프로젝트 소개

기존 오목 AI 오픈소스의 학습 모델 구조 수정 및 기능 추가하였습니다.

기존 프로젝트에서 플레이 할 수 있는 판의 크기보다 더 확장하고,
다양한 크기에서 플레이 할 수 있도록 변경하였고,
AI 의 승률을 좀 더 올리기 위해서, 특수 상황에서는 알고리즘에 따라
수를 놓도록 수정하였습니다

판의 크기를 키울 수록 늘어나는 경우의 수로 학습 효율이 떨어지기 때문에,
일반적으로 렌주룰 오목 게임에서 사용하는 15x15 크기의 경우에는
자가학습만을 이용해서 모델을 학습 하지 않고, 오목 대회의 기보 데이터를 이용해서
학습하였습니다.

오목에 사용되는 룰을 렌주룰입니다.

프로젝트 깃허브 주소 : https://github.com/SSUhs/OmokPlay

팀원

김동헌 : https://github.com/znfnfns0365
변석찬 : https://github.com/JockvarBotham
서진원 : https://github.com/sjw9307
임현성 : https://github.com/SSUhs

개발 환경 및 라이브러리

- 개발 언어 : Python 3.8
- GUI : Pygame / Tkinter
- 실행 파일(exe) : Pyinstaller
- 딥러닝 라이브러리 : Tensorflow 2.9.2 / Keras / Theano
- 딥러닝 모델 학습 환경 : Google Colab

프로젝트 설명

판 크기에 따라 학습 방식이나 횟수가 다르기 때문에
같은 난이도여도 판 크기에 따라 체감 되는 난이도가 차이가 날 수 있습니다
(판 크기별로 모델을 따로 학습하였기 때문에 난이도마다 학습 횟수가 다릅니다)

< 15x15 크기 >
- 대회 기보 데이터 기반 학습
- 상 : 정책망,가치망 예측 기반 몬테카를로 트리탐색 + 정책망 예측 결과
  + 특수한 수에는 신경망 예측 대신 알고리즘 적용
- 중 : 정책망 예측 결과 + 특수한 수에는 신경망 예측 대신 알고리즘 적용
- 하 : 정책망 예측 결과만 사용

< 13x13 크기 >
- 대회 기보 데이터 없이 AI 끼리의 자가 학습만을 이용
- 상 : 18970 번 자가 대전으로 학습된 모델
- 중 : 14990 번 자가 대전으로 학습된 모델
- 하 : 12010 번 자가 대전으로 학습된 모델

< 11x11 크기 >
- 대회 기보 데이터 없이 AI 끼리의 자가 학습만을 이용
- 상 : 13140 번 자가 대전으로 학습된 모델
- 중 : 10260 번 자가 대전으로 학습된 모델
- 하 : 6240 번 자가 대전으로 학습된 모델

< 9x9 크기 >
- 대회 기보 데이터 없이 AI 끼리의 자가 학습만을 이용
- 상 : 11400 번 자가 대전으로 학습된 모델
- 중 : 6680 번 자가 대전으로 학습된 모델
- 하 : 3320 번 자가 대전으로 학습된 모델

설치 방법 및 사용법

게임을 진행할 수 있는 방법은
1. 실행 파일을 받아서 진행 (exe 파일)
2. Colab 에서 게임 실행 (웹으로 실행)
3. 프로젝트 전체를 받아서 실행

이렇게 세가지 방법이 있습니다.

실행 파일을 받아서 진행하는 경우, pysinstaller 로 만든 exe 파일로
바로 실행할 수 있습니다. 따라서 파이썬 개발환경 및 별도의 라이브러리를 설치 없이도
게임의 기능을 전부 진행할 수 있습니다.

