

Project: Othello

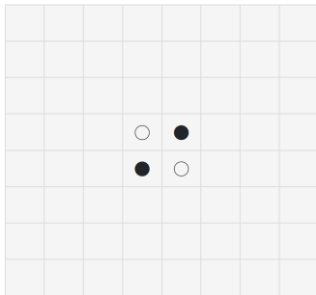
Due Date: 2025. 06. 19. (목) 23:59:59

[프로젝트 개요]

오델로(Othello)는 정사각형 보드 위에서 흑돌과 백돌을 번갈아 두며 진행되는 전략 보드게임으로, 자신의 돌로 상대의 돌을 양 끝에서 감싸 뒤집는 방식으로 진행되는 게임이다. 본 프로젝트 목적은 오델로 게임의 규칙과 전략 요소를 기반으로, 사용자와의 상호작용이 가능한 GUI 보드게임 애플리케이션을 직접 구현하는 데 있다. 이를 위해, C++ 기반의 GUI 프레임워크인 Qt를 사용한다. 프로젝트는 메인 화면, 게임 설정, 게임 진행, 리플레이 기능까지 포함한 오델로 게임 전체 흐름을 Qt 위젯 기반 인터페이스로 구현한다.

[게임 규칙 설명]

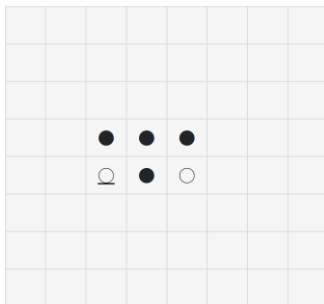
1. 시작



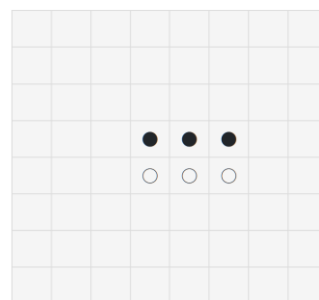
▲ 처음에는 정사각형 보드의 중앙 4칸에 흑, 백 돌을 교차로 배치해 놓고 게임을 시작한다.

2. 게임 진행

- 착수는 자신의 돌로 상대 돌을 가로, 세로, 대각선 중 최소 한 방향에서 감싸는 위치에만 가능하다.
- 돌을 두었을 때, 상대 돌이 자신의 돌 사이에 있으면 해당 돌은 자신의 돌로 뒤집힌다.
- 착수 이후에는 자동으로 턴이 전환되며, 다음 플레이어가 진행한다.
- 아무 돌도 뒤집히지 않는 위치에는 착수할 수 없다.

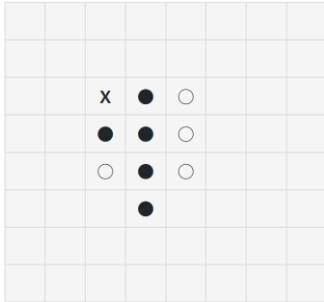


▲ 밑줄 표시처럼 돌을 두면,



▲ 이렇게 뒤집힌다.

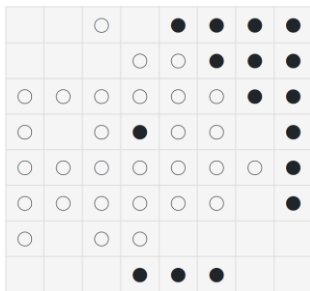
ex)



▲ 돌은 가로, 세로, 대각선 8방향으로 뒤집을 수 있으며, X친 곳에 흰색 돌을 놓게 된다
면, 가로, 세로, 대각선 방향으로 총 3개를 뒤집을 수 있다.

