

살려조.c 파일의 8 번째 라인에서 정의되었습니다.

#### 4.18.1.2 MAX\_IMAGES

```
#define MAX_IMAGES 20
```

살려조.c 파일의 7 번째 라인에서 정의되었습니다.

### 4.18.2 함수 문서화

## 4.18.2.1 display\_credits()

```
void display_credits ( )
```

살려조.c 파일의 190 번째 라인에서 정의되었습니다.

```
190     {
191         printf("\033[2J\033[H"); // 화면 지우기
192
193         printf("==\t윤현중\t\t==\n\n");
194         sleep(1);
195
196         printf("==\t홍술아\t\t==\n\n");
197         sleep(1);
198
199         printf("==\t이진우\t\t==\n\n");
200         sleep(1);
201
202         printf("==\t송해든\t\t==\n\n");
203         sleep(3);
204     }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



## 4.18.2.2 display\_images()

```
void display_images ( )
```

살려조.c 파일의 206 번째 라인에서 정의되었습니다.

```
206     {
207         printf("\033[2J\033[H"); // 화면 지우기 클리어 함수와 같음
208
209         for (int i = 0; i < MAX_IMAGES; ++i) {
210             printf("%s\n", images[i]); // 이미지 출력
211         }
212         sleep(10); // 이미지를 잠시 표시한 후 메인 메뉴로 복귀
213     }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



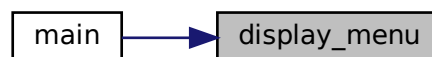
#### 4.18.2.3 display\_menu()

```
int display_menu ( )
```

살려조.c 파일의 172 번째 라인에서 정의되었습니다.

```
172     {
173         int menu;
174
175         system("clear");
176
177         printf("같은 그림 찾기 게임\n");
178         printf("=====\n");
179         printf("    게임 메뉴\n");
180         printf("=====\n");
181         printf("1. 게임 시작\n");
182         printf("2. 사진 체크\n");
183         printf("3. 크레딧\n");
184         printf("4. 게임 종료\n");
185
186         scanf("%d", &menu);
187         return menu;
188     }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.18.2.4 initializeBoard()

```
void initializeBoard (
    int boardSize )
```

살려조.c 파일의 224 번째 라인에서 정의되었습니다.

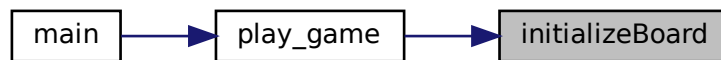
```
224     {
225         int board[MAX_BOARD][MAX_BOARD] = {0};
226         int cardIndexes[MAX_IMAGES / 2]; // 보드 크기의 절반만큼만 필요
227         int placedCards = 0; // 배치된 카드 수
228
229         // 각 카드 인덱스를 초기화
230         for (int i = 0; i < MAX_IMAGES / 2; i++) {
231             cardIndexes[i] = i;
232         }
233
234         // 보드 초기화
235         for (int i = 0; i < boardSize; i++) {
236             for (int j = 0; j < boardSize; j++) {
237                 board[i][j] = -1; // 초기값 설정
238             }
239         }
240
241         while (placedCards < boardSize * boardSize) {
242             for (int i = 0; i < MAX_IMAGES / 2; i++) {
243                 for (int repeat = 0; repeat < 2; repeat++) { // 각 카드를 두 번 배치
244                     int x, y;
245                     do {
246                         x = rand() % boardSize;
247                         y = rand() % boardSize;
248                     } while (board[x][y] != -1); // 빈 위치를 찾을 때까지 반복
249                 }
```

```

250             board[x][y] = cardIndexes[i];
251             placedCards++;
252         }
253     }
254 }
255
256 // 카드 출력 (디버깅 목적)
257 for (int i = 0; i < boardSize; i++) {
258     for (int j = 0; j < boardSize; j++) {
259         printf("%d ", board[i][j]);
260     }
261     printf("\n");
262 }
263 }

```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.18.2.5 main()

```

int main (
    void )

```

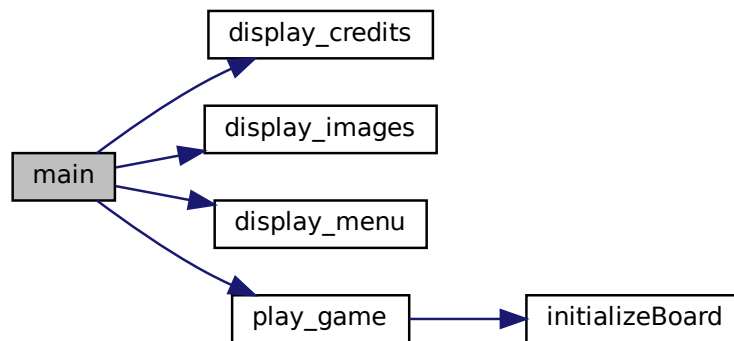
살려조.c 파일의 280 번째 라인에서 정의되었습니다.