Colab 으로 프로젝트를 실행하는 경우,
구글 Colab 사이트에 접속하여 실행할 수 있습니다.
어떠한 파일도 설치할 필요 없이 바로 게임을 시작 할 수 있고,
Colab 에서 고사양의 CPU, GPU 를 제공하기 때문에
트리 탐색 및 신경망을 통한 예측 과정에서 소요되는 시간이 적어서
가장 빠르게 게임을 진행할 수 있지만,
별도의 GUI 가 제공되지 않기 때문에 돌을 놓을 좌표를
숫자로 직접 입력 해야 되는 번거로움이 있습니다.
또한 Colab 에서 플레이 하는 경우,
게임 플레이 모드 중에서 'AI vs 플레이어'모드만 진행할 수 있습니다
(리플레이 / 플레이어 vs 플레이어 불가능)

프로젝트 전체를 받아서 실행하는 경우,
Tensorflow, Pygame 등의 라이브러리를 추가로 설치해야합니다.

< 실행 파일로 다운받아서 실행하는 경우 >

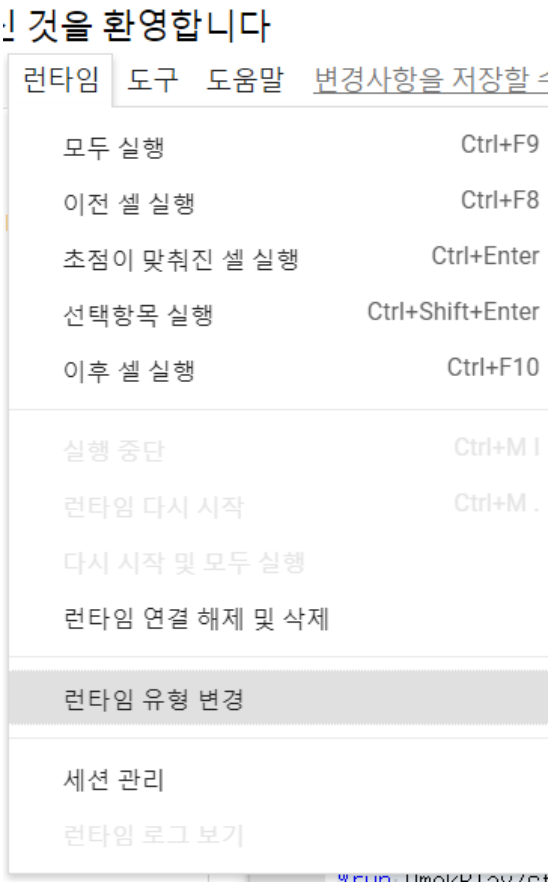
1.
https://github.com/SSUhs/OmokPlay#readme 에 나와 있는 실행 파일 링크로 다운로드