3. 패스

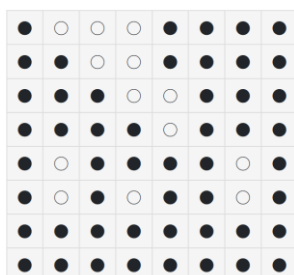
자신의 턴에 착수 가능한 위치가 하나도 없을 경우, 해당 턴은 자동으로 상대방에게 넘어가
며 이를 패스(Pass)라고 한다.



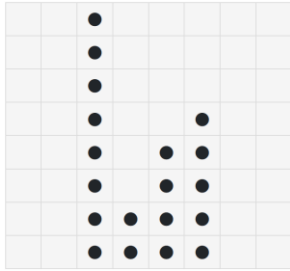
▲ 예를 들어, 백의 차례인데 착수 가능한 위치가 없는 상황에서는 패스(Pass)라고 하여 백을
건너뛰고 흑의 순서로 넘어가게 된다. 단, 착수 가능한 자리가 하나라도 있다면 반드시 착수
해야 하며, 의도적인 패스는 허용되지 않는다.

4. 게임 종료

- 흑과 백 모두 돌을 놓을 곳이 없으면 게임이 끝나고, 돌이 더 많은 쪽이 승리하게 된다.
흑백돌이 놓이지 않는 빈칸이 있더라도 마찬가지로, 흑과 백 모두가 착수 가능한 위치가
없을 경우이므로 게임은 종료된다. 양측 돌의 수가 똑같으면 무승부가 된다.



- 상대방의 돌을 전부 뒤집어 **한쪽 돌만 남게 되는 경우**에도 게임은 즉시 종료된다.



5. 변형 규칙

- 기본 8×8 외에도 6×6, 10×10 등 짝수 크기의 보드를 사용할 수 있다.
- 일부 칸을 **장애물로 지정해 사용할 수 없도록 설정**할 수 있다.

[필수 구현 사항]

1. 화면 구성
 - 메인 화면
 - 게임 설정 화면
 - 게임 화면
 - 리플레이 화면
2. 기본 게임 규칙 구현
 - 돌 놓기 시 착수 가능 여부 판단
 - 상대 돌 뒤집기 처리
 - 착수 불가 시 턴 패스
 - 게임 종료 조건 및 승패 판정
3. 게임 설정 기능
 - 보드 크기 설정
 - 장애물 개수 및 위치 설정
 - 선공 플레이어 설정
4. 게임 상태 표시
 - 현재 차례 표시
 - 각 플레이어의 돌 개수 표시
 - 착수 가능한 위치 표시
5. 리플레이 기능
 - 각 턴별 게임보드 저장
 - 슬라이더 또는 버튼을 통한 턴 이동
 - 리플레이 화면에서, 해당 턴에 돌을 둔 위치 시각화
6. 프로그램 안정성 및 품질

[필수 구현 사항별 요구조건]

구현해야 할 모든 기능에 대해 아래에 명시된 요구조건을 충실히 충족하는 한, 레이아웃이나 디자인 등 세부 구현 방식은 수강생의 자율적이고 창의적인 판단에 따라 자유롭게 구성해도 된다.

1. 화면 구성

프로그램은 총 4개의 주요 화면으로 구성되며, 각 화면은 사용자의 진행 상황에 따라 전환된다. 각 화면은 명확한 기능과 버튼 동작을 가져야 하며, 사용자는 자연스럽게 직관적으로 흐름을 따라갈 수 있어야 한다.

■ 메인 화면

- 사용자가 프로그램을 처음 실행했을 때 보게 되는 화면이다.
- 새 게임 시작, 이어하기, 리플레이, 종료 중 하나를 선택할 수 있도록 버튼이 표시되어야 한다.
- “새 게임” 버튼 클릭 시 게임 설정 화면이 팝업 형태로 출력되어야 한다.
- “이어하기” 버튼은 게임이 종료되지 않고 사용자가 '메인 화면' 버튼을 눌러 나간 경우에만 동작해야 하며, 그 외에는 비활성화되거나 무반응이어야 한다.
- “리플레이” 버튼은 게임 종료 후 사용자가 리플레이 저장을 선택한 경우에만 동작해야 하며, 그 외에는 비활성화되거나 무반응이어야 한다.
- “게임 종료” 버튼 클릭 시 프로그램이 정상 종료되어야 한다.
- 메인 화면의 적절한 위치에 라벨 등을 활용하여 과제 수행을 한 수강생 본인의 학번과 이름을 표시해야 한다

