

# Java Programming: Assignment 1

Seung-Hoon Na

May 6, 2018

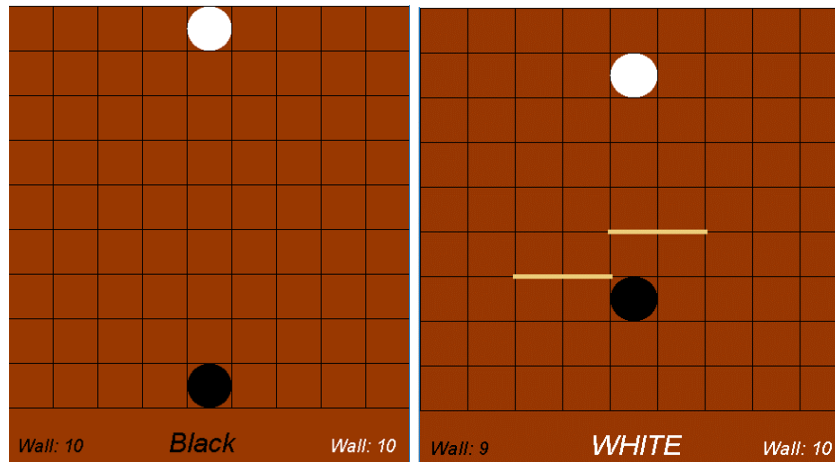
## 1 Quoridor

본 과제에서는 Quoridor 보드게임을 1인용 (인공지능기반), 2인용 (네트워크기반) 상에서 구동되도록 java로 프로그래밍하는 것을 목표로 한다.

### 1.1 게임 규칙

게임판은 9x9의 칸으로 나뉘어 있다. 게임판을 중앙에 놓고 가장 위쪽 중앙과 가장 아래쪽 중앙에 말을 위치시킨다. 각 플레이어는 10개의 장애물을 사용할 수 있다. 게임의 목표는 내 게임말을 상대방보다 먼저 상대방 쪽 시작 줄에 이동시키는 것이다.

플레이어는 번갈아 가면 2가지 중 하나의 행동을 할 수 있다. 말을 전, 후, 좌, 우로 1칸 움직이거나, 게임판에 장애물을 놓을 수 있다. 장애물은 게임말의 이동을 방해할 수 있는 요소이고 장애물을 설치할 때에는 최소 1칸의 이동할 수 있는 공간을 남겨두어야 한다. 이동하는 도중 상대방 말과 마주보는 경우 상대방을 뛰어넘을 수 있다.



### 1.2 Problem: Quoridor 게임 구현

앞절의 그림에서 처럼 Quoridor 아래 가이드라인을 따라 간단한 인공지능에 기반한 1인용 보드게임을 java로 작성하시오.

시작하기 전 화면은 다음과 같은 메뉴로 구성되어야 한다.

1. 1인용 게임

2. 2인용 게임
3. 2인용 네트워크 게임 (게임생성)
4. 2인용 네트워크 게임 (게임참여)
5. 이전게임 Load
6. 이전게임 복기
7. 환경설정
8. 종료

메뉴의 각 이이템은 버튼 (이미지 버튼)으로 구성하며 화면의 수식/수평 중앙정렬로 배치하라.

1인용의 경우, 플레이어는 검은말을 플레이한다. 말의 이동 및 장애물 설치하는 모두 마우스 클릭을 통해 이루어진다. 장애물 설치할 때는 게임판의 임의의 위치에 있는 줄에 마우스를 가져대면 2칸 거리에 장애물이 표시되어 클릭시 장애물이 설치된다. 컴퓨터는 흰색말을 플레이하며, 난수 생성에 기반한 간단한 인공지능을 통해 구현한다.

상대편의 끝지역에 도착시 승자의 이름을 팝업한다. 종료된 게임은 자동으로 저장되어, 나중에 복기해볼 수 있도록 한다.

아래는 구현과 관련된 상세 comments이다. 2인용 네트워크 게임 메뉴의 경우, 동일 컴퓨터상에서 localhost를 이용하여 2개의 개별 program으로 테스트가 되어야 한다.

(교재에 기술되지 않은 내용들은 각자 다른 참고자료를 통해 정보를 얻을 것).

