

snake gmae

0.0.1

다음에 의해 생성됨 : Doxygen 1.9.1

1 KIB_c_game_project_snake	1
2 파일 색인	3
2.1 파일 목록	3
3 파일 문서화	5
3.1 include/display_snake.h 파일 참조	5
3.1.1 함수 문서화	5
3.1.1.1 display_snake()	6
3.2 README.md 파일 참조	6
3.3 src/display_menu.c 파일 참조	6
3.3.1 함수 문서화	7
3.3.1.1 display_menu()	7
3.4 src/display_snake.c 파일 참조	8
3.4.1 매크로 문서화	9
3.4.1.1 height	9
3.4.1.2 width	9
3.4.2 함수 문서화	9
3.4.2.1 display_snake()	9
3.4.3 변수 문서화	10
3.4.3.1 baitX	10
3.4.3.2 baitY	10
3.4.3.3 count	10
3.4.3.4 entireTail	10
3.4.3.5 point	11
3.4.3.6 snakeX	11
3.4.3.7 snakeY	11
3.4.3.8 x	11
3.4.3.9 y	11
3.5 src/getch.c 파일 참조	12
3.5.1 매크로 문서화	12
3.5.1.1 CCHAR	12
3.5.1.2 CTIME	12
3.5.1.3 FLAG	13
3.5.2 함수 문서화	13
3.5.2.1 getch()	13
3.6 src/main.c 파일 참조	14
3.6.1 매크로 문서화	15
3.6.1.1 GAME_END	15
3.6.1.2 GAME_START	15
3.6.1.3 height	15
3.6.1.4 snake_init_length	15
3.6.1.5 width	16

3.6.2 함수 문서화	16
3.6.2.1 display_menu()	16
3.6.2.2 display_snake()	17
3.6.2.3 game_start()	17
3.6.2.4 main()	19
3.6.2.5 reset()	19
3.6.2.6 space()	20
3.6.2.7 update()	21
3.6.3 변수 문서화	22
3.6.3.1 entireTail	22
3.6.3.2 game	22
3.6.3.3 point	23
3.6.3.4 sdir	23
3.6.3.5 snakeX	23
3.6.3.6 snakeY	23
3.6.3.7 x	23
3.6.3.8 y	23
3.7 src/update.c 파일 참조	24
3.7.1 매크로 문서화	25
3.7.1.1 GAME_END	25
3.7.1.2 GAME_START	25
3.7.1.3 height	25
3.7.1.4 width	25
3.7.2 열거형 타입 문서화	25
3.7.2.1 sdir	25
3.7.3 함수 문서화	26
3.7.3.1 collision()	26
3.7.3.2 move_snake()	27
3.7.3.3 record()	27
3.7.3.4 space()	27
3.7.3.5 update()	28
3.7.4 변수 문서화	30
3.7.4.1 baitX	30
3.7.4.2 baitY	30
3.7.4.3 count	30
3.7.4.4 entireTail	30
3.7.4.5 game	30
3.7.4.6 point	31
3.7.4.7 snakeX	31
3.7.4.8 snakeY	31
3.7.4.9 x	31
3.7.4.10 y	31

Chapter 1

KIB_c_game_project_snake

kib!! 최수길 왔다감 두번째 왔다감. 세번째 왔다감.

Chapter 2

파일 색인

2.1 파일 목록

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

include/display_snake.h	5
src/display_menu.c	6
src/display_snake.c	8
src/getch.c	12
src/main.c	14
src/update.c	24

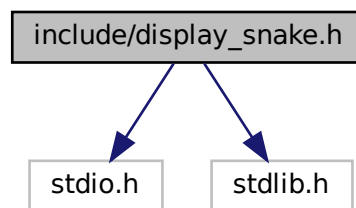
Chapter 3

파일 문서화

3.1 include/display_snake.h 파일 참조

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

display_snake.h에 대한 include 의존 그래프



함수

- void `display_snake` ()

3.1.1 함수 문서화

3.1.1.1 display_snake()

void display_snake ()

display_snake.c 파일의 20 번째 라인에서 정의되었습니다.

```

21 {
22     system("clear");
23     // ceiling print
24     for (int i = 0; i < height; ++i)
25     {
26         for (int j = 0; j < width; ++j)
27         {
28             for (int k = 0; k < entireTail; ++k)
29             {
30                 if (i == snakeX[k] && j == snakeY[k])
31                 {
32                     if (i == 0 || i == height - 1)
33                     {
34                         printf("@");
35                     }
36                     else if (j == 0 || j == width - 1)
37                     {
38                         printf("@");
39                     }
40                     else if (i == x && j == y)
41                     {
42                         printf("X");
43                     }
44                     else if (i == baitX && j == baitY)
45                     {
46                         printf("O");
47                     }
48                     else
49                         printf(" ");
50                 }
51                 printf("\n");
52             }
53             printf("\n\n");
54             printf("Your Point : %d\n", point);
55             printf("\n\n");
56             printf("Timer : %d\n", ++count);
57             printf("\n\n");
58             printf("tail : %d\n", entireTail);
59 }