■ 게임 설정 화면

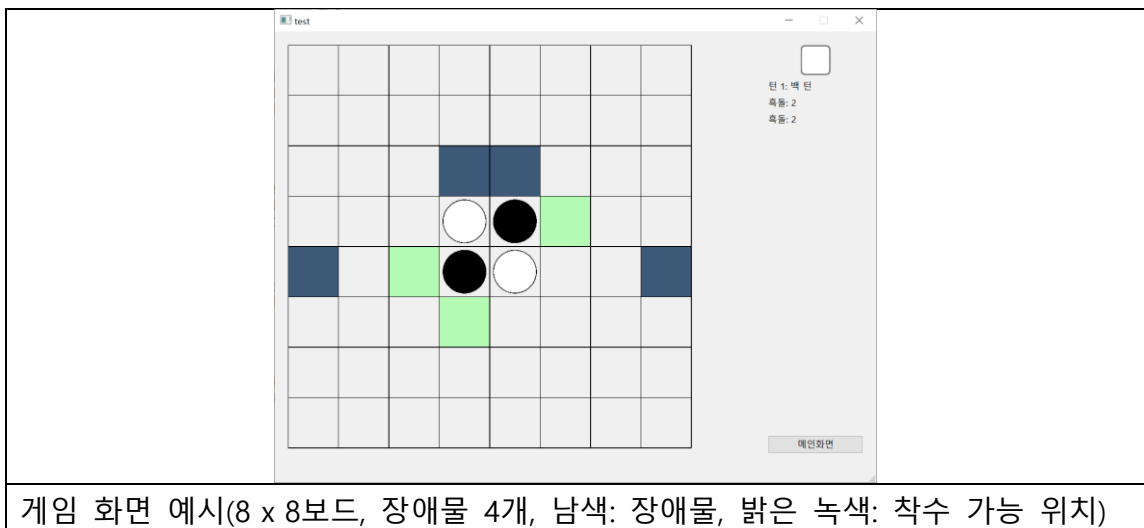
- 이 화면은 QDialog 기반의 모달 대화상자 형식으로 구성되며, 보드 크기 (6x6 ~ 12x12, 짝수), 장애물 개수(0~16), 선공 플레이어를 결정할 수 있는 UI요소를 제공해야 하며, “확인”, “취소” 버튼이 포함되어야 한다.
- 확인 버튼 클릭 시 설정이 저장되며, 게임 화면으로 전환되어야 한다.
- 취소 버튼 클릭 시 메인 화면으로 복귀하고, 설정은 반영되지 않아야 한다.

■ 게임 화면

- 실제 모델로 게임이 진행되는 주 화면으로, **보드 UI**, **상태 표시 창**, “**메인 화면**” 버튼의 요소로 구성되어야 한다.
- **보드 UI**는 설정한 보드 크기(6x6 ~ 12x12)에 따라 정확히 구성되어야 하며, 각 칸은 경계선으로 명확하게 구분되어야 한다.
- 각 칸에는 **돌**, **장애물**, **착수 가능 위치** 등을 시각적으로 구분하여 표시해야 하며, 착수 가능 위치와 장애물은 별도 색상으로 강조되어야 한다.
- **상태 표시 창**은 보드의 우측 또는 하단에 배치되며, 다음 정보가 항상 실시

간으로 표시되어야 한다.