```

280     {
281         int choice;
282         srand(time(NULL)); // 랜덤 시드 초기화
283
284         while (1) {
285             choice = display_menu();
286
287             switch(choice) {
288                 case 1:
289                     play_game();
290                     break;
291                 case 2:
292                     display_images();
293                     break;
294                 case 3:
295                     display_credits();
296                     // 크레딧 보여주고 메뉴로 돌아가야 하므로, 여기에 display_menu()를 호출하지 않음
297                     break;
298                 case 4:
299                     printf("\033[2J\033[H");
300                     printf("게임을 종료합니다.\n");
301                     exit(0); // 프로그램 종료
302                 default:
303                     printf("잘못된 선택입니다.\n");
304                     sleep(1); // 잠시 대기 후 다시 메뉴를 보여줌
305             }
306         }
307
308         return 0; // 실제로는 여기에 도달하지 않음
309     }

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



#### 4.18.2.6 play\_game()

```
void play_game ( )
```

살려조.c 파일의 266 번째 라인에서 정의되었습니다.

```

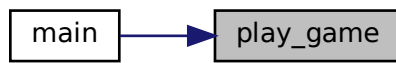
266 {
267     printf("\033[2J\033[H"); // 화면 지우기
268     printf("게임 시작!\n");
269
270
271     int boardSize = 6;
272     initializeBoard(boardSize);
273
274     sleep(3);
275     // 추가적인 게임 로직 구현이 필요
276     // 사용자 입력을 받아 카드를 선택하고, 매칭이 되는지 확인하는 로직
277 }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:





이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.18.2.7 shuffleCards()

```
void shuffleCards (
    int * array,
    int size )
```

살려조.c 파일의 215 번째 라인에서 정의되었습니다.

```
215 {
216     for (int i = 0; i < size; i++) {
217         int j = rand() % size;
218         int temp = array[i];
219         array[i] = array[j];
220         array[j] = temp;
221     }
222 }
```

### 4.18.3 변수 문서화

#### 4.18.3.1 images

```
const char* images[MAX_IMAGES]
```

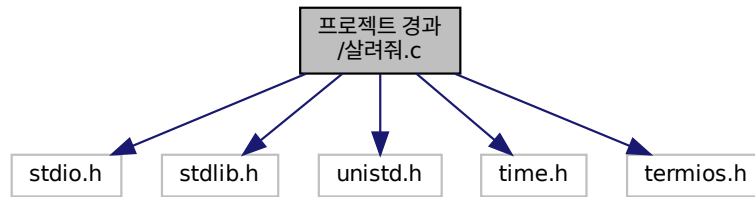
살려조.c 파일의 10 번째 라인에서 정의되었습니다.

## 4.19 프로젝트 경과/살려줘.c 파일 참조

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <time.h>
```

```
#include <termios.h>
```

살려줘.c에 대한 include 의존 그래프



## 매크로

- `#define MAX_IMAGES 20`
- `#define MAX_BOARD 6`

## 함수

- `int display_menu ()`
- `void display_credits ()`
- `void display_images ()`
- `void shuffleCards (int *array, int size)`
- `void initializeBoard (int boardSize)`
- `void play_game ()`
- `int main ()`

## 변수

- `const char * images [MAX_IMAGES]`

### 4.19.1 매크로 문서화

#### 4.19.1.1 MAX\_BOARD

```
#define MAX_BOARD 6
```

살려줘.c 파일의 8 번째 라인에서 정의되었습니다.

### 4.19.1.2 MAX\_IMAGES

```
#define MAX_IMAGES 20
```

살려줘.c 파일의 7 번째 라인에서 정의되었습니다.

## 4.19.2 함수 문서화

### 4.19.2.1 display\_credits()

```
void display_credits ( )
```

살려줘.c 파일의 202 번째 라인에서 정의되었습니다.

```
202 {
203     printf("\033[2J\033[H"); // 화면 지우기
204
205     printf("==\t윤현중\t\t==\n\n");
206     sleep(1);
207
208     printf("==\t홍솔아\t\t==\n\n");
209     sleep(1);
210
211     printf("==\t이진우\t\t==\n\n");
212     sleep(1);
213
214     printf("==\t송해든\t\t==\n\n");
215     sleep(3);
216 }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



### 4.19.2.2 display\_images()

```
void display_images ( )
```

살려줘.c 파일의 218 번째 라인에서 정의되었습니다.