```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



3.2 README.md 파일 참조

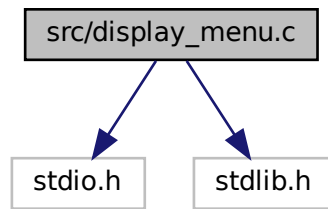
3.3 src/display_menu.c 파일 참조

```

#include <stdio.h>
#include <stdlib.h>

```

display_menu.c에 대한 include 의존 그래프



함수

- int `display_menu` ()

3.3.1 함수 문서화

3.3.1.1 display_menu()

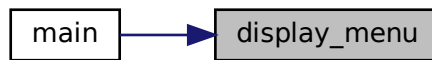
```
int display_menu ( )
```

display_menu.c 파일의 5 번째 라인에서 정의되었습니다.

```

5      {
6
7      int menu;
8
9      system("clear");
10
11
12     printf("\n\n\n");
13
14     printf(" SSSS  N  N  AAA  K  k  EEEEE  GGG  AAA  M  M  EEEEE\n");
15     printf("  S      NN  N  A  A  K  K  E      G      A  A  MM MM E  \n");
16     printf("   SSS  N N N  AAAAA  KK   EEEE  GGGG  AAAAA  M M M  EEEE  \n");
17     printf("    S  N  NN  A  A  K  K  E      G  G  A  A  M  M  E  \n");
18     printf(" SSSS  N  N  A  A  K  K  EEEEE  GGGG  A  A  M  M  EEEEE\n");
19
20
21
22
23     printf("\tWelcome to Snake Game!\n");
24
25     printf("-----\n");
26     printf("\tAre You Ready To Play The Snake Game?\n");
27     printf("-----\n");
28     printf("\t\tGAME MENU\n");
29     printf("-----\n");
30     printf("\t\t1. Game Start\n");
31     printf("\t\t2. Record\n");
32     printf("\t\t3. Quit\n");
33     printf("-----\n");
34
35     printf("Press any key to start...\n");
36
37     scanf("%d", &menu);
38     return menu;
39 }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:

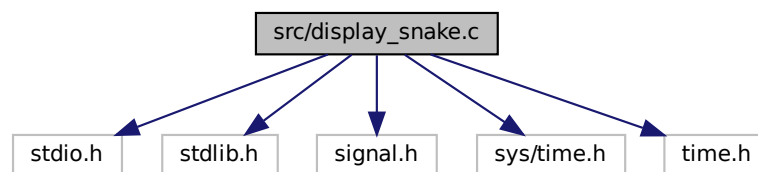


3.4 src/display_snake.c 파일 참조

```

#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <sys/time.h>
#include <time.h>
  
```

display_snake.c에 대한 include 의존 그래프



매크로

- #define width 40
- #define height 40

함수

- void display_snake ()

변수

- int count
- int x
- int y
- int baitX
- int baitY
- int entireTail
- int point
- int snakeX [100]
- int snakeY [100]

3.4.1 매크로 문서화

3.4.1.1 height

```
#define height 40
```

display_snake.c 파일의 8 번째 라인에서 정의되었습니다.

3.4.1.2 width

```
#define width 40
```

display_snake.c 파일의 7 번째 라인에서 정의되었습니다.

3.4.2 함수 문서화

3.4.2.1 display_snake()

```
void display_snake ( )
```

display_snake.c 파일의 20 번째 라인에서 정의되었습니다.

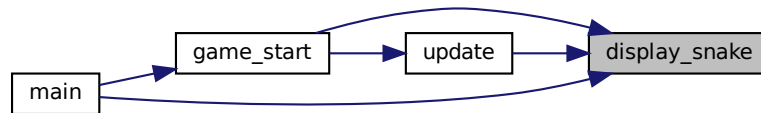
```
21 {
22     system("clear");
23     // ceiling print
24     for (int i = 0; i < height; ++i)
25     {
26         for (int j = 0; j < width; ++j)
27         {
28             for (int k = 0; k < entireTail; ++k)
29             {
30                 if (i == snakeX[k] && j == snakeY[k])
31                 {
32                     if (i == 0 || i == height - 1)
33                     {
34                         printf("@");
35                     }
36                     else if (j == 0 || j == width - 1)
37                     {
38                         printf("@");
39                     }
40                     else if (i == x && j == y)
41                     {
42                         printf("X");
43                     }
44                     else if (i == baitX && j == baitY)
45                     {
46                         printf("O");
47                     }
48                     else
49                         printf(" ");
50                 }
51             }
52             printf("\n");
53             printf("\n\n");
54             printf("Your Point : %d\n", point);
55             printf("\n\n");
56             printf("Timer : %d\n", ++count);
```

```

57     printf("\n\n");
58     printf("tail : %d\n", entireTail);
59 }

```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



3.4.3 변수 문서화

3.4.3.1 baitX

