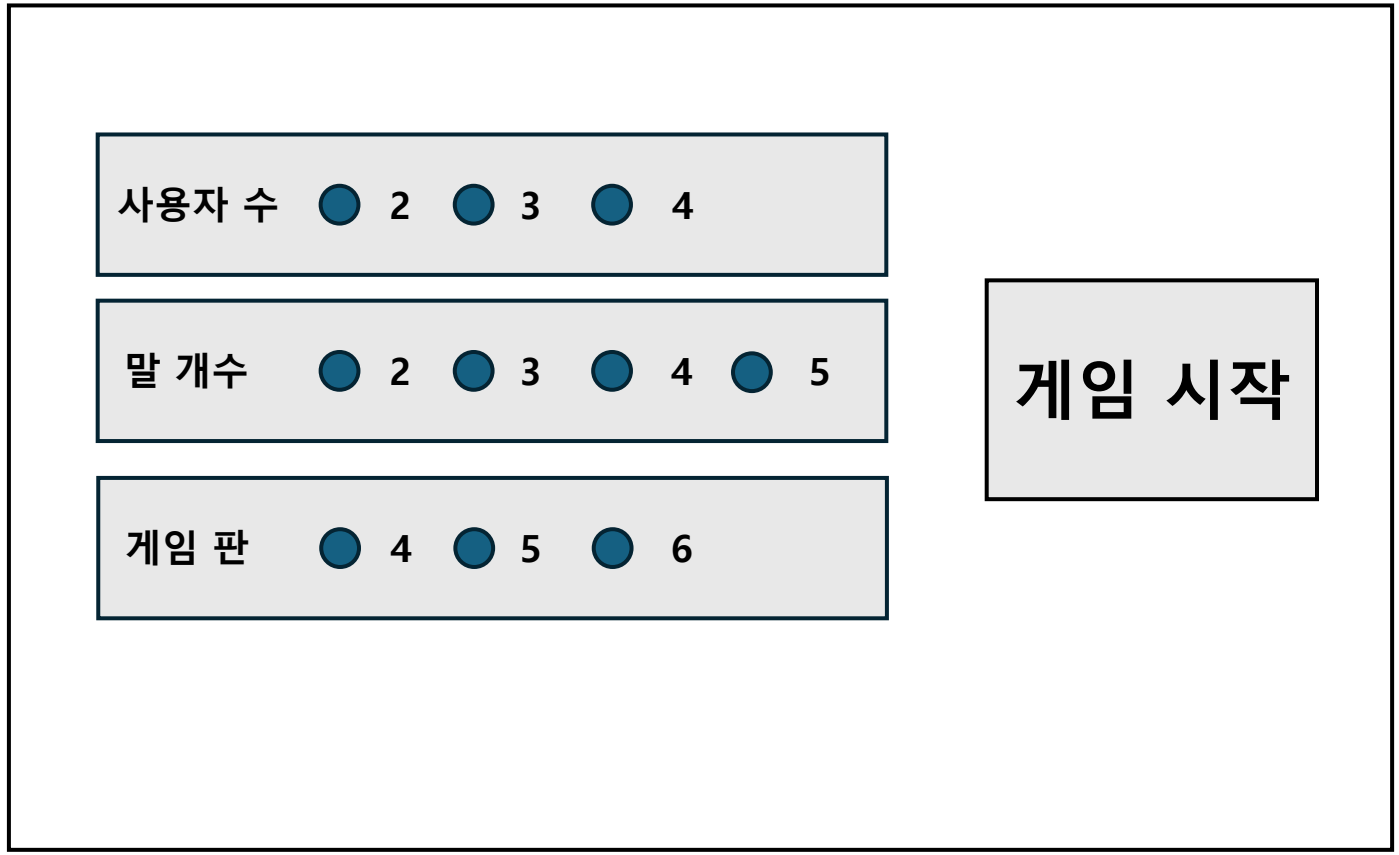
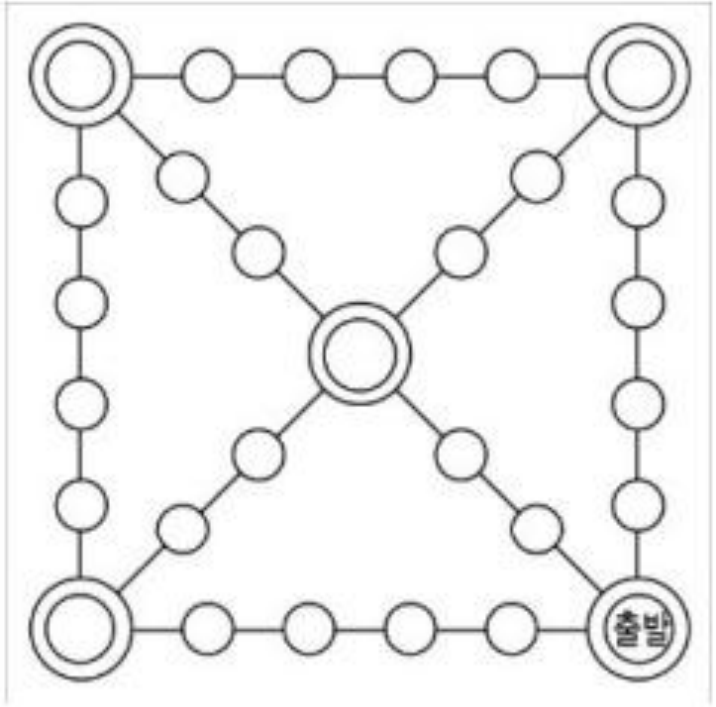
게임 시작 UI (사용자 입력 받기)