2.
압축 파일을 받아서 압축을 풀고 gui_main.exe 를 실행

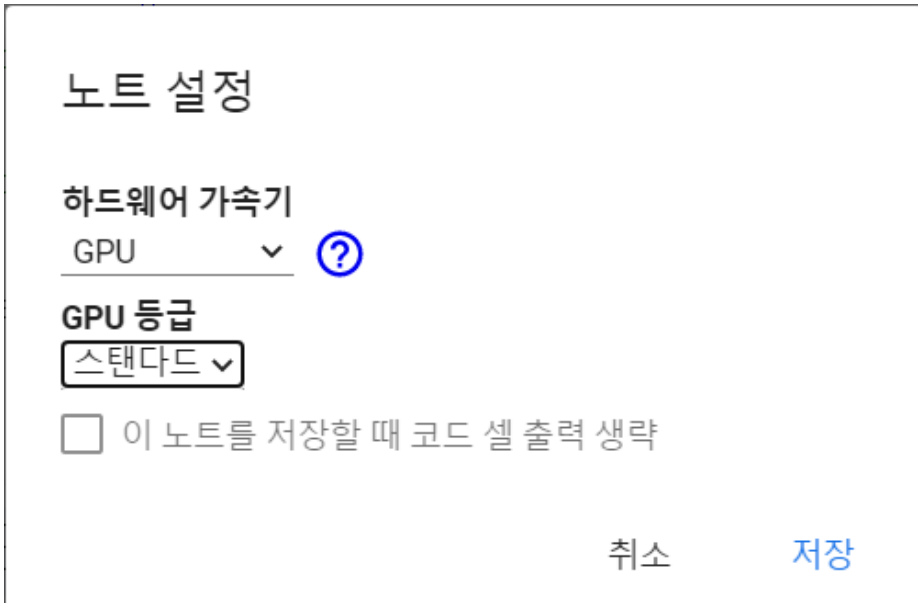
< Colab 으로 플레이 하는 경우 >

1. https://colab.research.google.com/ 접속 및 구글 로그인

2. (선택) 상단 메뉴에서



런타임 > 런타임 유형 변경 >

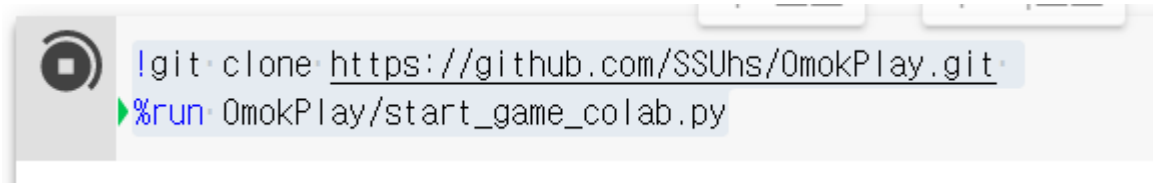


하드웨어 가속기를 GPU 로 바꾸고 저장

(GPU 가속을 설정 하지 않아도 상관 없지만, 15x15 크기에서 난이도 '상'으로 플레이시 AI 가 수를 선택하는 시간이 조금 더 오래 걸릴 수 있습니다)

3. 아래 코드 입력 후 실행

```
!git clone https://github.com/SSUhs/OmokPlay.git
%run OmokPlay/start_game_colab.py
```

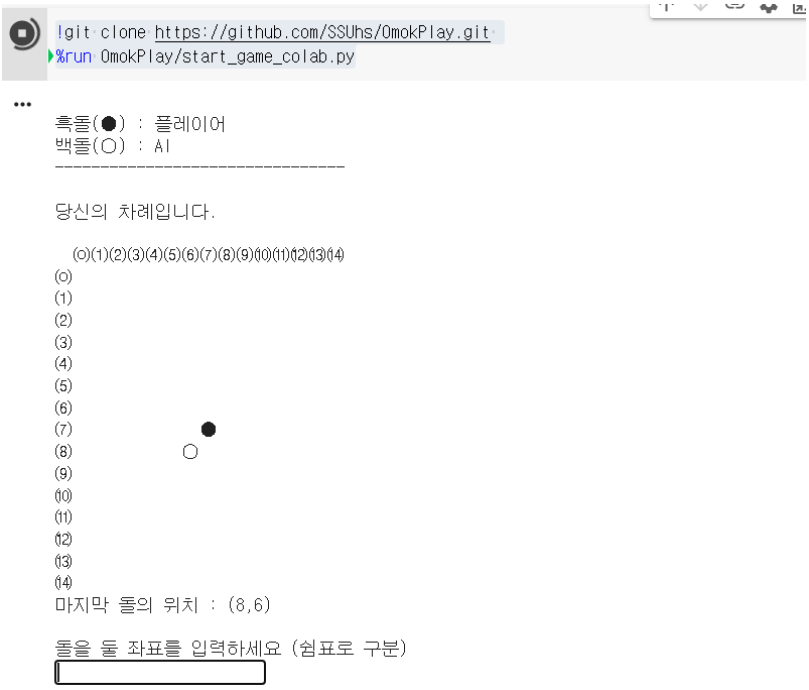


4. 난이도 선택 및 흑, 백 선택

```
fatal: destination path 'OmokPlay' already exists and is not an empty directory.
활성 Device : [PhysicalDevice(name='/physical_device:CPU:0', device_type='CPU')]

----------------------------------------
게임은 15x15 크기로 진행됩니다
난이도 : 상 / 중 / 하
중
자신이 선공(흑)인 경우에 0, 후공(백)인 경우에 1을 입력하세요.
0
```

5.



