플레이봇 퍼즐게임 보고서

**- 깃발을 찾아서 -**

**< 코드 설명 >  
< 플레이봇 편>**

**- 만든 이: 원인식 (MrWon)   
- 만든 날짜: 2020.1.20**

스크린샷이(가) 표시된 사진

자동 생성된 설명  
- playbot\_on\_move() : 플레이봇을 움직이게 하는 함수

- 768 : while(1)을 이용해 간단한 무한루프를 만들어줌  
- 769~771 : 현재 위치,방향을 저장  
- 773~775 : 바닥에 아이템/오브젝트가 있다면 그 아이템/오브젝트와 상호작용함  
- 777~739 : 상호작용 후 모종의 이유로 게임이 끝나게 된다면 break를 통해 무한루프문을 끝냄  
- 781~784 : 점프대와 상호작용 시 점프한 후 바닥에 상호작용해야 할 아이템/오브젝트가 또 있을 지 모르니 한칸 가지 말고 contiue로 다시 처음으로 돌아감  
- 786~788 : 만약 앞에 움직일 수 없는 아이템이 있다면 게임을 멈추라는 신호를 보냄.  
- 789~792 : 앞에 아무것도 없다면 딜레이를 준 다음 움직임 (기본 속도가 -1이라 딜레이를 안주면 게임 시작 버튼을 누름과 동시에 결과가 나오는 광경을 목격할 수 있음)  
- 794~797 : 앞으로 움직일려고 한 뒤 게임을 멈추라는 신호가 보내지면 break를 통해 무한루프문을 끝냄. 만약 이 구문이 없다면 종을 밟았을 때 종을 한번 더 치고 게임이 종료됨  
스크린샷, 텍스트이(가) 표시된 사진

자동 생성된 설명  
- interacting(name,x,y,d) : r1이 서있는 아이템과 상호작용하는 함수

- 801 : switch 문으로 아이템에 따라 상호작용할 행동을 나눔  
- 802~815 : 서있는 아이템이 화살표라면 딜레이를 준 뒤 화살표 방향대로 몸을 돌림  
- 816~825 : 서있는 아이템이 점프대라면 두칸 뒤에 움직일 수 없는 아이템이 있는지 없는지 확인 후 딜레이를 주고 점프함. 있으면 게임을 멈추라는 신호를 보냄  
- 826~830 : 서있는 오브젝트가 종(오브젝트 번호: 4)라면 약간의 딜레이를 준 뒤 맵에 있는 모든 문을 열거나 닫은 뒤 ringing의 값을 1 증가시킴. 증가시키는 이유는 변수 선언편을 참고바람. smart\_delay를 쓰는 이유는 그냥 딜레이를 쓰면 종을 치는 속도가 앞으로 움직이는 속도와 같아서 미묘하게 어색해서 smart\_delay를 쓰는 것임.  
- 831~834 : 서있는 오브젝트가 포탈류 (포탈류의 오브젝트 이름은 전부 “유리블럭”으로 똑같음) 라면 포탈의 색을 확인해 itemname을 그 색에 해당하는 포탈의 오브젝트 번호로 바꿈  
- 837~848 : 그 후 맵 전체를 스캔해서 r1이 서있는 포탈과 또다른 포탈을 찾아서 그 포탈로 이동함 (사족: port\_x/y 배열을 이용했다면 연산이 좀더 빨랐을 것)  
- 849 : 서있는 오브젝트가 도착 깃발(오브젝트 번호: 9)이라면 게임에서 이겼다는 신호를 보냄

스크린샷이(가) 표시된 사진

자동 생성된 설명  
- front\_is\_what\_item(x,y,d,r) : (x,y)에서 d 방향의 r칸 앞에 무슨 오브젝트가 있는지를 알려주는 함수  
- 856: (x,y)는 위치, d는 방향, r은 거리임. (d+1)%2\*(!((d-d%2)/2)\*2-1)는 x좌표 전용식으로 d가 0(동쪽)이면 1, 2(서쪽)이면 -1을, 그 외엔 0을 반환하는 식이며 d%2\*(!((d-d%2)/2)\*-1)은 y좌표 전용식으로 d가 1(북쪽)이면 1을 3(남쪽)이면 -1을, 그 외엔 0을 반환하는 식이다. 이 식에 거리를 나타내는 r을 곱해 d 방향의 r칸 앞에 무슨 오브젝트가 있는 지를 알려줄 수 있는 것이다.

-   
스크린샷이(가) 표시된 사진

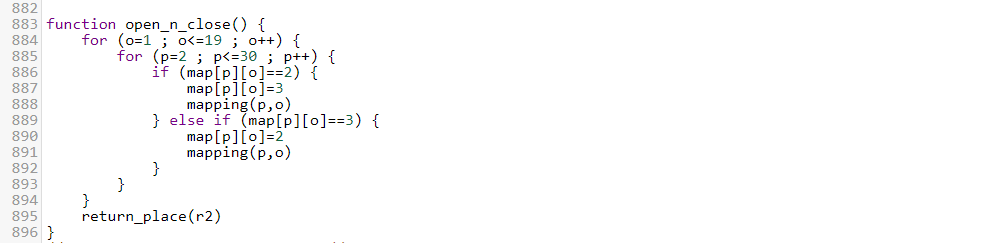
자동 생성된 설명  
- front\_is\_unmovable\_item(x,y,d,r) : front\_is\_what\_item(x,y,d,r)을 이용해 r칸 전방에 움직일 수 없는 아이템이 있는지 없는지를 확인하는 함수  
- 860 : r칸 전방에 있는 아이템의 오브젝트 번호 저장  
- 861~862 : 그 아이템의 오브젝트 번호가 벽(1)이나 닫힌 문(2)라면 1을 반환하고 그게 아니라면 0을 반환함  
  
스크린샷이(가) 표시된 사진

자동 생성된 설명  
- jump(x,y,d) : 1칸을 점프하는 함수  
- 866 : front\_is\_what\_item(x,y,d,r)에서 사용한 식을 이용해 2칸 앞으로 이동함  
- 867~868 : 현재 위치 저장

- 참고 표

|  |  |  |
| --- | --- | --- |
|  | x: (d+1)%2\*(!((d-d%2)/2)\*2-1) | y: d%2\*(!((d-d%2)/2)\*-1) |
| 동쪽(d=0) | +1 | 0 |
| 북쪽(d=1) | 0 | +1 |
| 서쪽(d=2) | -1 | 0 |
| 남쪽(d=3) | 0 | -1 |

  
- delay() : speed만큼 딜레이를 내주는 함수  
- 872~874 : space\_jump(x,y)는 딜레이가 있는 함수다. 이를 이용해 딜레이를 낸다.

  
- smart\_delay() : spd만큼 딜레이를 내주는 함수  
- 872~874 : delay()와 똑같은 원리, 다른 점이 있다면 스피드 입력값이 있다는 것이다.  
  
  
- open\_n\_close() : 맵에 있는 모든 문을 열거나 닫는 함수  
- 884~894 : 맵 데이터의 모든 열린 문(3)은 닫힌 문(2)으로, 닫힌 문은(2) 열린 문(3)으로 바꾼 뒤 그 자리만 mapping(x,y) 함수를 이용해 매핑함.  
- 895 : r2를 제자리로 돌아가게 만듬