P1 차례

P1 : 남은 윷 n개 [n은 남은 윷 개수]

P2 : 남은 윷 4개

P3 : 남은 윷 2개

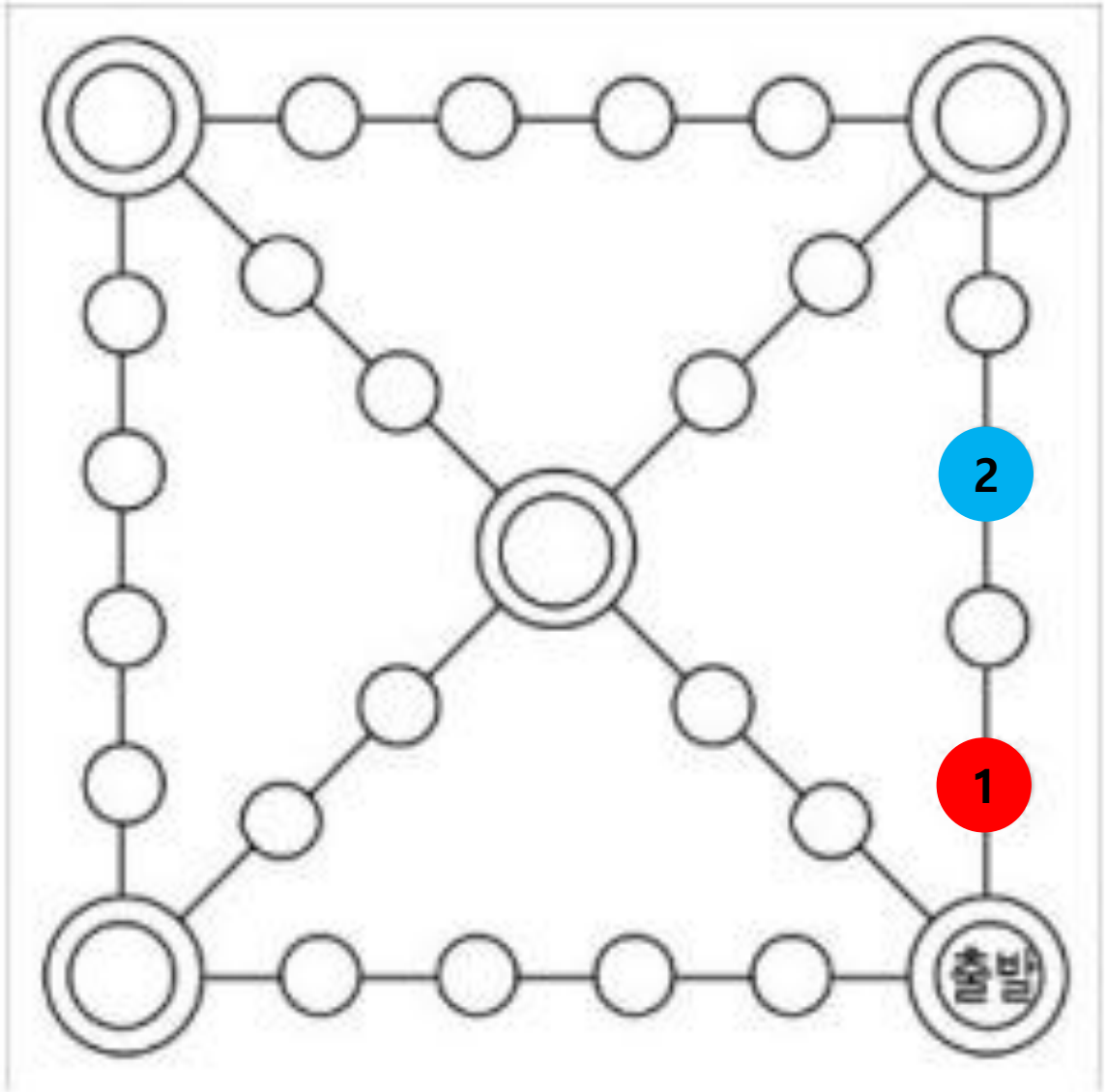
P4 : 남은 윷 4개

백도 | 도 | 개 | 걸 | 윷 | 모