- 현재 턴 플레이어
 - 보드 내 흑/백 돌 개수
 - 현재 턴의 돌 색상(아이콘, 라벨의 배경색 등으로 강조)
- 사용자의 마우스 클릭을 통해 착수(돌 두기), 뒤집기, 턴 넘기기 등 게임 진행이 이루어지며, 착수 가능 위치를 클릭했을 때만 착수가 가능하도록 해야 한다.
 - **“메인 화면” 버튼**은 하단 또는 우측 하단 등 적절한 위치에 배치되어야 하며, 클릭 시 메인 화면으로 전환되어야 한다.
 - 사용자가 “메인 화면” 버튼을 눌러 메인 화면으로 나갔다가 다시 “이어하기” 버튼을 통해 복귀했을 경우, 직전에 나갔던 시점의 상태가 정확히 복원되어야 하며, 이후 정상적인 게임 진행이 가능해야 한다.



게임 화면 예시(8 x 8보드, 장애물 4개, 남색: 장애물, 밝은 녹색: 착수 가능 위치)

■ 리플레이 화면

- 리플레이 화면은 이전에 저장된 게임을 턴 단위로 다시 확인할 수 있는 전용 화면이다.
- 게임 화면과 유사하게, **보드 UI**, **상태 표시 창**, **“메인 화면” 버튼**이 있어야 하며 **슬라이더 또는 이전/다음 버튼**을 통해 턴을 이동할 수 있어야 한다.
- **보드 UI**는 턴이 이동될 때마다 해당 턴의 돌 배치 상태를 정확히 반영해야 하며, 착수 가능한 위치는 표시되지 않아야 하고, 해당 턴에 실제 착수한 위치는 시각적으로 강조되어야 한다.
- **상태 표시 창**에는 해당 턴의 현재 플레이어 정보와 흑/백 돌 개수가 표시되어야 하며, 턴 이동 시 이 정보들도 함께 갱신되어야 한다.
- **“메인 화면”버튼**을 눌러 메인 화면으로 복귀할 수 있어야 한다.

2. 기본 게임 규칙 구현

오텔로 게임의 핵심 규칙은 돌 착수, 뒤집기, 턴 진행, 종료 및 승패 판단을 포함한다. 각 규칙은 명확하고 일관되게 구현되어야 하며, 게임 로직 전반에 적용되어야 한다.

- 돌 놓기 시 착수 가능 여부 판단
 - 사용자가 돌을 두려는 위치가 유효한 착수 위치인지 판단해야 하며, 해당 위치에서 상대 돌을 최소 하나 이상 뒤집을 수 있는 경우에만 착수가 가능해야 한다.
 - 착수 가능 여부는 8방향(상, 하, 좌, 우, 대각선)을 모두 검사하여 판단하며, 이 조건을 만족하지 않는 위치는 클릭해도 착수가 일어나지 않는다.
 - 매 턴마다 현재 플레이어에 따라, 플레이어가 둘 수 있는 위치를 사전에 계산하고, 이를 시각적으로 표시해야 한다.
- 상대 돌 뒤집기 처리
 - 착수 가능한 위치에 돌을 착수했을 경우, 해당 위치를 중심으로 연속된 상대 돌 뒤에 본인의 돌이 있는 방향을 찾아, 그 사이의 상대 돌을 모두 뒤집어야 한다.
 - 뒤집기는 8방향 모두에 대해 적용된다.
 - 착수 및 뒤집기 처리가 완료된 이후에는 턴이 전환되어야 한다.
- 착수 불가 시 턴 패스
 - 현재 플레이어가 착수할 수 있는 위치가 하나도 없는 경우, 자동으로 턴이 상대방에게 넘어가야 한다.
 - 턴이 넘어갔다는 사실은 사용자에게 팝업 메시지 또는 상태 표시창을 통해 안내되어야 한다.
 - 양쪽 플레이어 모두 착수 가능한 자리가 없을 경우, 게임이 종료되어야 한다.
- 게임 종료 조건 및 승패 판정
 - 게임 종료는 다음 두 조건 중 하나가 만족될 경우 발생해야 한다.
 - 보드가 전부 채워졌을 때
 - 양 플레이어 모두 더 이상 둘 수 있는 자리가 없을 때
 - 게임이 종료되면, 보드 위에 있는 흑돌과 백돌의 개수를 계산하여 승패 또는 무승부를 판정해야 한다.
 - 최종 결과는 사용자에게 팝업 메시지 또는 상태 표시 창을 통해 명확하게 안내되어야 하며, 그 후 리플레이 저장 여부를 묻는 절차가 이루어져야 한다.