```
218 {
219     printf("\033[2J\033[H"); // 화면 지우기 클리어 함수와 같음
220
221     for (int i = 0; i < MAX_IMAGES; ++i) {
222         printf("%s\n", images[i]); // 이미지 출력
223     }
224     sleep(10); // 이미지를 잠시 표시한 후 메인 메뉴로 복귀
225 }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.19.2.3 display\_menu()

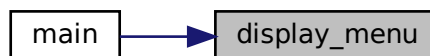
```
int display_menu ( )
```

살려줘.c 파일의 184 번째 라인에서 정의되었습니다.

```

184     {
185         int menu;
186
187         system("clear");
188
189         printf("같은 그림 찾기 게임\n");
190         printf("===== \n");
191         printf("    게임 메뉴\n");
192         printf("===== \n");
193         printf("1. 게임 시작\n");
194         printf("2. 사진 체크\n");
195         printf("3. 크레딧\n");
196         printf("4. 게임 종료\n");
197
198         scanf("%d", &menu);
199         return menu;
200     }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.19.2.4 initializeBoard()

```
void initializeBoard (
    int boardSize )
```

살려줘.c 파일의 236 번째 라인에서 정의되었습니다.

```

236     {
237         int board[MAX_BOARD][MAX_BOARD] = {0};
```

```

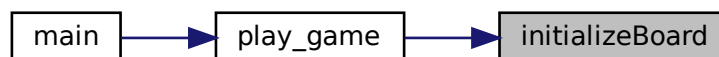
238     int cardIndexes[MAX_IMAGES]; // 이 배열은 사용 가능한 카드 인덱스를 저장
239
240     // 각 카드 인덱스를 초기화
241     for (int i = 0; i < MAX_IMAGES; i++) {
242         cardIndexes[i] = i;
243     }
244
245     // 카드 인덱스를 셔플
246     shuffleCards(cardIndexes, MAX_IMAGES);
247
248     // 보드에 카드 배치
249     int cardCount = 0;
250     for (int i = 0; i < boardSize; i++) {
251         for (int j = 0; j < boardSize; j += 2) {
252             // 같은 카드를 2개씩 배치
253             board[i][j] = board[i][j + 1] = cardIndexes[cardCount++];
254             if (cardCount >= boardSize * boardSize / 2) {
255                 // 필요한 카드 수에 도달하면 중단
256                 break;
257             }
258         }
259         if (cardCount >= boardSize * boardSize / 2) {
260             break;
261         }
262     }
263
264     // 카드 출력
265     for (int i = 0; i < boardSize; i++) {
266         for (int j = 0; j < boardSize; j++) {
267             printf("%s ", images[board[i][j]]);
268         }
269         printf("\n");
270     }
271 }

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.19.2.5 main()

```

int main (
    void )

```

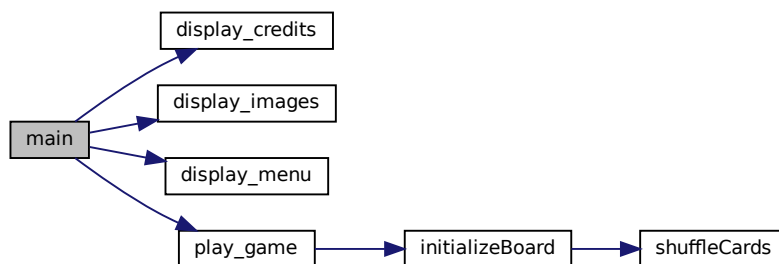
살려줘.c 파일의 287 번째 라인에서 정의되었습니다.

```

287 {
288     int choice;
289     srand(time(NULL)); // 랜덤 시드 초기화
290
291     while (1) {
292         choice = display_menu();
293
294         switch(choice) {
295             case 1:
296                 play_game();
297                 break;
298             case 2:
299                 display_images();
300                 break;
301             case 3:
302                 display_credits();
303                 // 크레딧 보여주고 메뉴로 돌아가야 하므로, 여기에 display_menu()를 호출하지 않음.
304                 break;
305             case 4:
306                 printf("\033[2J\033[H");
307                 printf("게임을 종료합니다.\n");
308                 exit(0); // 프로그램 종료
309             default:
310                 printf("잘못된 선택입니다.\n");
311                 sleep(1); // 잠시 대기 후 다시 메뉴를 보여줌
312         }
313     }
314
315     return 0; // 실제로는 여기에 도달하지 않음
316 }

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



#### 4.19.2.6 play\_game()

```
void play_game ( )
```

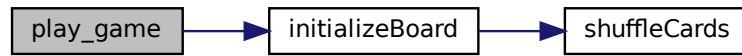
살려줘.c 파일의 273 번째 라인에서 정의되었습니다.