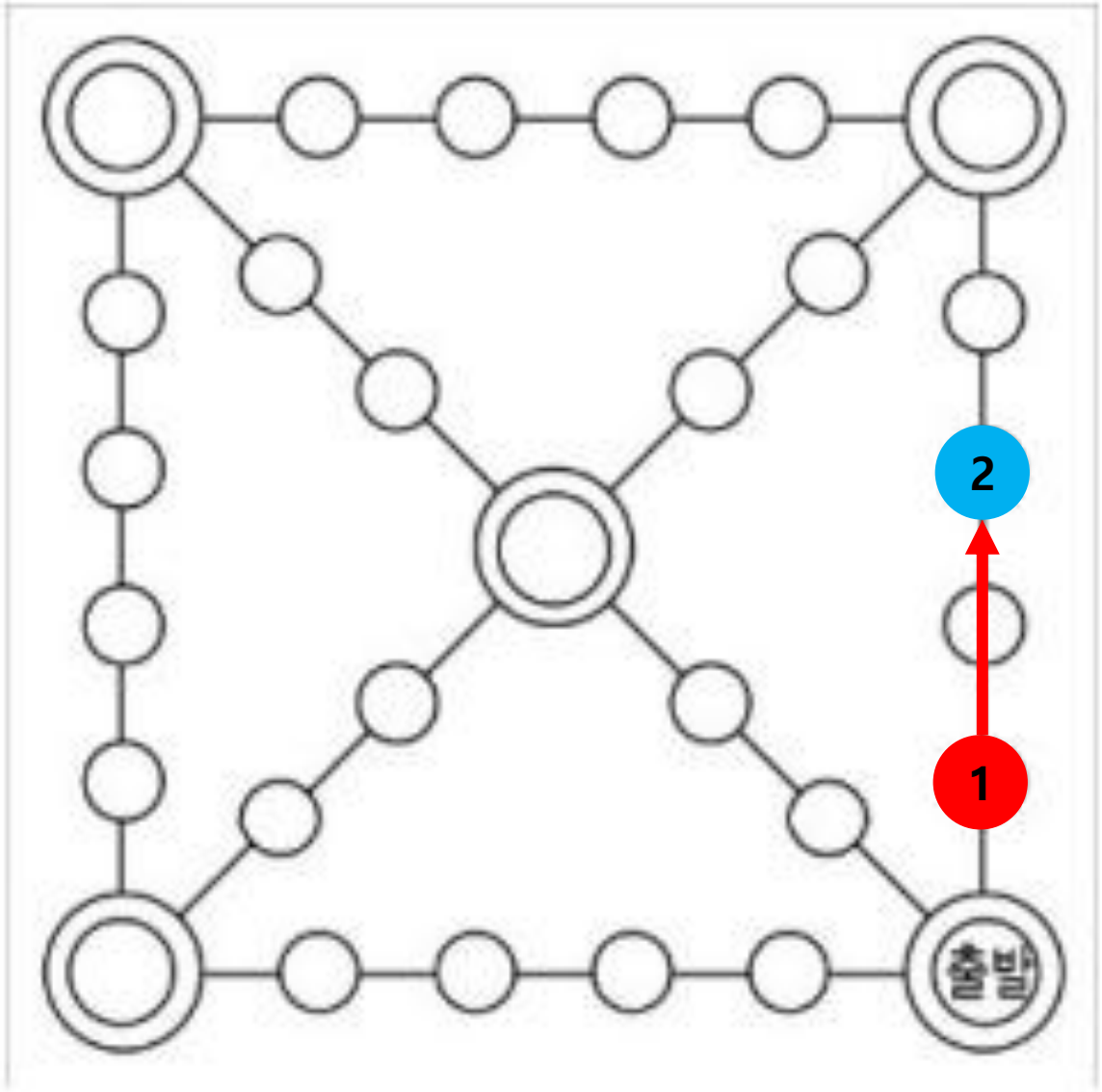
내보내기 | 판에서 선택

랜덤 | 윷 결과 (ex: 도!)

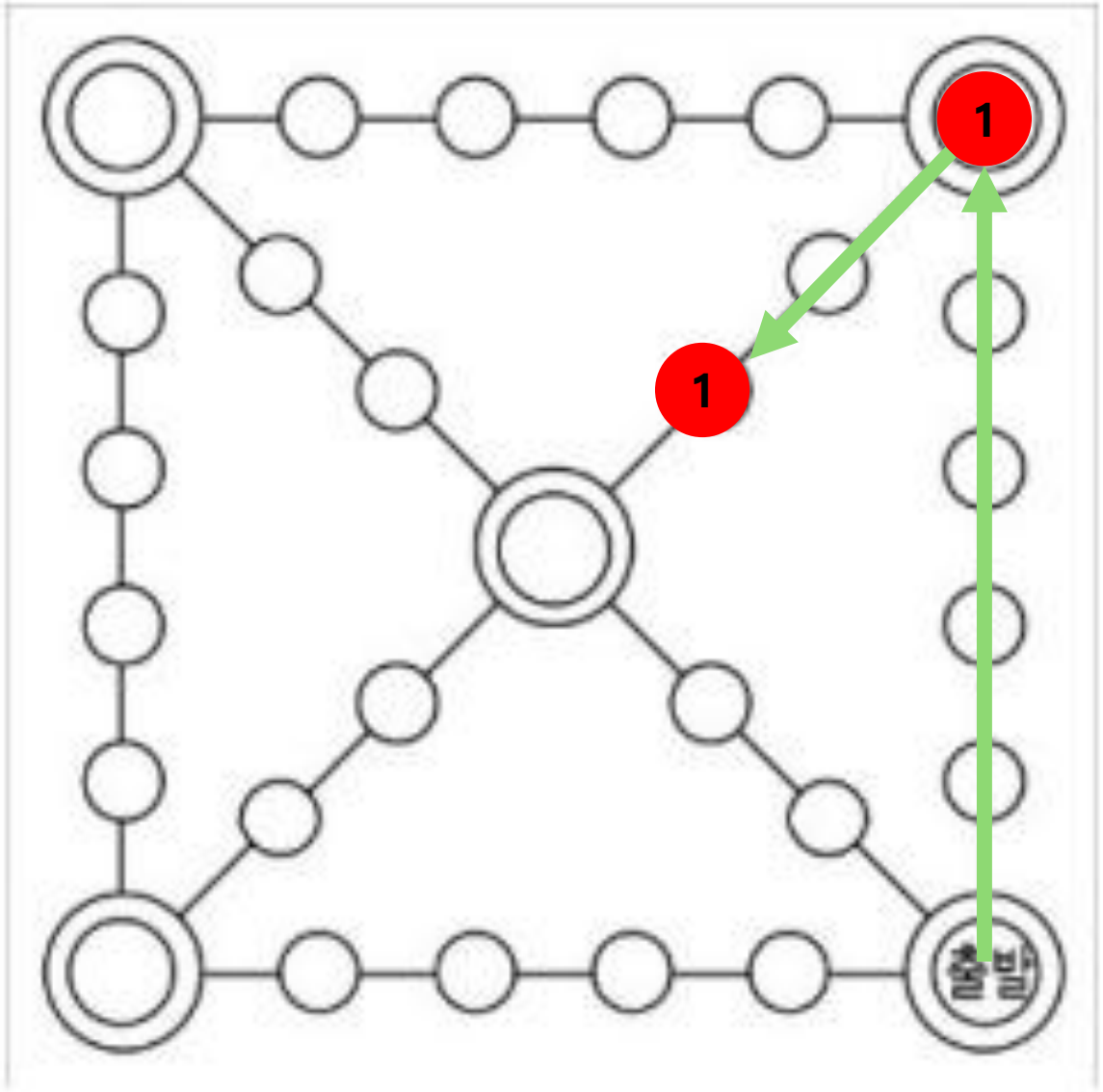
윷 스택 (윷, 모, 상대말을 잡으면 저장)