```
int baitX [extern]
```

update.c 파일의 19 번째 라인에서 정의되었습니다.

3.4.3.2 baitY

```
int baitY [extern]
```

update.c 파일의 20 번째 라인에서 정의되었습니다.

3.4.3.3 count

```
int count [extern]
```

update.c 파일의 12 번째 라인에서 정의되었습니다.

3.4.3.4 entireTail

```
int entireTail [extern]
```

main.c 파일의 24 번째 라인에서 정의되었습니다.

3.4.3.5 point

```
int point [extern]
```

main.c 파일의 21 번째 라인에서 정의되었습니다.

3.4.3.6 snakeX

```
int snakeX[100] [extern]
```

update.c 파일의 21 번째 라인에서 정의되었습니다.

3.4.3.7 snakeY

```
int snakeY[100] [extern]
```

update.c 파일의 22 번째 라인에서 정의되었습니다.

3.4.3.8 x

```
int x [extern]
```

main.c 파일의 22 번째 라인에서 정의되었습니다.

3.4.3.9 y

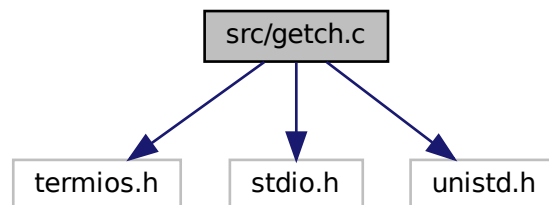
```
int y [extern]
```

main.c 파일의 23 번째 라인에서 정의되었습니다.

3.5 src/getch.c 파일 참조

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

getch.c에 대한 include 의존 그래프



매크로

- #define CCHAR 0
- #define CTIME 1
- #define FLAG TCSAFLUSH

함수

- int getch (void)

3.5.1 매크로 문서화

3.5.1.1 CCHAR

```
#define CCHAR 0
```

getch.c 파일의 6 번째 라인에서 정의되었습니다.

3.5.1.2 CTIME

```
#define CTIME 1
```

getch.c 파일의 10 번째 라인에서 정의되었습니다.

3.5.1.3 FLAG

```
#define FLAG TCSAFLUSH
```

3.5.2 함수 문서화

3.5.2.1 getch()

```
int getch (
    void )
```

getch.c 파일의 13 번째 라인에서 정의되었습니다.

```
14 {
15     char ch;
16     int error;
17     static struct termios Otty, Ntty;
18
19     // fflush(stdout);
20     tcgetattr(0, &Otty);
21     Ntty = Otty;
22     Ntty.c_iflag = 0;
23     Ntty.c_oflag = 0;
24     Ntty.c_lflag &= ~ICANON;
25 #if 1
26     Ntty.c_lflag &= ~ECHO;
27 #else
28     Ntty.c_lflag |= ECHO;
29 #endif
30     Ntty.c_cc[VMIN] = CCHAR;
31     Ntty.c_cc[VTIME] = CTIME;
32
33 #if 1
34 #define FLAG TCSAFLUSH
35 #else
36 #define FLAG TCSANOW
37 #endif
38
39     if (0 == (error = tcsetattr(0, FLAG, &Ntty)))
40     {
41         error = read(0, &ch, 1);
42         error += tcsetattr(0, FLAG, &Otty);
43     }
44
45     return (error == 1 ? (int)ch : -1);
46 }
```

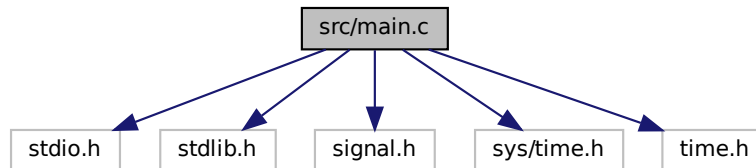
이 함수를 호출하는 함수들에 대한 그래프입니다.:



3.6 src/main.c 파일 참조

```
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <sys/time.h>
#include <time.h>
```

main.c에 대한 include 의존 그래프



매크로

- #define `GAME_START` 0
*printf*함수, *signal*함수, 타이머, *static struct*를 활용하기 위한 헤더파일을 포함시켰습니다. *game*, *point*, 필드의 크기, 처음 시작했을때 꼬리의길이, 꼬리의 변화를 주는 변수, 게임 시작과 끝을 정의하고 미끼의 위치, *snakedirection*(*sdir*), *update*변수, *disdplay_menu*변수, *display_snake*변수를 가져와서 사용했습니다.
- #define `GAME_END` 1
- #define `width` 40
- #define `height` 40
- #define `snake_init_length` 3

함수

- void `space` ()
- int `update` (int signum)
- int `display_menu` ()
- void `display_snake` ()
- int `game_start` ()
game_start 변수를 정의하였습니다. *itimerval* 구조체는 POSIX 시스템에서 타이머 값을 설정하고 제어하는 데 사용됩니다. 이 구조체는 시간 간격을 나타내는 두 개의 필드(*it.value* *it.interval*)을 포함하며 이 구조체를 정적변수(*static struct*)화 시켜서 정의했습니다. 또한 *signal*함수를 사용하여 가상타이머가 끝나면(*SIGVTALRM*) *update*함수를 불러오도록 설정하였습니다. 또한 *setitmer* 함수를 사용하여 타이머를 설정하였고 타이머의 종류(*ITIMER_VIRTUAL*(가상 타이머)), *timer* 구조체의 주소(&*timer*), 이전 타이머의 설정(보통 *NULL*로 표현한다.)으로 구성했습니다.
- int `main` (void)
- void `reset` ()

변수

- int `game`
- int `point` = 0
- int `x` = `width` / 2
- int `y` = `height` / 2
- int `entireTail`
- `sdir`
- int `snakeX` [100]
- int `snakeY` [100]

3.6.1 매크로 문서화

3.6.1.1 GAME_END