- 리플레이 저장 여부를 사용자가 선택하면, 메인 화면으로 전환되어야 한다.

3. 게임 설정 기능

게임을 시작하기 전, 사용자가 원하는 게임 환경을 설정할 수 있도록 설정 창을 띄워야 한다. 이 창은 다음 세 가지 항목에 대한 입력 또는 선택을 받을 수 있어야 하며, 사용자가 설정한 내용은 새 게임을 시작할 때 즉시 반영되어야 한다.

- **보드 크기 설정**
 - 사용자는 여러 가지 보드 크기 중 하나를 선택할 수 있어야 하며, 선택 가능한 보드 크기는 최소 6칸부터 최대 12칸까지의 **짝수 정사각형 크기**여야 한다.
 - 선택된 보드 크기에 따라, 실제 게임 보드의 칸 수가 자동으로 설정되어야 하며, 이후 돌의 착수 처리와 뒤집기 등 모든 연산은 이 크기를 기준으로 동작해야 한다.
- **장애물 개수 및 위치 설정**
 - 사용자는 장애물을 몇 개 둘 것인지 직접 입력하거나 선택할 수 있어야 한다.
 - 장애물의 개수는 최소 0개부터 최대 16개까지 될 수 있다.
 - 장애물의 개수는 대칭 위치에 배치된 것을 포함한 개수이다.
 - 이때, 장애물은 게임 시작 시 보드 위에 자동으로 배치되며, 보드의 가운데 4칸(초기 돌이 배치되는 위치)과는 겹치지 않도록 해야 한다.
 - 장애물은 보드 좌우 대칭 위치에 동시에 배치되어야 하며, 이미 돌이나 장애물이 있는 위치는 피해서 배치되어야 한다.
- **선공 플레이어 설정**
 - 사용자에게 선공 플레이어(흑, 백, 무작위)를 선택할 수 있는 UI를 제공한다.
 - 선택된 플레이어가 게임 시작 시 첫 번째 턴을 진행하게 된다.
 - 설정된 정보는 게임 화면의 상태 창 및 로직에서 정확히 반영되어야 한다.

4. 게임 상태 표시

게임이 진행되는 동안, 현재 어떤 상황인지 사용자에게 시각적으로 실시간 제공해야 하며, 이를 위한 상태 표시창이 항상 보이도록 구성해야 한다. 이 상태 표시창에는 다음 정보들이 포함되어야 하며, 매 턴마다 자동으로 갱신되어야 한다.