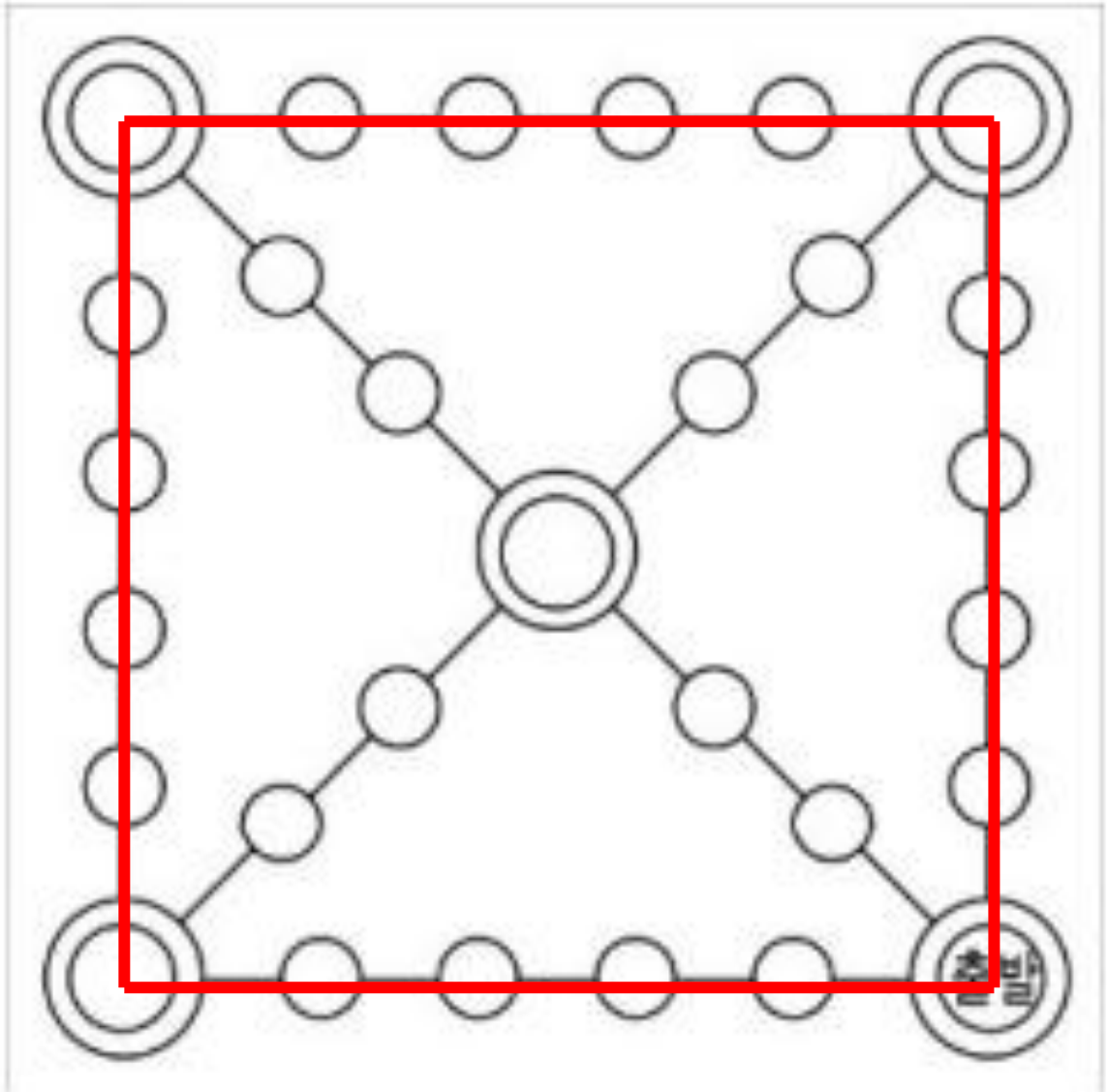
- (a) 플레이어가 2명일 때를 가정
- (b) P1은 빨간색, P2는 파란색 가정
- (c) 그림과 같이 숫자와 색깔로 플레이어를 표현
- (d) Grouping을 숫자로 표현함으로써 코드의 복잡성을 줄일 수 있음