- **게임모드에서 상태:** 현재 턴에서 “WHITE” 또는 “BLOCK”이라는 Status가 화면의 상단 또는 하단에 나타나도록 하고, 또한 각 플레이어의 남은 장애물의 갯수를 Wall:  $n$ 의 형태로 WHITE, BLACK의 Wall의 갯수가 서로 분간이 될 수 있도록 화면에 제시한다. 장애물 설치를 위해 “장애물” 버튼과, Back/Next버튼, 저장버튼, 기권버튼 등이 배치되어야 한다.
- **1인용 게임모드에서 말의 이동:** 현재 말의 위치에서 이동이 가능한 칸을 “열은 회색 circle”로 마킹하도록 한다. 사용자가 이동 가능한 칸을 마우스로 클릭하면 사용자의 말이 해당 칸으로 이동하고 턴이 종료된다. 상대방 말과 마주보는 경우 상대방을 뛰어넘을 수 있도록 해야 함을 유의하라.
- **1인용 게임모드에서 장애물 설치:** 장애물 설치를 위해 “장애물” 버튼을 화면에 구성하고, 이를 클릭하면 장애물 설치 모드가 되어, 사용자로부터 마우스 클릭을 입력 받는다. 보드에 마우스를 가져가면 설치될 장애물이 “열은 blue“으로 보드의 2칸에 걸쳐 표시되도록 하며, 다른 곳에 마우스를 이동하면 해당 위치에 맞는 2칸에서 다시 열은 blue의 장애물이 표시되도록 한다). 최종적으로 사용자가 보드에 마우스를 클릭하면 “열은 blue“로 표시된 장애물의 색깔은 “노랑색”으로 바뀌게 되고, 장애물은 고정되며, 턴이 종료된다. 만약, 현재 플레이어가 더이상 남은 장애물이 없으면 “장애물” 버튼은 자동 비활성화가 된다.
- **1인용 게임모드에서 Back/Next기능:** 1인용 게임모드에서 화면에 Back버튼 및 Next버튼을 추가로 구성한다. 플레이어 차례에 Back버튼을 클릭하면 이전 자신 차례의 게임 상태로 복귀된다. Back버튼은 무제한 클릭할 수 있으며,  $N$ 번 클릭하면  $N$ 번 이전의 자신의 차례의 게임 상태로 복귀 된다. 게임의

처음 상태로 가면 Back버튼은 비활성화가 되어야 한다. Next기능은 처음에 비활성화되어 있다가 플레이어가 Back버튼을 누르면 활성화로 바뀌게 된다. Back기능과 반대로 이미 Back된 상태에서 다시 해당 턴을 “회복”하는 기능이다. 다시 게임의 최근 상태가 되면 Next버튼은 비활성화가 되어야 한다.

- **2인용 게임모드:** 2인용 게임에서는 단일 컴퓨터상에서 2인용 게임을 플레이 할 수 있도록 하고, 사용자 입력을 받아 흰색말 (또는 벽), 검은말 (또는 벽) 을 . 이때, Back버튼 및 Next버튼이 디폴트로 비활성상태로 둔다.
- **2인용 네트워크 (게임생성):** 현재 플레이어의 컴퓨터가 서버가 되며, 게임 대기상태 윈도우가 새로 나타나, 상대방의 게임참여를 기다린다. 상대방이 게임을 참여하게 되면, 바로 게임을 시작한다 (게임을 위한 포트는 임의로 지정한다). 게임이 시작되면 게임모드로 전환되고, 화면은 1인용 게임모드와 같다.
- **2인용 네트워크 (게임참여):** 현재 플레이어의 컴퓨터가 클라이언트가 되고, 메뉴를 선택하면 게임대기상태 윈도우가 실행되며, 이 윈도우는 해당 서버의 IP를 입력하는 단일라인 텍스트 박스와, 접속 버튼, 게임 접속 상태의 로그를 보여주는 다중라인 텍스트 박스로 구성된다. 사용자의 IP입력후 접속버튼을 누르면 접속버튼은 임시로 비활성상태가 되면서 게임에 참여를 시도하고 다중라인 텍스트 박스에서 네트워크 접속 진행상태가 나타난다. 만약 해당 서버의 IP의 없거나 네트워크 에러가 발생시 에러메시지를 출력하고, 접속버튼은 다시 활성화가 되고 사용자의 IP변경 등 수정입력을 기다린다. 성공적으로 접속되면 게임모드로 전환되고, 화면은 1인용 게임모드와 같다.
- **2인용 네트워크 클라이언트-서버 프로토콜 (Hint):** 위의 2인용 네트워크는 TCP/IP상에서 구동되도록 하고,이를 위해 적절한 프로토콜이 정의가 되어야 한다. 클라이언트-서버 프로토콜은 최소한으로 하되, 유한 상태 오토마타 수준으로 미리 설계가 되어야 한다. 서버 및 클라이언트 각 상태를 정의하고, 단순화를 위해 메시지는 텍스트 문자열로 정의하라. 추가로, 서버와 클라이언트 각 상태마다 메시지 종류에 따라 다음 상태가 어떻게 바뀌는지도 미리 정의해야 한다.