```

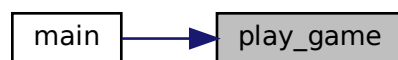
273 {
274     printf("\033[2J\033[H"); // 화면 지우기
275     printf("게임 시작!\n");
276
277
278     int boardSize = 6;
279     initializeBoard(boardSize);
280
281     sleep(3);
282     // 추가적인 게임 로직 구현이 필요합니다.
283     // 예: 사용자 입력을 받아 카드를 선택하고, 매칭이 되는지 확인하는 로직 등
284 }

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



#### 4.19.2.7 shuffleCards()

```

void shuffleCards (
    int * array,
    int size )
  
```

살려줘.c 파일의 227 번째 라인에서 정의되었습니다.

```

227 {
228     for (int i = 0; i < size; i++) {
229         int j = rand() % size;
230         int temp = array[i];
231         array[i] = array[j];
232         array[j] = temp;
233     }
234 }
  
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



### 4.19.3 변수 문서화

#### 4.19.3.1 images

```
const char* images[MAX_IMAGES]
```

살려줘.c 파일의 10 번째 라인에서 정의되었습니다.





# Index

almost2 (origin)/ending.c, [7](#)  
almost2 (origin)/game.c, [8](#)  
almost2 (origin)/game.h, [14](#)  
almost2 (origin)/helpme.c, [21](#)  
almost2 (origin)/main.c, [41](#)  
almost2 (origin)/utility.c, [43](#)  
almost2 (origin)/utility.h, [46](#)

board  
  game.c, [13](#)  
  game.h, [20](#)  
  helpme.c, [31](#), [40](#)  
BOARD\_AREA  
  game.h, [15](#)  
  helpme.c, [22](#), [33](#)  
BOARD\_SIZE  
  game.h, [15](#)  
  helpme.c, [22](#), [33](#)

checkMatch  
  game.c, [9](#)  
  game.h, [16](#)  
  helpme.c, [23](#), [34](#)

disableRawMode  
  helpme.c, [23](#), [34](#)  
  utility.c, [44](#)  
  utility.h, [46](#)

display\_credits  
  helpme.c, [24](#), [35](#)  
  loT3\_ver0.2.c, [57](#)  
  loT3\_ver\_0.1.c, [64](#)  
  loT3\_ver\_0.2.c, [70](#)  
  loT3\_ver\_0.3.c, [77](#)  
  loT3\_ver\_0.4.c, [84](#)  
  loT3\_ver\_0.5.c, [50](#)  
  utility.c, [44](#)  
  utility.h, [46](#)  
  살려조.c, [92](#)  
  살려줘.c, [99](#)

display\_images  
  helpme.c, [24](#), [35](#)  
  loT3\_ver0.2.c, [57](#)  
  loT3\_ver\_0.1.c, [64](#)  
  loT3\_ver\_0.2.c, [70](#)  
  loT3\_ver\_0.3.c, [78](#)  
  loT3\_ver\_0.4.c, [85](#)  
  loT3\_ver\_0.5.c, [51](#)  
  utility.c, [44](#)  
  utility.h, [47](#)

  살려조.c, [93](#)  
  살려줘.c, [99](#)  
display\_menu  
  helpme.c, [25](#), [35](#)  
  loT3\_ver0.2.c, [58](#)  
  loT3\_ver\_0.1.c, [65](#)  
  loT3\_ver\_0.2.c, [71](#)  
  loT3\_ver\_0.3.c, [78](#)  
  loT3\_ver\_0.4.c, [86](#)  
  loT3\_ver\_0.5.c, [51](#)  
  main.c, [41](#)  
  살려조.c, [93](#)  
  살려줘.c, [100](#)

ed.c, [48](#)  
  main, [49](#)  
enableRawMode  
  helpme.c, [25](#), [35](#)  
  utility.c, [45](#)  
  utility.h, [47](#)

ending.c  
  main, [7](#)

game.c  
  board, [13](#)  
  checkMatch, [9](#)  
  images, [13](#)  
  initializeBoard, [9](#)  
  play\_game, [10](#)  