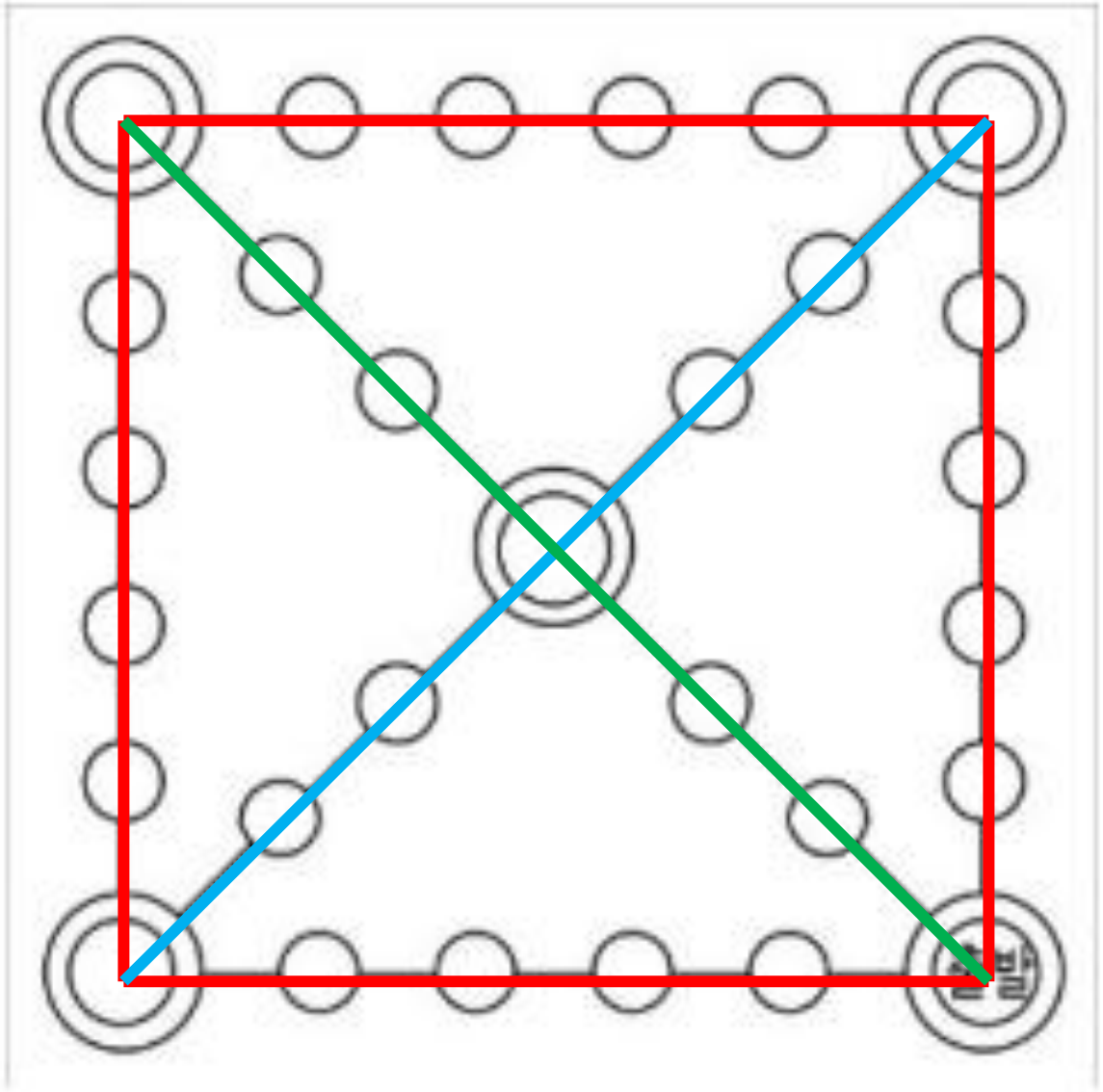
- (1) P1이 '개'가 나와서 P2를 잡는 상황을 가정
- (2) 우선 움직일 말에 다른 말이 있는지에 대한 여부를 탐색
- (3-1) 있으면 그 말이 아군인지 적인지를 판별
- (4-1-1) 적이라면 해당말의 색깔을 자신의 말로 바꾸고 숫자를 움직이는 말의 숫자로 바꿈. 그리고 잡힌 말의 숫자만큼 남은 말 표시를 바꾸어 줌
- (4-1-2) 아군이라면 해당말의 숫자를 가산해주어서 Grouping을 표현
- (3-2) 없으면 현재 자리의 숫자와 색깔을 흰색으로 전환
- (4-2) 해당 자리의 색깔을 지금 말의 색깔로 바꾸고 숫자도 똑같이 복사해서 집어넣음



- (a) 윷을 던졌을 때 결과가 모가 나온 다면 다시 던져서 결과를 대기함
- (b) 다시 던져서 개가 나온 경우 사용자가 '모'를 먼저 사용할 지 '개 ' 를 먼저 둘 지를 선택 해야함.
- (c) 그림의 경우는 모 – 개 순서로 사용자가 선택한 경우 임.