```
#define GAME_END 1
```

main.c 파일의 15 번째 라인에서 정의되었습니다.

3.6.1.2 GAME_START

```
#define GAME_START 0
```

printf함수, signal함수, 타이머, static struct를 활용하기 위한 헤더파일을 포함시켰습니다. game,point, 필드의 크기, 처음 시작했을때 꼬리의길이, 꼬리의 변화를 주는 변수, 게임 시작과 끝을 정의하고 미끼의 위치,snakedirection(sdir), update변수, disdplay_menu변수, display_snake변수를 가져와서 사용했습니다.

main.c 파일의 14 번째 라인에서 정의되었습니다.

3.6.1.3 height

```
#define height 40
```

main.c 파일의 17 번째 라인에서 정의되었습니다.

3.6.1.4 snake_init_length

```
#define snake_init_length 3
```

main.c 파일의 18 번째 라인에서 정의되었습니다.

3.6.1.5 width

```
#define width 40
```

main.c 파일의 16 번째 라인에서 정의되었습니다.

3.6.2 함수 문서화

3.6.2.1 display_menu()

```
int display_menu ( )
```

display_menu.c 파일의 5 번째 라인에서 정의되었습니다.

```
5      {
6
7      int menu;
8
9      system("clear");
10
11
12      printf("\n\n\n");
13
14      printf(" SSSS N N AAA K k EEEEE GGG AAA M M EEEEE\n");
15      printf(" S NN N A A K K E G A A MM MM E \n");
16      printf(" SSS N N N AAAAA KK EEEE GGGG AAAAA M M M EEEE \n");
17      printf(" S N NN A A K K E G G A A M M E \n");
18      printf(" SSSS N N A A K K EEEEE GGGG A A M M EEEEE\n");
19
20
21
22
23      printf("\tWelcome to Snake Game!\n");
24
25      printf("-----\n");
26      printf("\tAre You Ready To Play The Snake Game?\n");
27      printf("-----\n");
28      printf("\t\tGAME MENU\n");
29      printf("-----\n");
30      printf("\t\t1. Game Start\n");
31      printf("\t\t2. Record\n");
32      printf("\t\t3. Quit\n");
33      printf("-----\n");
34
35      printf("Press any key to start...\n");
36
37      scanf("%d", &menu);
38      return menu;
39 }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



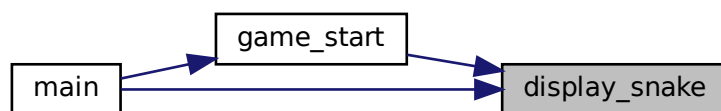
3.6.2.2 display_snake()

```
void display_snake ( )
```

display_snake.c 파일의 20 번째 라인에서 정의되었습니다.

```
21 {
22     system("clear");
23     // ceiling print
24     for (int i = 0; i < height; ++i)
25     {
26         for (int j = 0; j < width; ++j)
27         {
28             for (int k = 0; k < entireTail; ++k)
29             {
30                 if (i == snakeX[k] && j == snakeY[k])
31                 {
32                     if (i == 0 || i == height - 1)
33                     {
34                         printf("@");
35                     }
36                     else if (j == 0 || j == width - 1)
37                     {
38                         printf("@");
39                     }
40                     else if (i == x && j == y)
41                     {
42                         printf("X");
43                     }
44                     else if (i == baitX && j == baitY)
45                     {
46                         printf("O");
47                     }
48                     else
49                         printf(" ");
50                 }
51                 printf("\n");
52             }
53             printf("\n\n");
54             printf("Your Point : %d\n", point);
55             printf("\n\n");
56             printf("Timer : %d\n", ++count);
57             printf("\n\n");
58             printf("tail : %d\n", entireTail);
59 }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



3.6.2.3 game_start()