  printBoard, [11](#)  
  processInput, [12](#)  
  revealedCards, [14](#)  
  score, [14](#)  
  shuffleCards, [13](#)

game.h  
  board, [20](#)  
  BOARD\_AREA, [15](#)  
  BOARD\_SIZE, [15](#)  
  checkMatch, [16](#)  
  images, [21](#)  
  IMG\_HEIGHT, [15](#)  
  IMG\_WIDTH, [15](#)  
  initializeBoard, [16](#)  
  MAX\_IMAGES, [16](#)  
  play\_game, [17](#)  
  printBoard, [18](#)  
  processInput, [19](#)  
  revealedCards, [21](#)  
  score, [21](#)  
  shuffleCards, [20](#)

- helpme.c, 32
  - board, 31, 40
  - BOARD\_AREA, 22, 33
  - BOARD\_SIZE, 22, 33
  - checkMatch, 23, 34
  - disableRawMode, 23, 34
  - display\_credits, 24, 35
  - display\_images, 24, 35
  - display\_menu, 25, 35
  - enableRawMode, 25, 35
  - images, 31, 40
  - IMG\_HEIGHT, 22, 33
  - IMG\_WIDTH, 23, 34
  - initializeBoard, 26, 36
  - main, 27, 36
  - MAX\_IMAGES, 23, 34
  - orig\_termios, 32, 40
  - play\_game, 28, 37
  - printBoard, 29, 38
  - processInput, 30, 39
  - revealedCards, 32, 40
  - score, 32, 40
  - shuffleCards, 31, 39
- images
  - game.c, 13
  - game.h, 21
  - helpme.c, 31, 40
  - IoT3\_ver0.2.c, 62
  - IoT3\_ver\_0.1.c, 69
  - IoT3\_ver\_0.2.c, 75
  - IoT3\_ver\_0.3.c, 83
  - IoT3\_ver\_0.4.c, 90
  - IoT3\_ver\_0.5.c, 55
  - 살려조.c, 97
  - 살려줘.c, 103
- IMG\_HEIGHT
  - game.h, 15
  - helpme.c, 22, 33
- IMG\_WIDTH
  - game.h, 15
  - helpme.c, 23, 34
- initializeBoard
  - game.c, 9
  - game.h, 16
  - helpme.c, 26, 36
  - IoT3\_ver0.2.c, 59
  - IoT3\_ver\_0.1.c, 66
  - IoT3\_ver\_0.2.c, 72
  - IoT3\_ver\_0.3.c, 80
  - IoT3\_ver\_0.4.c, 87
  - IoT3\_ver\_0.5.c, 53
  - 살려조.c, 94
  - 살려줘.c, 100
- IoT3\_ver0.2.c
  - display\_credits, 57
  - display\_images, 57
  - display\_menu, 58
  - images, 62
- initializeBoard, 59
  - main, 60
  - MAX\_BOARD, 57
  - MAX\_IMAGES, 57
  - play\_game, 61
  - shuffleCards, 62
- IoT3\_ver\_0.1.c
  - display\_credits, 64
  - display\_images, 64
  - display\_menu, 65
  - images, 69
  - initializeBoard, 66
  - main, 66
  - MAX\_BOARD, 63
  - MAX\_IMAGES, 64
  - play\_game, 67
  - shuffleCards, 68
- IoT3\_ver\_0.2.c
  - display\_credits, 70
  - display\_images, 70
  - display\_menu, 71
  - images, 75
  - initializeBoard, 72
  - main, 73
  - MAX\_BOARD, 70
  - MAX\_IMAGES, 70
  - play\_game, 74
  - shuffleCards, 75
- IoT3\_ver\_0.3.c
  - display\_credits, 77
  - display\_images, 78
  - display\_menu, 78
  - images, 83
  - initializeBoard, 80
  - main, 81
  - MAX\_BOARD, 76
  - MAX\_IMAGES, 77
  - play\_game, 82
  - shuffleCards, 82
- IoT3\_ver\_0.4.c
  - display\_credits, 84
  - display\_images, 85
  - display\_menu, 86
  - images, 90
  - initializeBoard, 87
  - main, 88
  - MAX\_BOARD, 84
  - MAX\_IMAGES, 84
  - play\_game, 89