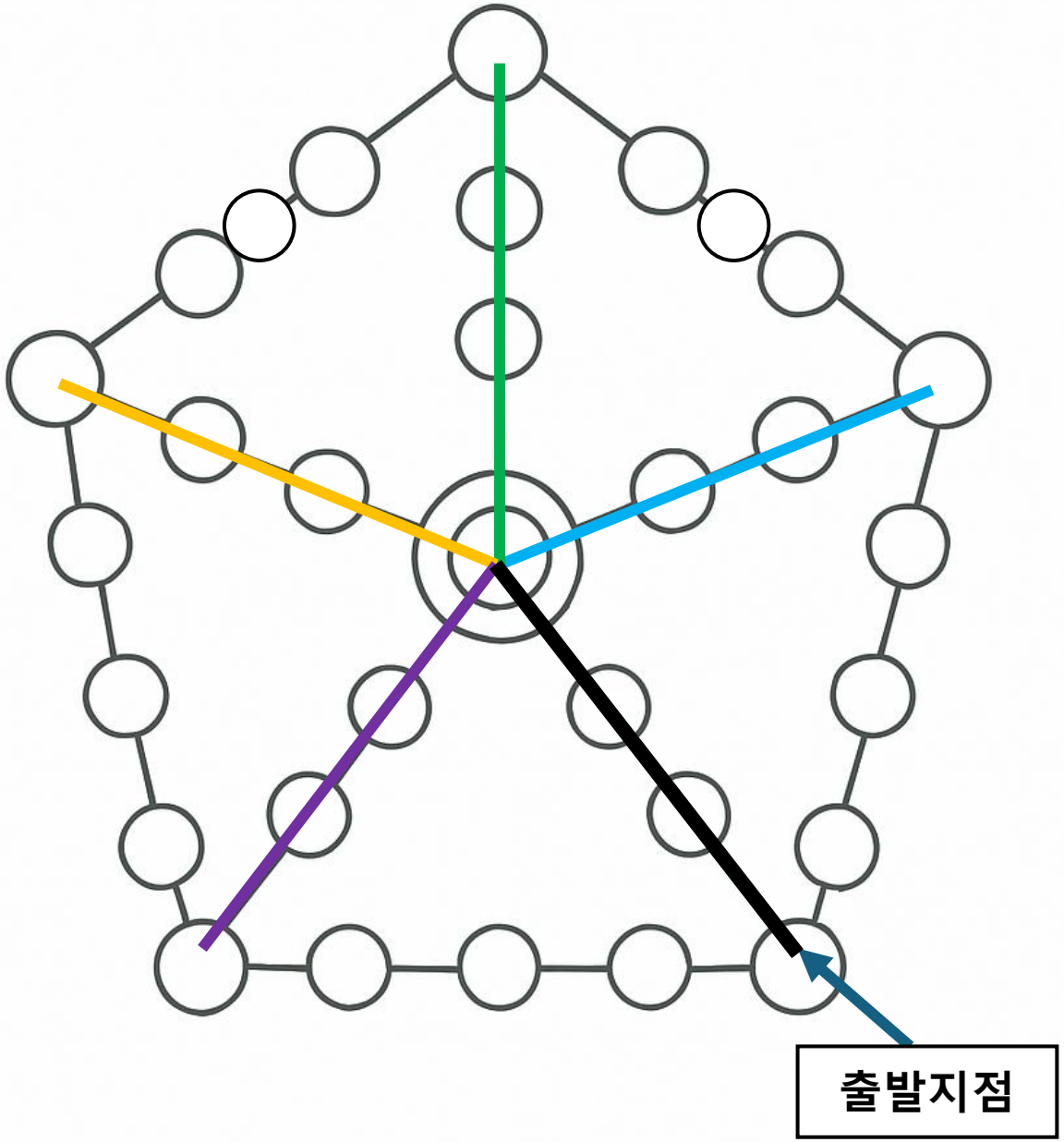


- (1) 해당 루트를 point[0][0~19]로 표현
- (2) 좌표가 point[0][5 or 10]일 때 안쪽으로 꺾는 것을 표현하기



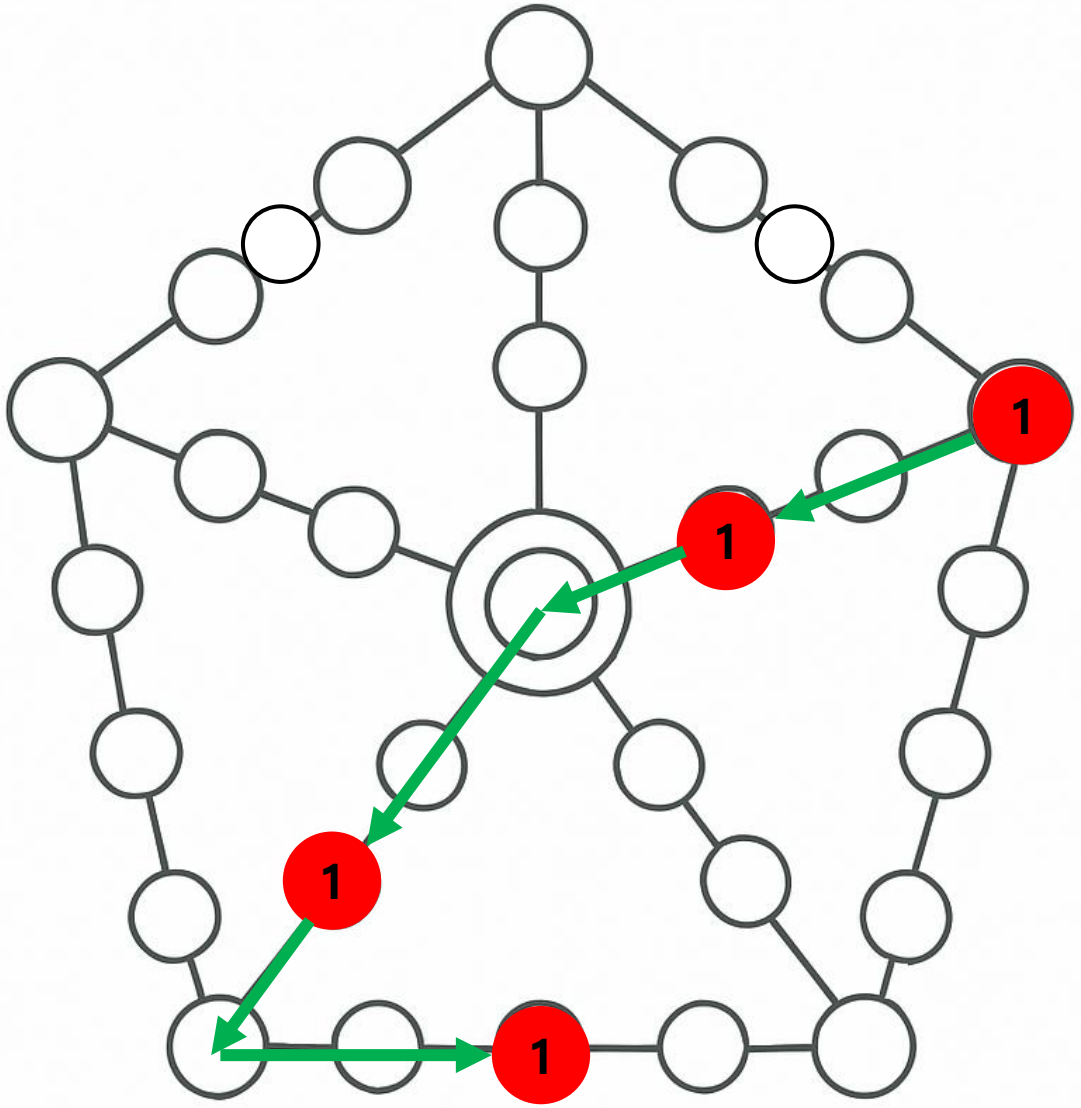
- (a) 파란 선을 point[1][0~6]으로 표현
- (b) 초록 선을 point[2][0~6]으로 표현
- (c) 빨간 선의 좌표가 point[0][5]가 된다면 point[1][0]으로 치환하기
- (d) 빨간 선의 좌표가 point[0][10]이면 point[2][0]으로 치환하기
- (e) 파란 선의 좌표가 point[1][3]이 된다면 초록 선상의 좌표로 변환 (point[2][3])

- <종점 판단>
- point[2][현재 좌표 + 움직일 좌표] > point[2][6] -> 남은 말 개수 하나 줄이기
- Point[0][현재 좌표 + 움직일 좌표] > point[0][20] -> 남은 말 개수 하나 줄이기
- 종료 후에 남은 창과 모든 말, player을 해제 해야함.