```
int game_start ( )
```

game_start 변수를 정의하였습니다. itimerval 구조체는 POSIX 시스템에서 타이머 값을 설정하고 제어하는 데 사용됩니다. 이 구조체는 시간 간격을 나타내는 두 개의 필드(it.value it.interval)을 포함하며 이 구조체를 정적변수

(static struct)화 시켜서 정의했습니다. 또한 signal함수를 사용하여 가상타이머가 끝나면(SIGVTALRM) update함수를 불러오도록 설정하였습니다. 또한 setitimer 함수를 사용하여 타이머를 설정하였고 타이머의 종류(ITIMER_VIRTUAL(가상 타이머)), timer 구조체의 주소(&timer), 이전 타이머의 설정(보통 NULL로 표현한다.)으로 구성했습니다.

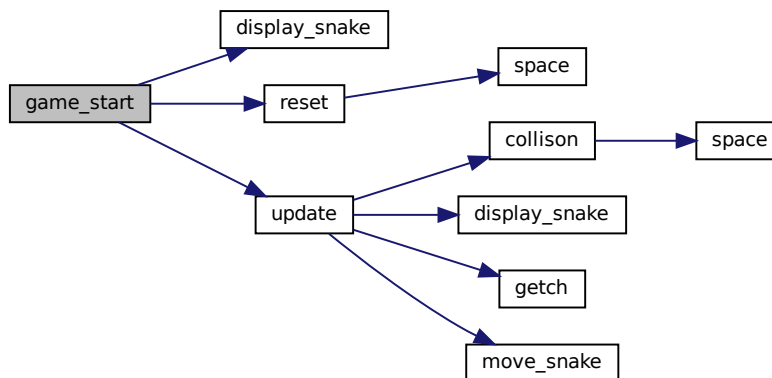
main.c 파일의 45 번째 라인에서 정의되었습니다.

```

46 {
47     // signal 설정
48     static struct itimerval timer;
49     signal(SIGVTALRM, update);
50     timer.it_value.tv_sec = 0;
51     timer.it_value.tv_usec = 16667;
52     timer.it_interval.tv_sec = 0;
53     timer.it_interval.tv_usec = 16667;
54     setitimer(ITIMER_VIRTUAL, &timer, NULL);
55
56     display_snake();
57     int update();
58
59     while (1){
60         if (game == GAME_END){
61             signal(SIGVTALRM, SIG_IGN);
62             //save_result(point);
63             reset();
64             return 1;
65         }
66     }
67     game = GAME_END;
68 }

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



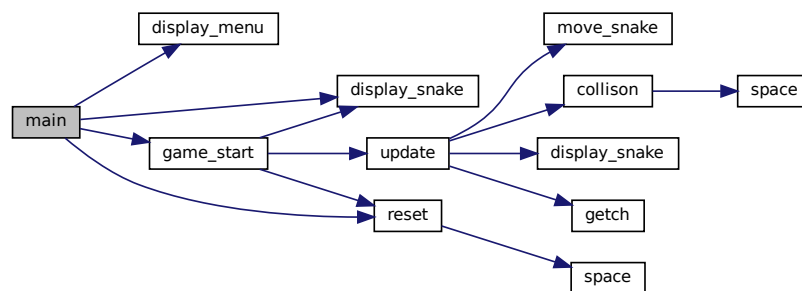
3.6.2.4 main()

```
int main (
    void )
```

main.c 파일의 71 번째 라인에서 정의되었습니다.

```
72 {
73     int menu = 1;
74     menu = display_menu();
75
76     switch (menu)
77     {
78     case 1:
79         game = GAME_START;
80         display_snake();
81         reset();
82         menu = game_start();
83         break;
84     case 2:
85         printf("Record\n");
86         break;
87     case 3:
88         printf("Quit\n");
89         break;
90     return 0;
91     }
92 }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



3.6.2.5 reset()

```
void reset ( )
```

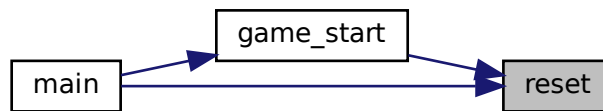
main.c 파일의 95 번째 라인에서 정의되었습니다.

```
96 {
97     // move_snake(STOP);
98     point = 0;
99     x = width / 2;
100     y = height / 2;
101     snakeX[0] = x;
102     snakeY[0] = y;
103     space();
104     // baitX = rand() % width;
105     // baitY = rand() % height;
106 }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



3.6.2.6 space()

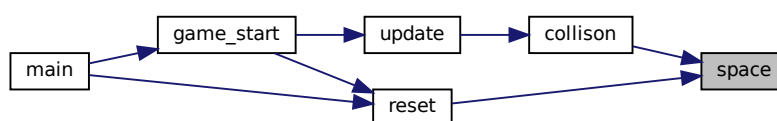
void space ()

update.c 파일의 26 번째 라인에서 정의되었습니다.

```

27 {
28     srand(time(NULL));
29
30     baitX = rand()%height-1;
31     baitY = rand()%width-1;
32
33     if(baitX == 0 || baitX == height)
34     {baitX = rand()%height-1;}
35     else if(baitY == 0 || baitY == width)
36     {baitY = rand()%width-1;}
37
38     if (x == baitX && y == baitY)
39     {
40         baitX = rand()%height-1;
41         baitY = rand()%width-1;
42     }
43 }
44 }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



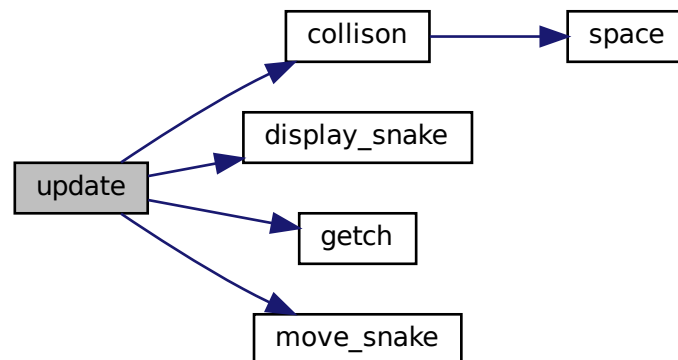
3.6.2.7 update()

