**4 게임 스테이지**

4.1 설정

게임을 시작할 때 플레이어는 난이도를 선택해야합니다. 난이도는 퍼즐의 행과 열 수를 나타내는 정수입니다. 이것은 퍼즐에 사용되는 단어의 크기를 결정하며 각 단어가 한 행에 들어갑니다. 플레이어가 제공하는 난이도 (퍼즐 크기)는 2에서 14 사이의 정수라고 가정할 수 있습니다.

난이도를 지정한 후 단어 파일에 제공된 단어에서 솔루션이 생성됩니다. 그런 다음 솔루션의 타일을 섞고 오른쪽 아래 모서리에 있는 타일을 빈 타일로 바꾸면 섞인 퍼즐이 생성됩니다.

4.2 퍼즐 풀기

플레이어는 목표에 도달하기 위해 타일을 밀어 퍼즐을 풀려고 합니다. 매 턴마다 솔루션과 게임의 현재 상태가 표시됩니다. 그런 다음 플레이어는 작업을 입력하라는 메시지가 표시됩니다. 조치는 다음 중 하나 일 수 있습니다.

“H : 도움말 메시지를 표시

“GU” : 현재 게임을 중지

“U” or “D” or “L” or “R” : 빈타일을 위/아래/왼쪽/오른쪽으로 각각 이동

플레이어가 게임에서 이기거나 "GU"을 입력하여 포기할 때까지 프롬프트가 반복됩니다.

4.3 게임의 끝

게임이 끝나면 플레이어는 새 게임을 시작할지 또는 프로그램을 중지할지 선택하라는 메시지가 표시됩니다.

“Y” or “y” or “” : 새로운 게임 시작

제외한 다른 것: 프로그램 종료

새로운 게임을 시작하기로 결정했다면 새로운 난이도를 설정해야 합니다. 새로운 솔루션과 퍼즐이 생성되고 새로운 게임이 시작됩니다.

**5 규칙**

이 과제는 아래에 지정된 규칙을 따릅니다.

1. 솔루션: 솔루션은 문자열에 저장됩니다. 이 문자열에는 특수 문자 및 / 또는 새 줄 없이 그리드의 모든 문자가 왼쪽에서 오른쪽으로 그리고 위에서 아래로 포함됩니다. 예를 들어, “dogcatpig” 솔루션은 아래 그리드를 나타냅니다.

d o g

c a t

p I g

2. 퍼즐: 퍼즐은 마지막 문자를 제외하고 솔루션의 복사 본인 문자열입니다. 퍼즐의 모든 캐릭터는 게임이 시작될 때 섞입니다. 빈 타일을 나타내기 위해 공백 문자 (예 : "")가 퍼즐 끝에 추가됩니다. 예를 들어, 여러 번 이동한 후 퍼즐 "fhgbcade"는 아래 그리드로 표시됩니다.

f h g

b c

a d e

타일을 슬라이딩 하는 논리를 구현하려면 슬라이딩 되는 문자열의 문자 위치를 공백 문자의 위치로 바꿔야합니다. 논리는 문자와 공간이 그리드에서 서로 인접한 경우에만 스왑을 수행하도록 해야 합니다.

3. 방향: 방향은 "U"또는 "D"또는 "L"또는 "R"이 될 수 있으며 위/아래/왼쪽/오른쪽에 해당합니다.

**6 구현**

다음 함수는 a1.py에서 구현되어야 합니다. 논리에 도움이 되거나 코드를 더 쉽게 이해할 수 있도록 추가 기능을 구현할 수 있습니다. 이러한 기능은 난이도가 증가하는 순서대로 나열되었습니다.

(1)win (puzzle : str, solution : str) 확인-> bool

퍼즐과 솔루션이 주어지면 게임이 이기면 True를 반환하고 그렇지 않으면 False를 반환합니다. check win 호출의 예 :

(2) 스왑 위치 (puzzle : str, from index : int, to index : int)-> str

인덱스에서 문자의 위치를 바꾸고 퍼즐에서 인덱스로 바꾸고 업데이트 된 퍼즐을 반환합니다. 스왑 포지션 호출의 예 :

(3) move (puzzle : str, direction : str)-> Optional [str]

빈 타일을 주어진 방향으로 이동하고 업데이트 된 퍼즐을 반환합니다. 이동할 수 없는 경우 (즉, 지정된 방향으로 이동하면 빈 타일이 그리드 외부에 있음) None을 반환합니다.

이동 호출의 예 :

----

위 예제의 첫 번째 코드 줄은 빈 타일을 아래쪽 모서리에서 위로 한 타일씩 이동합니다. 다음 그래픽 데모에 해당합니다.

두 번째 코드 줄은 아무것도 반환하지 않습니다. 빈 타일이 오른쪽 하단에 있고 오른쪽으로 이동할 수 없기 때문입니다.

(4)그리드 인쇄 (퍼즐 : str)-> 없음

사용자 친화적 인 형식으로 퍼즐을 표시합니다. 인쇄 그리드 호출의 예 :

(5)main ()-> 없음

섹션 4.2에 지정된대로 게임에서 주요 사용자 상호 작용을 처리합니다. print 문을 프로그램의 기본 논리로 바꿔야합니다.

6.1 지원 코드

파일 a1.py에는 셔플 퍼즐 기능의 구현이 포함되어 있습니다. 이 기능을 수정해서는 안됩니다. 셔플 퍼즐을 만들려면 이 함수를 게임당 한 번만 호출해야 합니다.

셔플 퍼즐 (솔루션 : str)-> str

주어진 솔루션에 따라 풀 수 있는 슬라이딩 게임을 생성합니다.

이 함수는 위에 지정된 스왑 위치를 호출합니다. 셔플 퍼즐을 성공적으로 호출하려면 스왑 위치를 구현해야 합니다.

할당을 구현하려면 a1 support.py에 제공된 코드를 사용해야합니다. 예상치 못한 오류가 발생할 수 있으므로 a1 support.py를 변경하지 마십시오. 이 파일에는 미리 정의된 몇 가지 명명된 상수 및 함수가 있습니다. 게임 구현 시 제공된 명명된 상수를 사용해야합니다. 세 가지 함수 중 다음을 호출하기 만하면 됩니다.

게임 솔루션 얻기 (파일 이름 : str, 그리드 크기 : int)-> str

이 함수는 퍼즐의 해답 인 문자열을 반환합니다. 첫 번째 매개 변수는

단어를 로드 할 파일입니다. 두 번째 매개 변수는 그리드의 크기입니다.