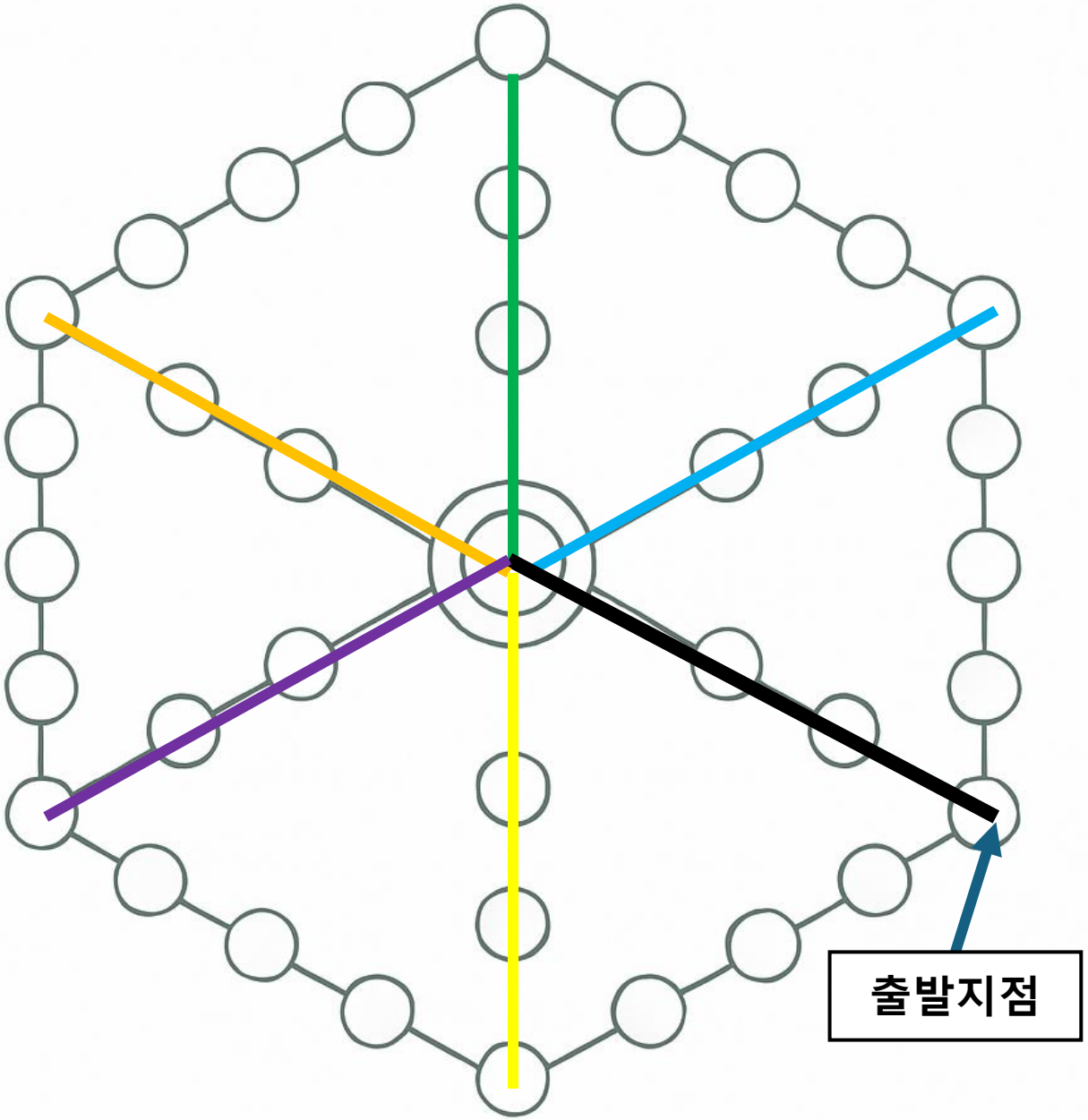


- (a) 5각형의 경우 모서리에서 중앙으로 향할 때 중앙 원을 지나치는 경우 보라색 선의 경로로 모두 보내면 된다.
- (b) 파란색 선은 point[1][0~3], 초록색 선은 point[2][0~3], 노란색 선은 point[3][0~3], 보라색 선은 point[4][0~4], 검은색 선은 point[5][0~3].
- (c) 예를 들어 point[0][5]에 위치한 말의 경우 point[1][0]의 위치와 동일하게 되는데 개가 나와 point[1][2]로 갔고 그 이후 걸이 나오면 중앙을 지나치고 point[4][2]을 가게 된다. 이와 같은 연산은 module 연산자를 이용해 초과한 길이를 계산하면 된다.

- 양 좌 우측 상단의 중앙의 원은 GPT가 그림을 잘 못 그려줘서 직접 넣은 것임을 유의한다.

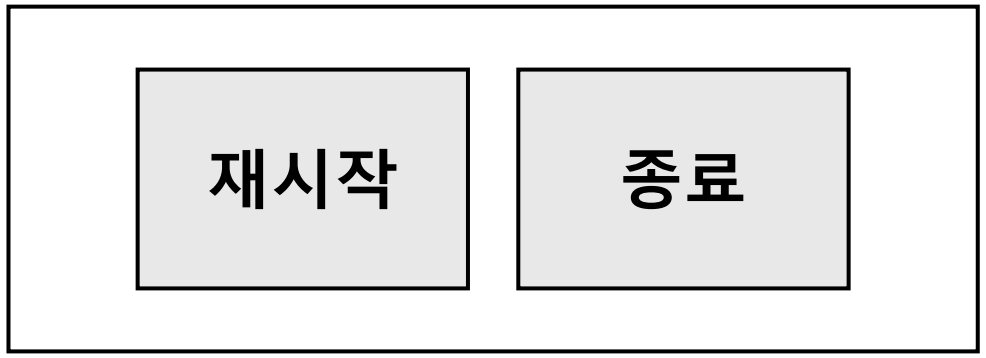


- (a) 현 시점에서 개 – 걸 – 걸이 나온 경우를 생각하자.
- (1) 현재 위치는 point[0][5]이며 이때 개가 나오면 point[1][0]으로 재 인식하고 +2을 해주어서 point[1][2]으로 간다.
- (2) 이후 걸이 나오므로 +3을 해주되 중앙을 지난다는 점을 생각해서 계산을 해야 한다. Point[4][2]로 이동하게 되는 것을 명심하자.
- (3) 이후 다시 걸이 나오므로 +3을 해주는데 다시 point[0]부분의 루트로 가게 된다. 최종 목적지는 point[0][22]



- (a) 6각형의 경우 모서리에서 중앙으로 향할 때 중앙 원을 지나치는 경우 노란색 선의 경로로 모두 보내면 된다.
- (b) 파란색 선은 point[1][0~3], 초록색 선은 point[2][0~3], 주황색 선은 point[3][0~3], 보라색 선은 point[4][0~4], 노란색 선은 point[5][0~3], 검은색 선은 point[6][0~3].
- (c) 예를 들어 point[0][5]에 위치한 말의 경우 point[1][0]의 위치와 동일하게 되는데 개가 나와 point[1][2]로 갔고 그 이후 걸이 나오면 중앙을 지나치고 point[5][2]을 가게 된다. 이와 같은 연산은 module 연산자를 이용해 초과한 길이를 계산하면 된다.

게임 종료 UI (사용자 입력 받기)



- **(a) 재시작 시 게임 시작 UI로 회귀**
- **(b) 종료 시 모든 창을 닫기**