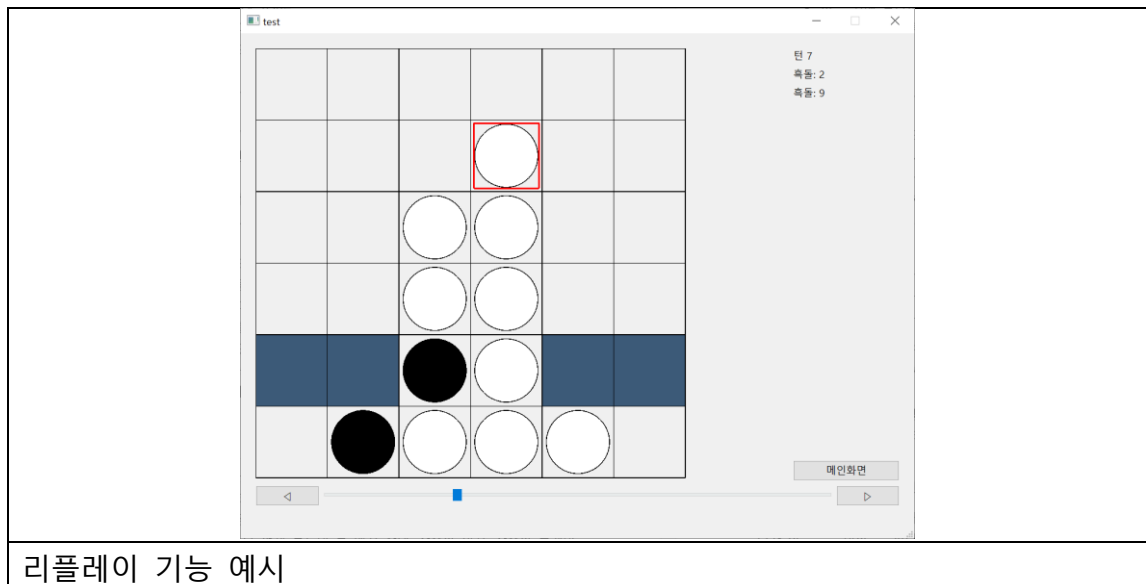
- **현재 플레이어의 차례 표시**
 - 지금 어떤 플레이어가 착수를 해야 하는지 분명하게 표시해야 하며, 돌 색상과 함께 텍스트 또는 배경 색으로 강조 표시되어야 한다.
 - 예시) 흑돌의 차례일 경우 '흑 턴'이라는 문구와 검정색 배경을, 백돌일

경우 '백 턴'이라는 문구와 흰색 배경을 표시

- 각 플레이어의 돌 개수 표시
 - 게임 보드 위에 현재 놓여 있는 흑돌과 백돌의 개수를 계산하여 각각 몇 개씩 있는지 숫자로 표시해야 한다.
 - 이 정보는 돌을 착수하거나 상대 돌을 뒤집을 때마다 자동으로 갱신되어야 하며, 게임 종료 시 최종 결과 판단에도 활용된다.
- 착수 가능한 위치 표시
 - 현재 플레이어가 돌을 둘 수 있는 위치는 보드 위에서 눈에 띄게 색을 달리 하여 표시해야 한다.
 - 사용자는 이 표시를 보고 어디에 둘 수 있는지를 쉽게 판단할 수 있어야 하며, 표시가 없는 곳을 클릭했을 경우에는 아무 동작도 일어나지 않아야 한다.
 - 이 위치 표시 기능은 매 턴 자동으로 갱신되며, 상대 돌을 최소 하나 이상 뒤집을 수 있는 위치만 표시되어야 한다.

5. 리플레이 기능

게임이 종료된 후, 사용자가 원할 경우 게임을 처음부터 다시 재생할 수 있는 리플레이 기능을 제공해야 한다. 이를 위해 각 턴의 보드 상태와 착수 정보가 저장되어야 하며, 리플레이 전용 화면에서 이를 순차적으로 재생할 수 있어야 한다.



리플레이 기능 예시

- 턴별 보드 상태 저장
 - 게임 도중 각 턴마다 현재 보드의 돌 배치 상태, 착수한 위치, 착수한 플레이어, 그리고 해당 턴의 돌 개수를 별도로 저장해야 한다.

