

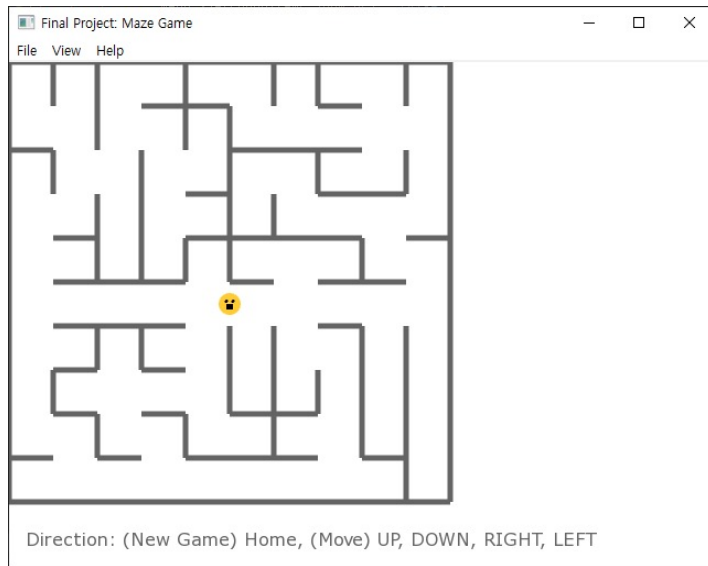
컴실 1 기말 프로젝트: 미로 프로젝트

20150555 남민혁

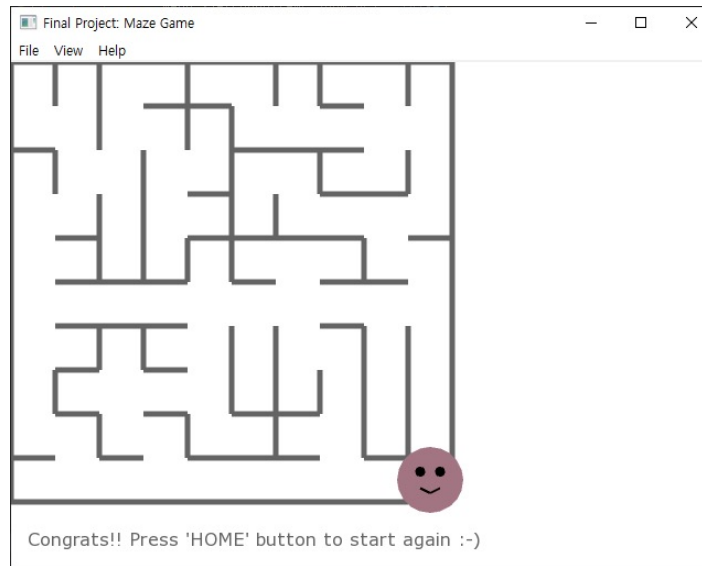
프로젝트 개요

❖ 미로 프로젝트의 확장

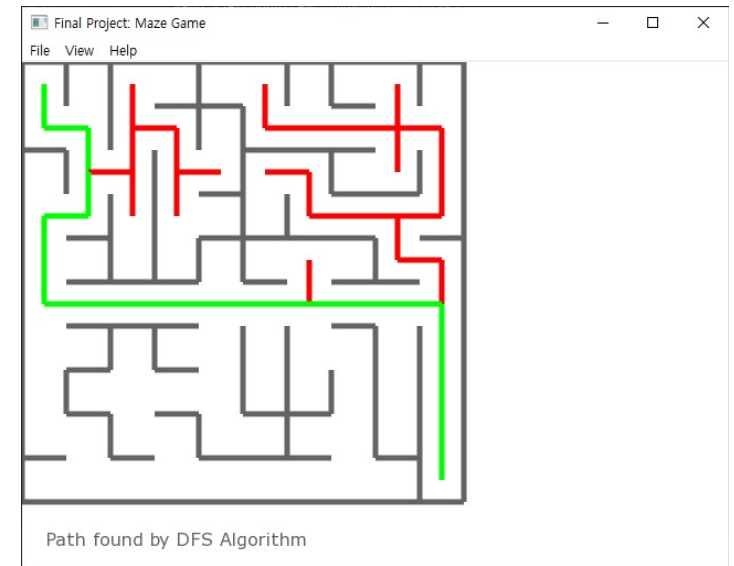
- 사용자가 만든 미로 + 내부에서 생성된 미로를 이용
- 사용자가 미로 찾기 게임 실행
- DFS, BFS 알고리즘으로 경로 찾기