  - shuffleCards, 90
- IoT3\_ver\_0.5.c, 49
  - display\_credits, 50
  - display\_images, 51
  - display\_menu, 51
  - images, 55
  - initializeBoard, 53
  - main, 54
  - MAX\_BOARD, 50

- MAX\_IMAGES, 50
- play\_game, 54
- shuffleCards, 55
- main
  - ed.c, 49
  - ending.c, 7
  - helpme.c, 27, 36
  - IoT3\_ver0.2.c, 60
  - IoT3\_ver\_0.1.c, 66
  - IoT3\_ver\_0.2.c, 73
  - IoT3\_ver\_0.3.c, 81
  - IoT3\_ver\_0.4.c, 88
  - IoT3\_ver\_0.5.c, 54
  - main.c, 42
  - 살려조.c, 95
  - 살려줘.c, 101
- main.c
  - display\_menu, 41
  - main, 42
- MAX\_BOARD
  - IoT3\_ver0.2.c, 57
  - IoT3\_ver\_0.1.c, 63
  - IoT3\_ver\_0.2.c, 70
  - IoT3\_ver\_0.3.c, 76
  - IoT3\_ver\_0.4.c, 84
  - IoT3\_ver\_0.5.c, 50
  - 살려조.c, 92
  - 살려줘.c, 98
- MAX\_IMAGES
  - game.h, 16
  - helpme.c, 23, 34
  - IoT3\_ver0.2.c, 57
  - IoT3\_ver\_0.1.c, 64
  - IoT3\_ver\_0.2.c, 70
  - IoT3\_ver\_0.3.c, 77
  - IoT3\_ver\_0.4.c, 84
  - IoT3\_ver\_0.5.c, 50
  - 살려조.c, 92
  - 살려줘.c, 98
- orig\_termios
  - helpme.c, 32, 40
  - utility.c, 45
- play\_game
  - game.c, 10
  - game.h, 17
  - helpme.c, 28, 37
  - IoT3\_ver0.2.c, 61
  - IoT3\_ver\_0.1.c, 67
  - IoT3\_ver\_0.2.c, 74
  - IoT3\_ver\_0.3.c, 82
  - IoT3\_ver\_0.4.c, 89
  - IoT3\_ver\_0.5.c, 54
  - 살려조.c, 96
  - 살려줘.c, 102
- printBoard
  - game.c, 11
- game.h, 18
- helpme.c, 29, 38
- processInput
  - game.c, 12
  - game.h, 19
  - helpme.c, 30, 39
- README.md, 56
- revealedCards
  - game.c, 14
  - game.h, 21
  - helpme.c, 32, 40
- score
  - game.c, 14
  - game.h, 21
  - helpme.c, 32, 40
- shuffleCards
  - game.c, 13
  - game.h, 20
  - helpme.c, 31, 39
  - IoT3\_ver0.2.c, 62
  - IoT3\_ver\_0.1.c, 68
  - IoT3\_ver\_0.2.c, 75
  - IoT3\_ver\_0.3.c, 82
  - IoT3\_ver\_0.4.c, 90
  - IoT3\_ver\_0.5.c, 55
  - 살려조.c, 97
  - 살려줘.c, 103
- utility.c
  - disableRawMode, 44
  - display\_credits, 44
  - display\_images, 44
  - enableRawMode, 45
  - orig\_termios, 45
- utility.h
  - disableRawMode, 46
  - display\_credits, 46
  - display\_images, 47
  - enableRawMode, 47
- 살려조.c
  - display\_credits, 92
  - display\_images, 93
  - display\_menu, 93
  - images, 97
  - initializeBoard, 94
  - main, 95
  - MAX\_BOARD, 92
  - MAX\_IMAGES, 92
  - play\_game, 96
  - shuffleCards, 97
- 살려줘.c
  - display\_credits, 99
  - display\_images, 99
  - display\_menu, 100
  - images, 103
  - initializeBoard, 100

main, [101](#)  
MAX\_BOARD, [98](#)  
MAX\_IMAGES, [98](#)  
play\_game, [102](#)  
shuffleCards, [103](#)

프로젝트 경과/loT3\_ver0.2.c, [56](#)  
프로젝트 경과/loT3\_ver\_0.1.c, [63](#)  
프로젝트 경과/loT3\_ver\_0.2.c, [69](#)  
프로젝트 경과/loT3\_ver\_0.3.c, [76](#)  
프로젝트 경과/loT3\_ver\_0.4.c, [83](#)  
프로젝트 경과/README.md, [56](#)  
프로젝트 경과/살려조.c, [91](#)  
프로젝트 경과/살려취.c, [97](#)