

Project 1: Maze Generator

프로젝트 소개

- 첫 번째 프로젝트는 탐색을 통해 미로를 만들고, 만든 미로를 다시 탐색해보는 것입니다.
- 프로젝트는 코드의 구현보다는 "실험"을 해보시라는 취지입니다.
 - 그래서 당연히 정답이 없습니다.
 - 제출하실 때 코드와 함께 짧은 (A4용지 2~3장 내외) 보고서를 같이 내 주세요.
- Stack과 Queue 강의에서 Location2D 미로를 탐색하는 두 가지 알고리즘을 배웠습니다
 - 깊이 우선 탐색 (Depth First Search)
 - 너비 우선 탐색 (Breadth First Search)
- 미로를 만드는 방법은 여러가지가 있지만, 깊이 우선 탐색을 사용하여 구현할 수 있습니다.
 - 눈으로 보시면 이해가 편하시겠죠.
 - 제가 만든 영상은 아니지만 [동영상](#) 을 보시면 대충 어떤 과정인지 느낌이 오실겁니다.
- 이번 프로젝트에서는 DFS를 통해서 미로를 만들고, 이 만든 미로를 DFS/BFS 두 가지 방식으로 탐색을 하는 것을 구현하시면 됩니다
 - 미로의 크기는 임의의 정수 n 에 대해서 $n * n$ 으로 주어집니다
 - n 의 최대 크기는 임의로 설정하시면 됩니다.
 - 저는 제출하신 코드는 실제 구현을 하셨는지만 검사할 예정입니다.
 - 그렇기 때문에 제출하신 코드의 구현 질은 고려하지 않습니다.

해야 할 일들

- 프로젝트는 자세한 구현과 함수의 이름을 드리지 않습니다. 본인이 직접 함수 이름도 정하시고, 실험도 직접 해 보시면 됩니다.
- 미로를 만드는 것을 구현합니다
 - 미로의 조건은 아래와 같습니다
 - 시작점이 존재해야 하고
 - 종료점이 존재해야 하고
 - 두 점을 이어주는 길이 존재해야 합니다
 - 미로의 길 칸 4개가 블록으로 접해있을 수 없습니다.
 - 블록으로 접해있다는 뜻은 좌표 (a, b) 가 길일 때, $(a, b-1)$, $(a-1, b)$, $(a-1, b-1)$ 까지 네 개가 모두 길일 수 없다는 뜻입니다.
 - 다르게 말하자면, 길은 최소한 주변 8개의 점 중 4개 이상이 벽이어야 합니다.
 - 일단 $(0,0)$ 에서 탐색을 시작하게 하시고, 종료점은 시작점에 접하지 않은 두 테두리 (예를 들어서 $row = n-1$ 혹은 $col = n-1$ 인 좌표에 처음 도달한 경우를 종료점으로 가정하시면 됩니다.
 - 최종 점에 도달하더라도 더이상 길을 만들 수 없을 때 까지 DFS를 지속합니다. 이를 통해서 전체 미로를 만듭니다.
 - 해보실 수 있다면 더 복잡한 방법으로 생성하는 미로를 구현하셔도 됩니다.

- 힌트들
 - 처음에는 모든 점이 벽이라고 가정하시면 구현이 편합니다.
 - 시작점부터 한 점씩 벽을 길로 바꾸면서 지나간다고 생각해보세요. 다르게 말하자면, 빈 공간에서 그냥 자유롭게 이동하고, 그 길을 기록하는 것과 거의 같습니다. 다만 갈 수 있는 경로에 약간의 제약조건이 있을 뿐.
 - 예제 구현처럼 네 방향으로 돌아다니는 것에 정해진 순서를 두면 제대로된 미로가 나오지 않습니다. 상하좌우를 임의로 고르게 해 보세요 (random 함수 등을 사용해 보세요).
 - 이동은 상하좌우 네 방향으로만 하셔야 합니다 (대각선 이동을 허용하면 복잡해집니다)
 - 추가적으로
 - 꼭 Location2D로 구현하지 않으셔도 됩니다. 2차원 구조를 Graph나 다른 방법으로 구현하실 수 있다면 그것도 좋습니다.
 - 실제 여러분이 대학에서, 그리고 사회에 나가서 접하는 문제를 풀기 위한 정답은 한가지가 아닐 수 있습니다. 그리고 거기서 가장 좋은 답을 찾는 방법을 배우는 것이 여러분이 소프트웨어 엔지니어로서 가지는 가장 큰 역량이 될 것이라고 생각합니다.
- 이렇게 만들어진 임의 생성 미로에 대해서 여러가지 실험을 해 보세요
 - 예를 들면 크기가 달라질 때 평균적인 탐색 횟수가 BFS/DFS가 어떻게 달라지는지 보시거나
 - DFS로 만들어진 경로의 "평균" 길이를 보시거나
 - 미로를 만들 때 이미 생긴 길을 다시 방문을 허용하는지 여부에 따라서 어떻게 두 방식이 탐색하는 결과가 달라지는지를 보시거나
 - 위에 해당하지 않아도 다양한 실험을 해 보시면 좋을 것 같습니다.
 - 꼭 위의 실험을 하지 않아도 됩니다. 특성을 보기 위한 다른 종류의 실험을 하셔도 됩니다.
- 미로 "만들기"를 전혀 못하겠다 라고 하시는 분은, 뒤의 실험 파트를 포기하지 마세요
 - 이미 DFS와 BFS는 수업시간에 구현한 코드가 있습니다.
 - 미로를 생성을 못해도, 특정 조건의 미로를 수동으로 만들어서 실험하실 수도 있습니다. 그러니 포기하지 마세요.

제출 방법

- Due Date: 2020년 5월 22일 23시 59분 59초
 - 바로 다음 혹은 듀가 끝나기 전에 Project 2번도 나갈 계획입니다.
- 이메일로 제출합니다: jinhyuk.yun@ssu.ac.kr
 - 이메일 제목은 [자료구조 PR1] 20202020 홍길동 형태로 보내주세요
 - 이 형태가 아닌 경우 제대로 제출되지 않을 수 있습니다.
- 보고서 PDF파일과 소스코드를 압축해서 한 파일로 만들어주세요
 - 파일은 DS2020_PR1_20202020_홍길동.zip 형태로 부탁드립니다
 - 제발 간절히 부탁드립니다. egg 혹은 alz는 쓰지 말아주세요... (감점할거예요)
- 구현체는 제대로 작동을 하면 어떤 형태도 좋습니다.
 - 구현체는 구현 여부로 평가하겠습니다
 - 점수의 차이는 주로 해 보신 실험과 보고서에서 있을 것입니다. 재밌는 실험 많이 해보세요.
 - 굿럭!