미로찾기 과정 중

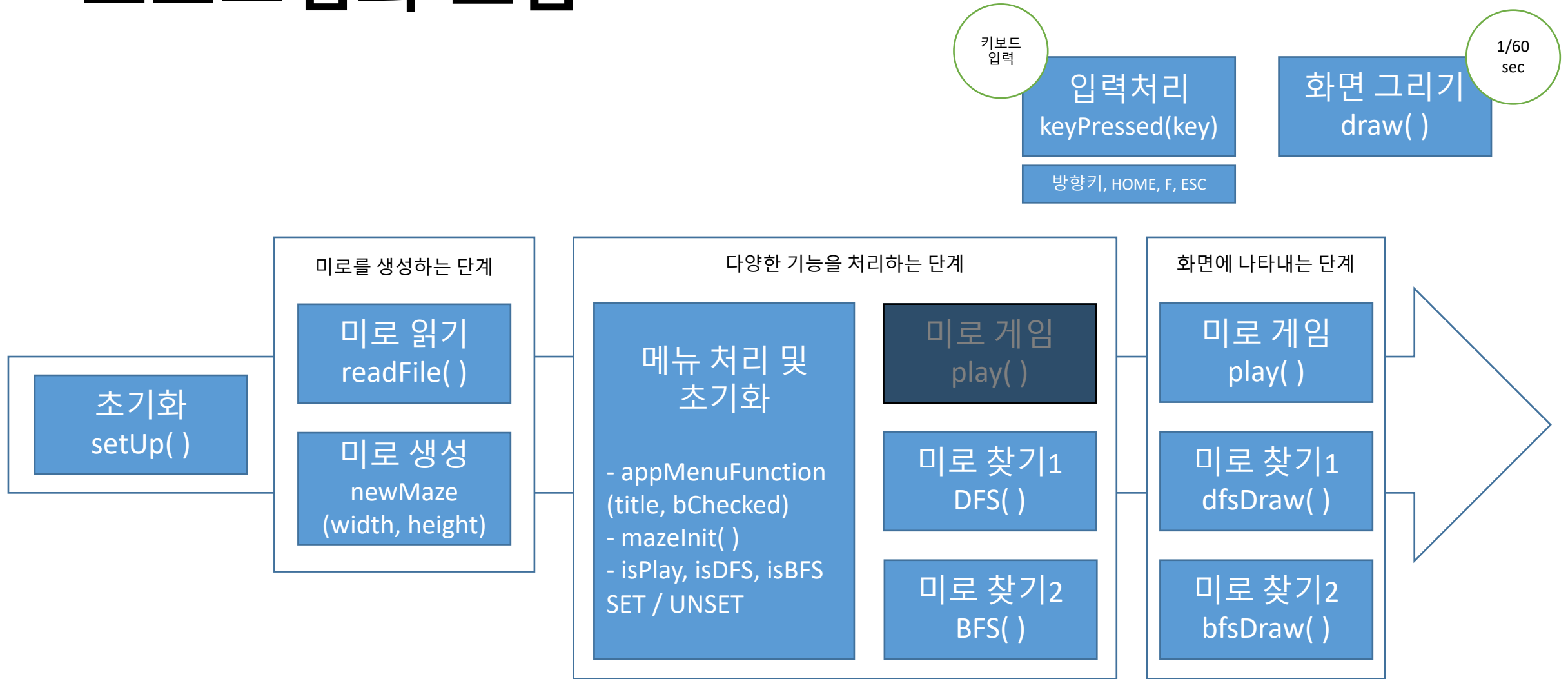


미로찾기 완료한 모습



DFS 알고리즘으로 찾은 경로

프로그램의 흐름



프로그램 시현

자료구조와 알고리즘: 자료구조

```
int** input = (int**)malloc(sizeof(int*) * HEIGHT);
input[0] = (int*)malloc(sizeof(int) * HEIGHT * WIDTH);
for(i = 1; i < HEIGHT; i++)
    input[i] = input[i-1] + WIDTH;
```

```
stack<point> stck;
queue<point> qu;
```

```
typedef struct point {
    int y;
    int x;
}point;
```

❖ 동적 할당 배열

- 미로의 모양
- 미로 방문 여부
- (BFS 알고리즘) 부모 노드의 정보 저장

❖ 스택과 큐

- DFS, BFS 알고리즘 구현시

❖ 2D 좌표계 성분 구조체

- 사용자에게 의한 미로 찾기 게임 진행시 위치
- (BFS 알고리즘) 부모 노드를 저장하는 배열의 성분

자료구조와 알고리즘: 알고리즘

❖ Eller's Algorithm

- 미로 생성 알고리즘
- 시간복잡도: $O(MN)$

❖ DFS

- 미로 경로 찾기 알고리즘
- 방문할 수 있는 최대 깊이까지 탐색한 후에 다른 경로 탐색
- 스택 사용
- 시간복잡도: $O(MN)$

❖ BFS

- 미로 경로 찾기 알고리즘
- 방문할 수 있는 모든 경로를 같은 시점에 탐색
- 큐 사용
- 시간복잡도: $O(MN)$

자료구조와 알고리즘: 구현

❖ 주요 변수

- `char** input; /*미로*/`
- `int** visited; /*방문한 위치*/`
- `point** backMaze; /*BFS 구현을 위함*/`
- `stack <point> stck; /*DFS를 위함*/`
- `queue <point> qu; /*BFS를 위함*/`
- `point player = { 1, 1}; /*초기값은 1, 1*/`

❖ 주요 함수

- `void setUp(); /*프로그램 초기화*/`
- `void draw(); /*1/60초마다 화면 그림*/`
- `bool readFile(); /*.maz 파일 읽기*/`
- `bool newMaze(); /*미로 생성*/`
- `void appMenuFunction(string, bool);`
- `void keyPressed(int); /*키보드 입력 처리*/`
- `void play(); /*미로 게임*/`
- `void mazelnit(); /*미로 초기화*/`
- `bool DFS(); void dfsDraw(); /*DFS*/`
- `bool BFS(); void bfsDraw(); /*BFS*/`

창의적인 구현

- ❖ 사용자에게 의한 미로 문제 해결 기능 추가
 - 방향키와 HOME 키를 이용하여 미로 해결
- ❖ 개별적인 프로그램으로 나뉘어져 있던 두개의 프로그램을 합침
 - 사용자가 미리 미로를 만들지 않아도 프로그램 내부에서 미로 생성 가능
 - 3가지 난이도: 10x10, 15x15, 20x20 크기로 제공

감사합니다 :-)