```
int update (
    int signum )
```

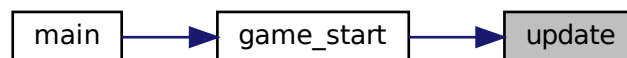
update.c 파일의 93 번째 라인에서 정의되었습니다.

```
94 {
95     int ch;
96     ch = getch();
97
98
99     switch (ch)
100     {
101         case 'i':
102             move_snake(UP);
103             break;
104         case 'j':
105             move_snake(LEFT);
106             break;
107         case 'k':
108             move_snake(DOWN);
109             break;
110         case 'l':
111             move_snake(RIGHT);
112             break;
113         case 's':
114             game = GAME_END;
115             break;
116
117     }
118
119
120
121     ++ count;
122     display_snake();
123     collison();
124     // gameover if snake touch it's own body
125     //for (int i = 1; i < entireTail; ++i)
126     // {
127         //if(snakeX[0]==snakeX[i] && snakeY[0]==snakeY[i])
128         // game = GAME_END;
129     // }
130
131
132     // gameover if snake touch wall
133     if (x == 0){
134         game = GAME_END;
135     }
136     if (y == 0){
137         game = GAME_END;
138     }
139
140     if (x == width-1){
141         game = GAME_END;
142     }
143
144     if (y == height-1){
145         game = GAME_END;
146     }
147
148 }
149
150
151 // snake tail logic
152 int tempX = snakeX[0];
153 int tempY = snakeY[0];
154 int tempX2, tempY2;
155 snakeX[0] = x;
156 snakeY[0] = y;
157 for (int i = 1; i < entireTail; i++) {
158     tempX2 = snakeX[i];
159     tempY2 = snakeY[i];
160     snakeX[i] = tempX;
161     snakeY[i] = tempY;
162     tempX = tempX2;
163     tempY = tempY2;
164 }
165 return 0;
166 }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



3.6.3 변수 문서화

3.6.3.1 entireTail

```
int entireTail
```

main.c 파일의 24 번째 라인에서 정의되었습니다.

3.6.3.2 game

```
int game
```

main.c 파일의 20 번째 라인에서 정의되었습니다.

3.6.3.3 point

```
int point = 0
```

main.c 파일의 21 번째 라인에서 정의되었습니다.

3.6.3.4 sdir

```
sdir [extern]
```

3.6.3.5 snakeX

```
int snakeX[100] [extern]
```

update.c 파일의 21 번째 라인에서 정의되었습니다.

3.6.3.6 snakeY

```
int snakeY[100] [extern]
```

update.c 파일의 22 번째 라인에서 정의되었습니다.

3.6.3.7 x

```
int x = width / 2
```

main.c 파일의 22 번째 라인에서 정의되었습니다.

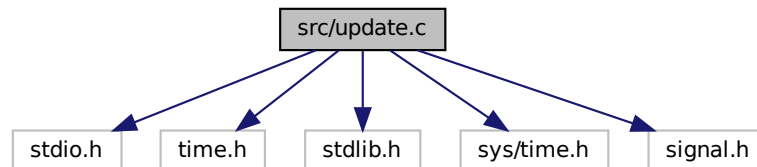
3.6.3.8 y

```
int y = height / 2
```

main.c 파일의 23 번째 라인에서 정의되었습니다.

3.7 src/update.c 파일 참조

```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <sys/time.h>
#include <signal.h>
update.c에 대한 include 의존 그래프
```



매크로

- #define `GAME_START` 0
- #define `GAME_END` 1
- #define `width` 40
- #define `height` 40

열거형 타입

- enum `sdir` {
 `STOP` = 0 , `LEFT` , `RIGHT` , `UP` ,
 `DOWN` }

함수

- int `record` ()
- void `space` ()
- int `move_snake` (int `sdir`)
- void `collision` ()
- int `update` (int signum)

변수

- int `count`
- int `x`
- int `y`
- int `game`
- int `point`
- `entireTail`
- int `baitX`
- int `baitY`
- int `snakeX` [100]
- int `snakeY` [100]

3.7.1 매크로 문서화

3.7.1.1 GAME_END

```
#define GAME_END 1
```

update.c 파일의 8 번째 라인에서 정의되었습니다.

3.7.1.2 GAME_START

```
#define GAME_START 0
```

update.c 파일의 7 번째 라인에서 정의되었습니다.

3.7.1.3 height

```
#define height 40
```

update.c 파일의 10 번째 라인에서 정의되었습니다.

