

## Lab2022 – Classic Game: Snake

```
#define MINX 21
#define MAXX 100
#define MINY 6
#define MAXY 25

// show map region,
// show snake on screen with direction and length according to snake position,
// and show stone on screen according to stone position
void show(int x, int y, int dir, int len, int sx, int sy)
{
    // Ddaw map region
    // implement something here
    .....
    .....

    // Ddaw snake
    for (int i=0; i<len; i++) {

        switch (dir) {
            case 1: gotoxy(x, (y+i-MINY)%(MAXY-MINY+1)+MINY); break;
            case 2: gotoxy((x-i+(MAXX-MINX+1)-MINX)%(MAXX-MINX+1)+MINX, y); break;
            case 3: gotoxy(x, (y-i+(MAXY-MINY+1)-MINY)%(MAXY-MINY+1)+MINY); break;
            case 4: gotoxy((x+i-MINX)%(MAXX-MINX+1)+MINX, y); break;
        }

        // Ddaw snake head and body
        // implement something here
        .....
        .....
    }

    // Draw stone
    // implement something here
    .....
    .....
}
```

```
// change ahead direction of snake according to key 'w', 'd', 's', 'a'
// 'p' is for exit this game
// the parameter is direction of snake
```

```
void control(int *dir)
```

```
{
```

```
    char key;
```

```
    // int kbhit(void); check if there is a key input,
    // if yes, return nonzero, or return zero
```

```
    if (kbhit())
```

```
        key = getch();
```

```
    // implement something here
```

```
    .....
```

```
    .....
```

```
}
```

```
// change position of snake according to ahead direction of snake
```

```
// the first parameter is direction of ahead
```

```
// the latter two parameters are current position of snake
```

```
// change position to next position in ahead direction
```

```
void move(int dir, int *x, int*y)
```

```
{
```

```
    // implement something here
```

```
    .....
```

```
    .....
```

```
}
```

```
// randomly generate stone
```

```
// the former two parameters are for position of stone
```

```
// the latter parameter is a checking flag for regenerating new stone or not
```

```
void gen_stone(int *x, int*y, int *flag)
```

```
{
```

```
    // implement something here
```

```
    .....
```

```
    .....
```

```
}
```

```
// check if snake eats the stone and add the length of snake
```

```
// the first three parameters are position of snake and length of snake
```

```
// the last three parameters are position of stone and flag for regenerating stone
or not
```

```
void eat(int x, int y, int *len, int sx, int sy, int *sflag)
```

```
{
```

```
    // implement something here
```

```
    .....
```

```
    .....
```

```
}
```

```

int main()
{
    int x, y, dir, len=1; // snake information
    int sx, sy, sflag=0; //stone position

    // generate position and direction of snake head randomly
    // implement something here
    .....
    .....

    while (1) {

        textbackground(BLACK);
        clrscr();
        gen_stone(&sx, &sy, &sflag);
        control(&dir);
        move(dir, &x, &y);
        eat(x,y,&len,sx,sy,&sflag);
        show (x,y,dir,len,sx,sy);
        _sleep(200);
    }

    return 0;
}

```