좌표를 숫자로 입력하여 진행
(y 좌표,x 좌표) 순서로 입력.
(0,0)부터 (14,14)

입력 예시 : 7,7

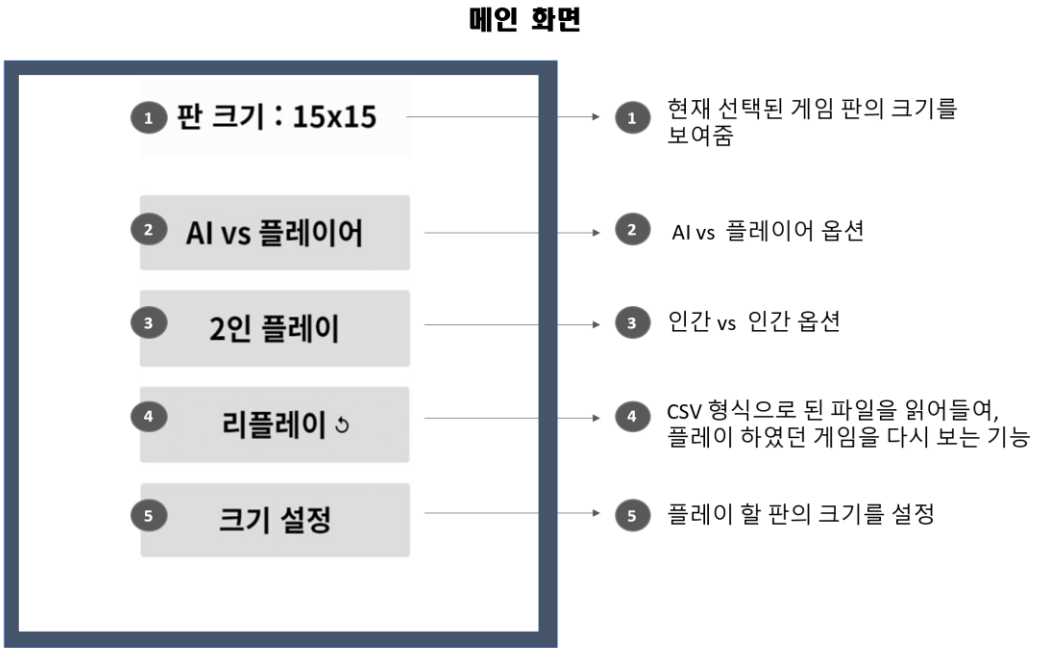
< 프로젝트로 실행 >

1. 깃허브에서 프로젝트 다운로드

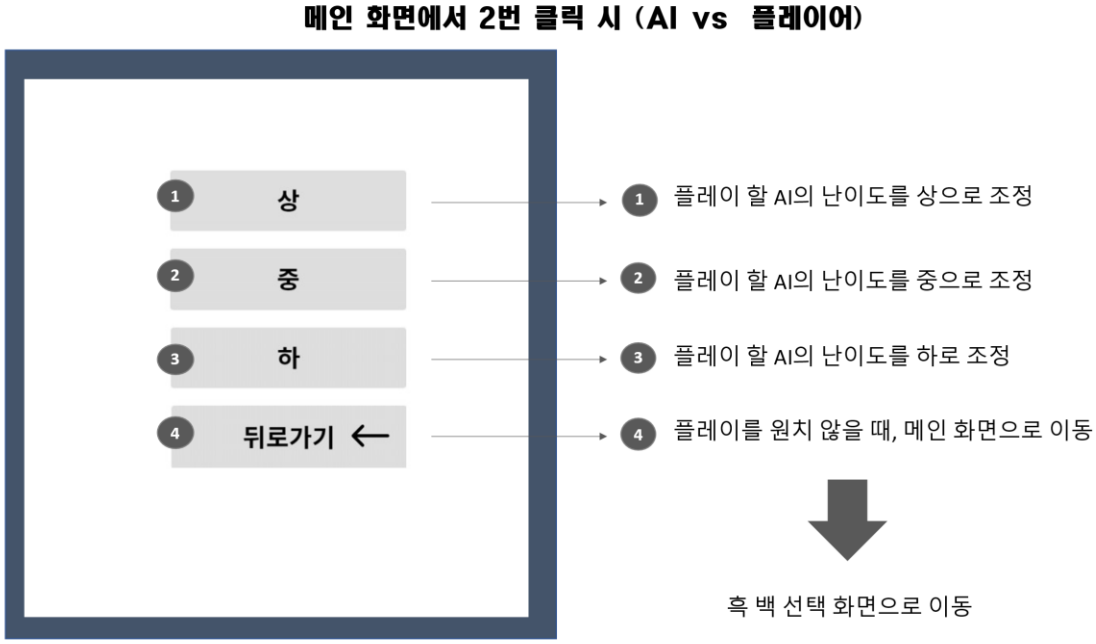
2. gui_main.py 의 메인 함수 실행

게임 플레이 방법

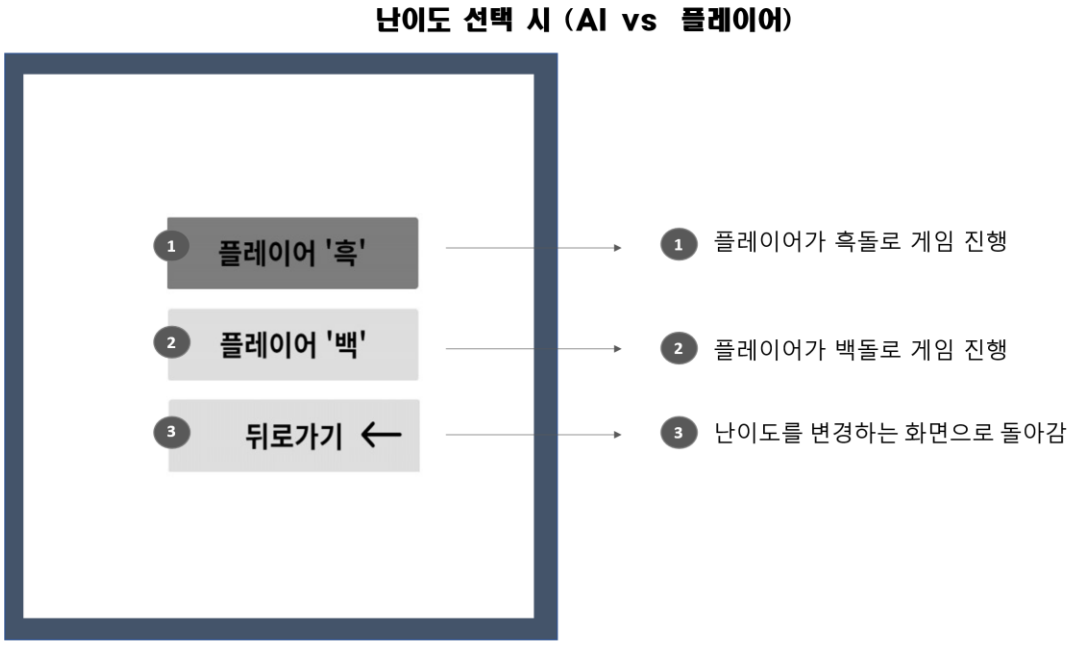
1) 메인 화면