3.7.1.4 width

```
#define width 40
```

update.c 파일의 9 번째 라인에서 정의되었습니다.

3.7.2 열거형 타입 문서화

3.7.2.1 sdir

```
enum sdir
```

열거형 멤버

STOP	
LEFT	
RIGHT	
UP	
DOWN	

다음에 의해 생성됨 : Doxygen

update.c 파일의 46 번째 라인에서 정의되었습니다.

```
47 {
48     STOP = 0,
49     LEFT,
50     RIGHT,
51     UP,
52     DOWN
53 } sdir;
```

3.7.3 함수 문서화

3.7.3.1 collision()

```
void collision ( )
```

update.c 파일의 84 번째 라인에서 정의되었습니다.

```
85 {
86     if (x == baitX && y == baitY)
87     {++point;
88     ++entireTail;
89     space();
90     void update_snaketail();}
91 }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



3.7.3.2 move_snake()

```
int move_snake (
    int sdir )
```

update.c 파일의 55 번째 라인에서 정의되었습니다.

```
56 {
57     int newx, newy;
58
59     newx = x;
60     newy = y;
61
62     switch (sdir){
63         case LEFT:
64             newy--;
65             break;
66         case RIGHT:
67             newy++;
68             break;
69         case DOWN:
70             newx++;
71             break;
72         case UP:
73             newx--;
74             break;
75     }
76
77     x = newx;
78     y = newy;
79
80     return 0;
81 }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



3.7.3.3 record()

```
int record ( )
```

3.7.3.4 space()

```
void space ( )
```

update.c 파일의 26 번째 라인에서 정의되었습니다.

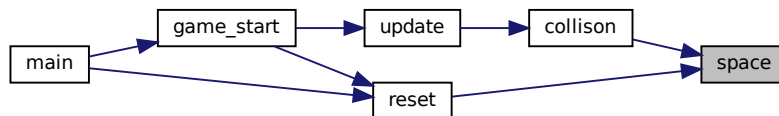
```
27 {
28     srand(time(NULL));
29
30     baitX = rand()%height-1;
31     baitY = rand()%width-1;
32
33     if(baitX == 0 || baitX == height)
34     {baitX = rand()%height-1;}
```

```

35     else if(baitY == 0 || baitY == width)
36     {baitY = rand()%width-1;}
37
38     if (x == baitX && y == baitY)
39     {
40         baitX = rand()%height-1;
41         baitY = rand()%width-1;
42     }
43
44 }

```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



3.7.3.5 update()

```

int update (
    int signum )

```

update.c 파일의 93 번째 라인에서 정의되었습니다.

```

94 {
95     int ch;
96     ch = getch();
97
98
99     switch (ch)
100     {
101         case 'i':
102             move_snake(UP);
103             break;
104         case 'j':
105             move_snake(LEFT);
106             break;
107         case 'k':
108             move_snake(DOWN);
109             break;
110         case 'l':
111             move_snake(RIGHT);
112             break;
113         case 's':
114             game = GAME_END;
115             break;
116     }
117
118
119
120
121     ++ count;
122     display_snake();
123     collison();
124     // gameover if snake touch it's own body
125     //for (int i = 1; i < entireTail; ++i)
126     // {
127         //if (snakeX[0]==snakeX[i] && snakeY[0]==snakeY[i])
128         // game = GAME_END;
129     // }
130
131
132     // gameover if snake touch wall
133     if (x == 0){
134         game = GAME_END;

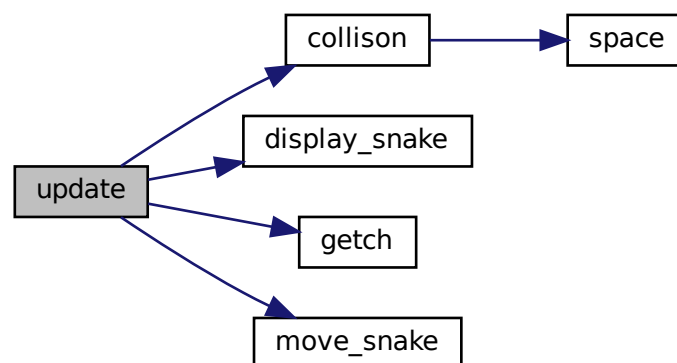
```

```

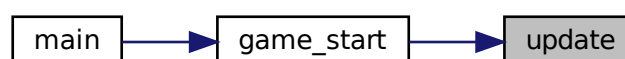
135
136     }
137     if (y == 0){
138         game = GAME_END;
139     }
140
141     if (x == width-1){
142         game = GAME_END;
143     }
144
145     if (y == height-1){
146         game = GAME_END;
147     }
148 }
149
150
151 // snake tail logic
152 int tempX = snakeX[0];
153 int tempY = snakeY[0];
154 int tempX2, tempY2;
155 snakeX[0] = x;
156 snakeY[0] = y;
157 for (int i = 1; i < entireTail; i++) {
158     tempX2 = snakeX[i];
159     tempY2 = snakeY[i];
160     snakeX[i] = tempX;
161     snakeY[i] = tempY;
162     tempX = tempX2;
163     tempY = tempY2;
164 }
165 return 0;
166 }

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



3.7.4 변수 문서화

3.7.4.1 baitX