메시지는 종류 (type) 및 인자 (arguments)로 구성시키는것이 일반적이다. 예를 들어, 게임모드에서 메시지 “move white 3 4”는 WHITE말을 (3,4)로 이동하라는 의미로 사용할 수 있고, “put wall 5 2 5 3”은 장애물을 위치 (5,2)와 (5,3)에 배치하라는 의미로 사용할 수 있다. 각 메시지의 구성요소는 해당 프로토콜 정의의 맞게 설계되어야 한다. 또한, 사용자와 입력과 관계없이 연결 및 동기 (synchronization)과 관련된 내부 메시지가 별도로 필요할 수 있다.

- **게임중 Save/Load기능:** Save/Load기능은 게임이 종료되기전에 플레이어 차례에서 화면의 Save버튼을 클릭하면 현재 게임 상태를 저장하여 이후에 다시 로딩해서 게임을 계속 진행할 수 있도록 하는 기능이다. 1) **Save기능:** 특정 디렉토리 (예: incomplete\_save) 내에 저장되어야 하며, 저장시 사용자로부터 파일이름을 입력받도록 하고, 디폴트 파일이름으로 현재날짜와 시간정보로 이전 파일과 중복되지 않도록 제시하고 이를 사용자가 수정할 수 있도록 해야 한다. 만약 Load기능을 통해 진행된 게임이라면 디폴트 파일이름은 현재 Loading된 파일과 동일한 이름으로 제시하면 된다.
- 2) **Load기능:** Save기능을 통해 저장된 게임상태 파일을 읽어들이 계속 진행할 수 있도록 하는 기능으로, 메뉴에서 Load버튼을 클릭하면 실행되도록 하라.

- **복기기능:** 모든 게임은 별도 디렉토리 (예: complete\_same)내에 자동으로 저장된다. 복기기능은 기 저장된 게임을 불러들여 이를 복기하는 것이다. 복기모드에서는 - **앞으로 (Previous)**, **재생(또는 중단)**, **뒤로 (Next)** - 세 가지 버튼으로 구성된다. 복기시 1초간격으로 자동으로 턴이 진행되도록 하고, 중단을 클릭하면 자동 턴 진행 모드가 해제되며, 해당 버튼 “재생”버튼이 된다. 그리고, “재생”버튼을 클릭하면 다시 1초간 자동 턴 진행모드로 바뀌고, 해당 버튼은 “종료”버튼으로 변경된다. 사용자가 Previous또는 Next를 누르면 턴이 각각 이전상태와 다음상태로 바로 이동이 된다. 또한 키패드의 왼쪽방향키와 오른쪽방향키가 각각 Previous와 Next기능을 수행하도록 하라. 게임 첫수의 경우 Previous버튼이 비활성화가 되며, 게임 마지막수가 끝난 경우 Next버튼이 비활성화가 되어야 한다.
- **Sound (효과음):** 말을 이동할때 그리고 벽을 놓을 때 각각 1초이내의 사운드파일을 효과음으로 재생하면 된다. 웹에서 적절한 파일을 찾아서 연결하면 된다.
- **환경설정:** 사용자 이름 변경, Save/Load 저장디렉토리 변경, 복기용 Game 디렉토리 변경, 효과음 크기, 장애물의 갯수, 저장된 게임 관리 등과 관련된 게임 상세 옵션 변경을 위한 메뉴이다.
- **1인용 인공지능:** 랜덤하게 말을 이동하거나 장애물을 배치하는 간단한 인공지능을 구현하라. 예를 들어, 이동과 장애물 설치 선택시, 이동할 확률이 장애물을 설치할 확률보다 높도록 하고, 이동 시에는 아래, 오른쪽, 왼쪽, 위쪽 순으로 높은 확률을, 장애물 설치시에는 세로 장애물과 가로 장애물이 동일한 확률을 가지도록 한다. 장애물은 무작위로 임의의 위치에 설치되도록 한다.

### 1.3 제출 내용 및 평가 방식

본 Quoridor과제 결과물로 필수적으로 제출해야 내용들은 다음과 같다.

- **코드 전체**
- **결과보고서:** 코드 설계 (클래스 계층도 포함), 구현 방법 및 실행 결과를 요약한 보고서

추가로, 본 Quoridor과제의 평가항목 및 배점은 다음과 같다.

- **게임의 기본기능 및 세부기능의 범위 및 완결성 (70점)**
- **구성 코드의 정확성, 간결성, Readability, 및 객체지향적 접근성 (20점)**
- **결과 보고서의 구체성 및 완결성 (10점)**
- **\*보너스 항목: 1인용 인공지능 심화 (추가 30점) - 인공지능의 정교함 및 완성도를 평가**