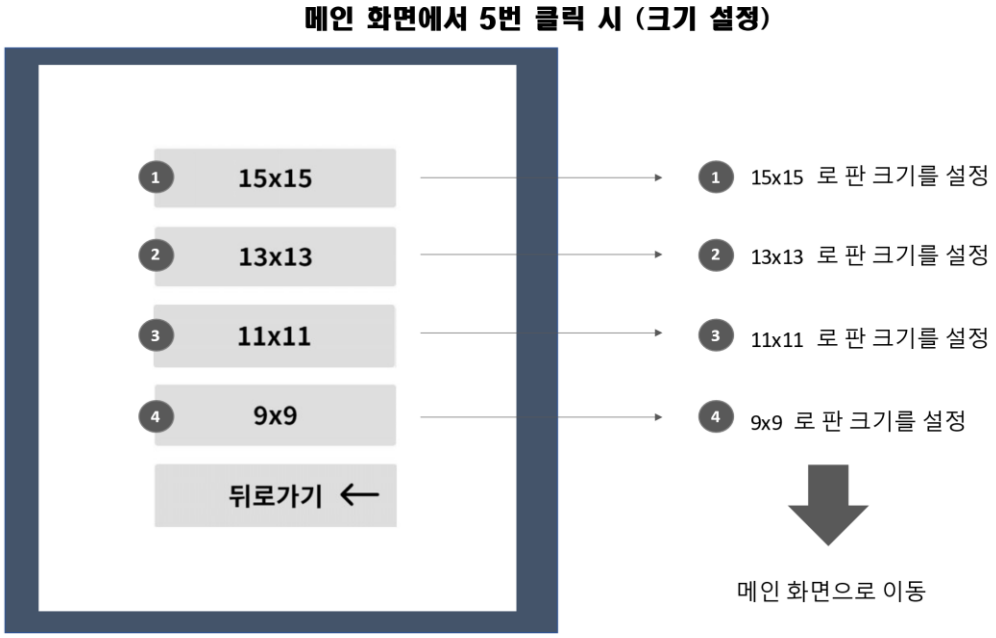
2) AI vs 플레이어 선택 시



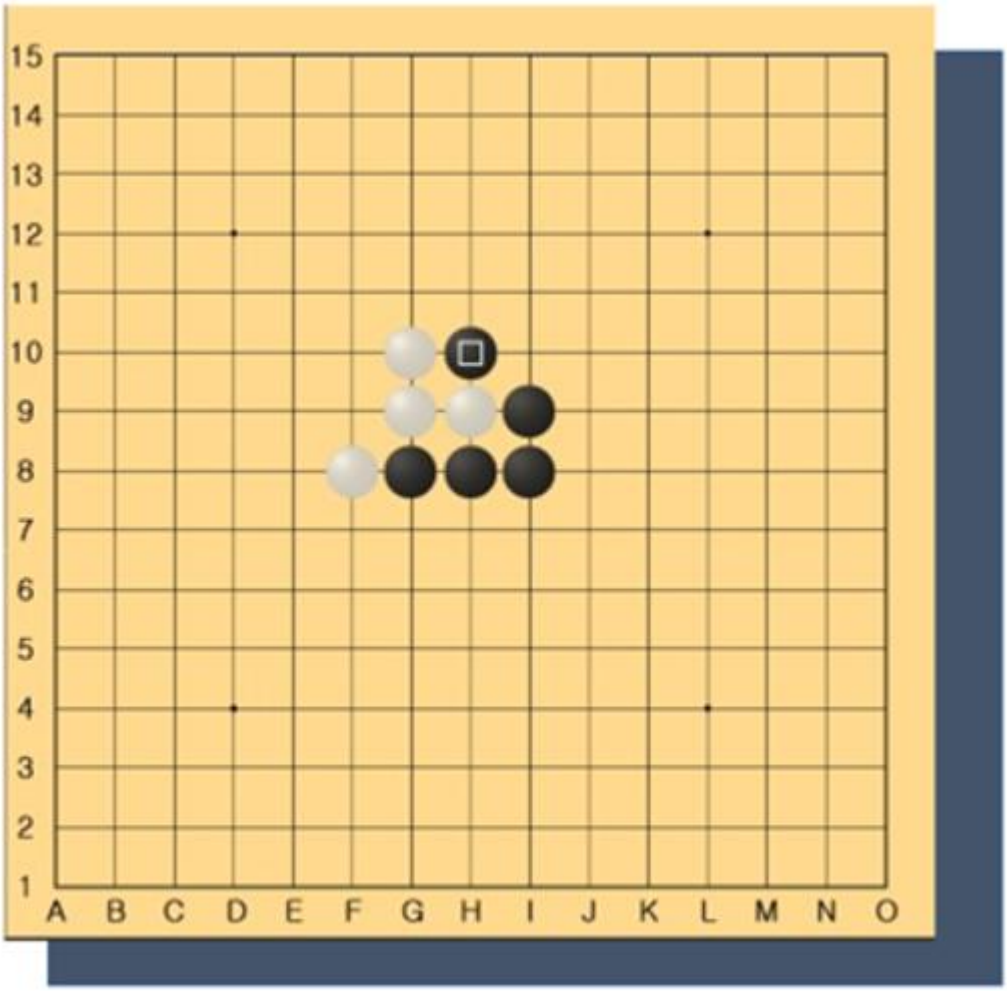
3) 난이도 선택 시



4) 크기 설정 선택 시



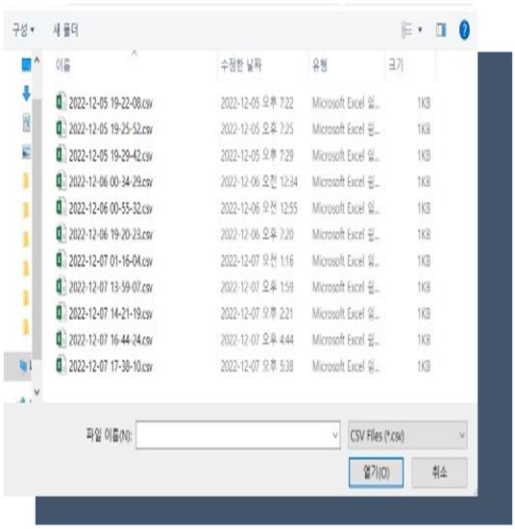
5) 게임 화면 (player vs player 또는 Ai vs Player)



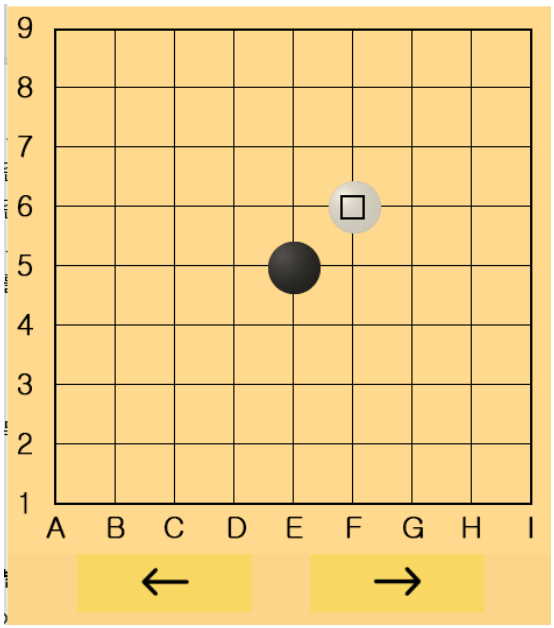
6) 리플레이 선택 시

메인 화면에서 4번을 클릭 시 (리플레이)

메인 화면에서 리플레이를 선택할 시,
csv 파일을 선택하는 화면이 나옴



(CSV 파일은 AI vs 플레이어 모드 또는 플레이어 vs 플레이어 모드에서 게임이 끝나면 자동 저장됩니다. Colab 으로 게임을 실행시 CSV 는 저장되지 않습니다)



csv 파일을 선택하면 해당 게임의 판 크기에 맞게 대국 화면이 실행됩니다.
아래의 이전, 다음 화살표로 이전 다음 상황을 확인할 수 있습니다

< 참고 URL >

https://github.com/yzhq97/AlphaGomokuZero/

https://www.moderndescartes.com/essays/deep_dive_mcts/

https://towardsdatascience.com/from-scratch-implementation-of-alphazero-for-connect4-f73d4554002a

https://github.com/ssof12/omok

https://github.com/kairess/omok-ai

기보 데이터 다운
https://gomocup.org/results/