```
int baitX
```

update.c 파일의 19 번째 라인에서 정의되었습니다.

3.7.4.2 baitY

```
int baitY
```

update.c 파일의 20 번째 라인에서 정의되었습니다.

3.7.4.3 count

```
int count
```

update.c 파일의 12 번째 라인에서 정의되었습니다.

3.7.4.4 entireTail

```
entireTail [extern]
```

main.c 파일의 24 번째 라인에서 정의되었습니다.

3.7.4.5 game

```
int game [extern]
```

main.c 파일의 20 번째 라인에서 정의되었습니다.

3.7.4.6 point

```
int point [extern]
```

main.c 파일의 21 번째 라인에서 정의되었습니다.

3.7.4.7 snakeX

```
int snakeX[100]
```

update.c 파일의 21 번째 라인에서 정의되었습니다.

3.7.4.8 snakeY

```
int snakeY[100]
```

update.c 파일의 22 번째 라인에서 정의되었습니다.

3.7.4.9 x

```
int x [extern]
```

main.c 파일의 22 번째 라인에서 정의되었습니다.

3.7.4.10 y

```
int y [extern]
```

main.c 파일의 23 번째 라인에서 정의되었습니다.

Index

- baitX
 - display_snake.c, 10
 - update.c, 30
- baitY
 - display_snake.c, 10
 - update.c, 30
- CCHAR
 - getch.c, 12
- collison
 - update.c, 26
- count
 - display_snake.c, 10
 - update.c, 30
- CTIME
 - getch.c, 12
- display_menu
 - display_menu.c, 7
 - main.c, 16
- display_menu.c
 - display_menu, 7
- display_snake
 - display_snake.c, 9
 - display_snake.h, 5
 - main.c, 16
- display_snake.c
 - baitX, 10
 - baitY, 10
 - count, 10
 - display_snake, 9
 - entireTail, 10
 - height, 9
 - point, 10
 - snakeX, 11
 - snakeY, 11
 - width, 9
 - x, 11
 - y, 11
- display_snake.h
 - display_snake, 5
- DOWN
 - update.c, 25
- entireTail
 - display_snake.c, 10
 - main.c, 22
 - update.c, 30
- FLAG
 - getch.c, 12
- game
 - main.c, 22
 - update.c, 30
- GAME_END
 - main.c, 15
 - update.c, 25
- GAME_START
 - main.c, 15
 - update.c, 25
- game_start
 - main.c, 17
- getch
 - getch.c, 13
- getch.c
 - CCHAR, 12
 - CTIME, 12
 - FLAG, 12
 - getch, 13
- height
 - display_snake.c, 9
 - main.c, 15
 - update.c, 25
- include/display_snake.h, 5
- LEFT
 - update.c, 25
- main
 - main.c, 18
- main.c
 - display_menu, 16
 - display_snake, 16
 - entireTail, 22
 - game, 22
 - GAME_END, 15
 - GAME_START, 15
 - game_start, 17
 - height, 15
 - main, 18
 - point, 22
 - reset, 19
 - sdir, 23
 - snake_init_length, 15
 - snakeX, 23
 - snakeY, 23
 - space, 20
 - update, 21

- width, 15
- x, 23
- y, 23
- move_snake
 - update.c, 26
- point
 - display_snake.c, 10
 - main.c, 22
 - update.c, 30
- README.md, 6
- record
 - update.c, 27
- reset
 - main.c, 19
- RIGHT
 - update.c, 25
- sdir
 - main.c, 23
 - update.c, 25
- snake_init_length
 - main.c, 15
- snakeX
 - display_snake.c, 11
 - main.c, 23
 - update.c, 31
- snakeY
 - display_snake.c, 11
 - main.c, 23
 - update.c, 31
- space
 - main.c, 20
 - update.c, 27
- src/display_menu.c, 6
- src/display_snake.c, 8
- src/getch.c, 12
- src/main.c, 14
- src/update.c, 24
- STOP
 - update.c, 25
- UP
 - update.c, 25
- update
 - main.c, 21
 - update.c, 28
- update.c
 - baitX, 30
 - baitY, 30
 - collison, 26
 - count, 30
 - DOWN, 25
 - entireTail, 30
 - game, 30
 - GAME_END, 25
 - GAME_START, 25
 - height, 25
 - LEFT, 25
 - move_snake, 26
 - point, 30
 - record, 27
 - RIGHT, 25
 - sdir, 25
 - snakeX, 31
 - snakeY, 31
 - space, 27
 - STOP, 25
 - UP, 25
 - update, 28
 - width, 25
 - x, 31
 - y, 31
- width
 - display_snake.c, 9
 - main.c, 15
 - update.c, 25
- x
 - display_snake.c, 11
 - main.c, 23
 - update.c, 31
- y
 - display_snake.c, 11
 - main.c, 23
 - update.c, 31