- 이 저장은 게임이 실제로 진행되는 도중 실시간으로 이루어져야 하며, 리플레이 기능을 위한 데이터로 활용된다.
- 게임 종료 시, 사용자가 리플레이 저장 여부를 선택할 수 있도록 해야 하며, 저장하지 않은 경우에는 리플레이 기능을 사용할 수 없어야 한다.
- 턴 이동 기능 제공
 - 리플레이 화면에서는 사용자가 이전 턴이나 다음 턴으로 자유롭게 이동할 수 있어야 한다.
 - 이를 위해 슬라이더 또는 버튼 형태의 사용자 인터페이스를 제공해야 하며, 해당 조작이 이루어질 때마다 보드 상태가 선택한 턴에 맞게 즉시 반영되어야 한다.
 - 리플레이 중에는 착수 가능한 위치 표시나 사용자 입력은 차단되어야 하며, 돌 배치 상태만 보여주는 관람 모드로 동작해야 한다.
- 착수 위치 강조 표시
 - 리플레이 화면에서는 각 턴에 착수한 위치가 시각적으로 강조되어 표시되어야 하며, 예를 들어 빨간 테두리나 굵은 선으로 표시할 수 있다.
 - 이 강조는 오직 해당 턴에서 착수한 위치에만 적용되어야 하며, 이전 턴의 강조는 자동으로 사라져야 한다.
 - 이를 통해 사용자가 게임의 흐름을 쉽게 추적할 수 있도록 시각적 연속성을 제공해야 한다.

6. 프로그램 안정성 및 품질

- 모든 사용자 입력 및 게임 실행 과정에서 프로그램은 예외 없이 정상적으로 동작해야 하며, 비정상 종료가 발생해서는 안 된다.
- 게임을 여러 번 실행하거나 리플레이를 반복해도 성능 저하가 없어야 한다.
- 동적으로 할당된 자원은 적절히 해제되어야 하며, 자원의 누수 없이 종료되어야 한다.

[선택 구현 사항] (추가 점수 항목)

다음 기능들은 필수 구현 사항은 아니지만, 구현한 경우 완성도와 기능 확장 측면에서 추가 점수를 부여할 수 있다. 구현 시 사용자 입장에서 명확하게 동작해야 하며, 기존 기능과 충돌 없이 통합되어야 한다.

- 제한 시간 모드(체스의 증초/피셔 방식 적용) (+10점)
 - 각 플레이어는 게임 시작 시 총 제한 시간(예: 300초, 600초)을 할당 받는다.
 - 매 착수마다 추가 시간이 부여되어(예: +5초, +10초) 남은 시간에 누적된다.
 - 제한 시간 모드가 활성화된 경우, 게임 화면의 상태 표시창에 각 플레이어의 남은 시간이 실시간으로 표시되어야 하며, 초 단위로 감소해야 한다.
 - 한 플레이어의 시간이 모두 소진될 경우, 해당 플레이어는 즉시 패배 처리된다.
 - 제한 시간 모드는 게임 설정 화면에서 모드 활성화 여부 및 시간 설정 값(총 시간, 증가 시간)을 사용자가 직접 지정할 수 있어야 한다.
 - 제한 시간 모드가 비활성화 된 경우에는 일반적인 무제한 플레이가 가능해야 한다.
- 되돌리기(Undo) 기능 (+5점)
 - 플레이어는 자신의 이전 턴 상태로 되돌아갈 수 있어야 한다.
 - 되돌리기 기능을 사용하면 보드 상태, 턴 정보, 돌 개수 등 모든 게임 상태가 이전 턴으로 정확히 복원되어야 하며, 화면에도 동일하게 반영되어야 한다.
 - 되돌리기 버튼은 게임 화면에 적절히 배치되어야 하며, 클릭 시 즉시 동작해야 한다.
 - 리플레이 화면에서는 되돌리기 기능이 비활성화 되어야 한다.
- 게임 저장 기능(게임 상태 파일 출력 및 불러오기) (+5점)
 - 게임 도중 '저장'버튼을 눌러 현재 게임 상태를 파일 형태로 저장할 수 있어야 한다.
 - 저장된 정보에는 보드 상태, 턴, 돌 개수, 장애물 위치 등 게임 진행에 필요한 모든 정보가 포함되어야 한다.
 - 저장한 게임은 프로그램을 종료한 후에도 파일을 통해 복원이 가능해야 하며, 이어서 플레이할 수 있어야 한다.
 - 이 기능을 구현할 경우, 메인 화면에 "불러오기"버튼을 추가하여 저장된 게임을 불러온 뒤 게임을 이어서 진행할 수 있어야 한다.

[보고서 작성]

- 보고서에는 수강생이 구현한 “필수 구현 사항” 및 “선택 구현 사항”을 각각 명확히 구분하여 기술하고, 각 기능이 실제 화면에서 어떻게 동작하는지를 결과 화면의 스크린샷과 함께 설명해야 한다. 이는 채점자가 해당 기능의 구현 여부와 완성도를 객관적으로 판단하는 데 도움이 되어야 한다.
- 실제로 구현을 한 사항이라도 보고서에 명시하지 않으면 감점될 수 있으므로, 모든 구현 내용을 빠짐없이 작성해야 한다.
- 영문 또는 한글로 작성한다.
- 반드시 PDF로 제출한다.
- 표지를 제외하고 4페이지 이상으로 작성한다.
- 소스코드는 보고서에 포함하지 않는다.

[평가 기준]

평가 항목	평가 내용	배점
기본 게임 규칙	오델로의 착수 규칙, 돌 뒤집기, 턴 전환, 착수 불가 시 자동 패스, 게임 종료 조건 등이 정확히 구현되었는가	20
설정 기능 구현	보드 크기, 장애물 개수, 선공 설정이 정확히 반영되어 보드와 초기 상태가 구성되는가	10
보드 시각화 및 반응 처리	보드의 돌, 장애물, 착수 가능 위치가 시각적으로 명확히 표현되고, 마우스 클릭 시 정상적으로 반응하는가	15
상태 정보 갱신	매 턴마다 돌 개수, 현재 턴 정보 등이 정확하게 계산되고 상태창에 반영되는가	15
리플레이 기능 구현	각 턴의 게임 상태가 저장되고, 슬라이더 및 버튼을 통해 해당 턴의 상태를 정확히 불러올 수 있는가	20
리플레이 시각화	리플레이 모드에서 착수 가능 위치를 표시하지 않으며, 각 턴의 착수 위치가 강조되어 흐름을 파악할 수 있는가	10
메인화면 / 이어하기 기능	게임 중 메인화면으로 이동 후 다시 이어하기 버튼을 눌렀을 때 게임 상태가 정상 복원되는가	5
프로그램 안정성 및 품질	게임 진행 중 에러 없이 작동하며, 예외 상황(무효 클릭, 종료 등)에 대해 올바르게 대응하는가	5
총점		100
추가 구현 항목		추가 점수
제한 시간 모드		10
되돌리기 기능		5
게임 저장 기능		5

[제출 기한 및 제출 방법]

제출 기한: 2025년 6월 19일 (목요일) 23:59:59 까지 제출

제출 방법: **build** 폴더를 제외한 프로젝트 파일 및 보고서를 포함하여 zip 파일로 **Klas** 과제 제출

- 객체지향프로그래밍 설계/실습 모두 수강할 경우, 실습 수업에 제출
- 객체지향프로그래밍 설계만 수강할 경우, 설계 수업에 제출
- 객체지향프로그래밍 실습만 수강할 경우, 실습 수업에 제출

제출 형식:

- 설계반_실습반_학번_이름.zip
- 예) 설계 수강, 실습 A반: 1_A_학번_이름.zip
- 예) 설계 수강, 실습 미수강: 2_0_학번_이름.zip
- 예) 설계 미 수강, 실습 C반: 0_C_학번_이름.zip
- 채점 시 코드를 수정해야 하는 일이 없도록 한다.
- 과제 수정하여 업로드 시 버전 명시: 설계반_실습반_학번_이름_verX.zip