

OOP-Term Project

18정주현, 20최동현, 18최원준

목차

- 제약 사항 체크
- 평가 기준 체크
 - 기능성
 - 안전성
 - 디자인 패턴
 - 보고서 (UML)
- 게임 시나리오 진행

제약사항

- 구현한 기능에 대해 Gtest 진행
- 게임 보드의 크기는 사용자 입력을 통해 결정
- Makefile, Lint 진행

Gtest 내용

게임 진행 및 기능에
관여하는
주요 함수들에 대해
Gtest를 진행함

```
151 TEST(0thelloTest, CheckValidTest) {
152     //기본 설정
153     Judge* judge_test = Judge::GetInstance();
154     judge_test -> GetBoard() -> SetSize(8);
155     std::vector<std::vector<char> > matrix_now = judge_test -> GetBoard() -> Getma
156
157     //돌을 놓은 상태
158     judge_test -> GetBoard() -> PutStone(3, 5, 'B');
159     judge_test -> ModStoneStatus(3, 5, 'B');
160     judge_test -> ChangeTurn('B');
161     judge_test -> GetBoard() -> RestoreBoard();
162
163     bool to = false;
164     bool from = judge_test -> CheckValid(0, 0, 'B');
165     EXPECT_EQ(to, from);
166 }
167
168 //candidate 정상 실행 여부
169 TEST(0thelloTest, CandidateTest) {
170     //기본 설정
171     Judge* judge_test = Judge::GetInstance();
172     judge_test -> GetBoard() -> SetSize(8);
173 }
```

Gtest 진행 사진

```
Othello — -zsh — 80x42
g++ -isystem /Users/wanpile/Documents/GitHub/Othello/googletest/googletest/include -std=c++11 --std=c++17 othello_test.o Judge.o Board.o Player.o gtest_main.a -o test
[wanpile@MacBookPro Othello % ./test
Running main() from /Users/wanpile/Documents/GitHub/Othello/googletest/googletest/src/gtest_main.cc
[=====] Running 12 tests from 1 test suite.
[-----] Global test environment set-up.
[-----] 12 tests from OthelloTest
[ RUN      ] OthelloTest.StartNameTest
[       OK ] OthelloTest.StartNameTest (0 ms)
[ RUN      ] OthelloTest.StartColorTest
[       OK ] OthelloTest.StartColorTest (0 ms)
[ RUN      ] OthelloTest.StartSetBoardTest
[       OK ] OthelloTest.StartSetBoardTest (0 ms)
[ RUN      ] OthelloTest.StartPutStoneTest
[       OK ] OthelloTest.StartPutStoneTest (0 ms)
[ RUN      ] OthelloTest.ScoreTest1
[       OK ] OthelloTest.ScoreTest1 (0 ms)
[ RUN      ] OthelloTest.ScoreTest2
[       OK ] OthelloTest.ScoreTest2 (0 ms)
[ RUN      ] OthelloTest.CheckValidTest
[       OK ] OthelloTest.CheckValidTest (0 ms)
[ RUN      ] OthelloTest.CandidateTest
[       OK ] OthelloTest.CandidateTest (0 ms)
[ RUN      ] OthelloTest.IsFullBoardTest
[       OK ] OthelloTest.IsFullBoardTest (0 ms)
[ RUN      ] OthelloTest.IsGameContinueTest
Possible Coordinates (*)
| 2  4 | 3  5 | 4  2 | 5  3 |

[       OK ] OthelloTest.IsGameContinueTest (0 ms)
[ RUN      ] OthelloTest.HasOnlyOneTypeTest
[       OK ] OthelloTest.HasOnlyOneTypeTest (0 ms)
[ RUN      ] OthelloTest.EndingTest
[       OK ] OthelloTest.EndingTest (0 ms)
[-----] 12 tests from OthelloTest (0 ms total)

[-----] Global test environment tear-down
[=====] 12 tests from 1 test suite ran. (0 ms total)
[ PASSED  ] 12 tests.
wanpile@MacBookPro Othello %
```

보드 크기 설정

Judge::CreateBoard()
함수를 통해
보드 크기 입력을 받고,
불가능한 입력은
제한하였음

```
/* CreateBoard : board의 크기를 입력받는 함수 */  
void Judge::CreateBoard() {  
    std::string input = "";  
    int size = 0;  
  
    std::cout << "Enter the size of board : ";  
    std::cin >> input;  
    size = atoi(input.c_str());  
    while ((size % 2 == 1) || (size < 4)) {  
        std::cout << "Enter an even number of 6 or more : ";  
        std::cin >> input;  
        size = atoi(input.c_str());  
    }  
    (this->board_)->SetSize(size);  
}
```

Makefile, Lint 진행

```
Othello > M makefile
29      $(AR) $(ARFLAGS) $@ $^
30
31 Board.o : Board.h Board.cpp
32     g++ --std=c++17 -c Board.cpp -o $@
33
34 Player.o : Player.h Player.cpp
35     g++ --std=c++17 -c Player.cpp -o $@
36
37 Judge.o : Judge.h Judge.cpp Board.o Player.o
38     g++ --std=c++17 -c Judge.cpp -o $@
39
40 main : main.cpp Board.o Player.o Judge.o $(GTEST_HEADERS)
41     g++ --std=c++17 main.cpp Board.o Player.o Judge.o -o $@
42
43 othello_test.o : othello_test.cpp $(GTEST_HEADERS)
44     g++ $(CPPFLAGS) $(CXXFLAGS) --std=c++17 -c othello_test.cpp -o $@
45
46 test : othello_test.o Judge.o Board.o Player.o gtest_main.a
47     g++ $(CPPFLAGS) $(CXXFLAGS) --std=c++17 $^ -o $@
48
49 .PHONY: all clean
50
```

```
wanpile@MacBookPro Othello % make
g++ --std=c++17 -c Board.cpp -o Board.o
g++ --std=c++17 -c Player.cpp -o Player.o
g++ --std=c++17 -c Judge.cpp -o Judge.o
g++ --std=c++17 main.cpp Board.o Player.o Judge.o -o main
```

```
wanpile@MacBookPro Othello % cpplint --filter=-build/include_subdir,-legal/copyr
ight *.h *cpp
Done processing Board.cpp
Done processing Board.h
Done processing Judge.cpp
Done processing Judge.h
Done processing Player.cpp
Done processing Player.h
Done processing main.cpp
Done processing othello_test.cpp
wanpile@MacBookPro Othello %
```

기능성

- 구현한 기능
 - 현재 점수 확인
 - 가능한 좌표 표시
- 미 구현 및 추가 했으면 좋았을 기능
 - GUI 기능
 - 001로 입력시 1로 입력되는 버그 해결

안전성

- 입력에 제한을 두는 함수 설정
 - Judge::CreateBoard() – 보드 사이즈 입력 제한
 - Judge::SetPlayers() – 돌 색 제한
- 예외 상황 처리를 위한 함수
 - Judge::IsGameContinue() – 게임 진행 가능 여부 표시
- 잘못 두는 상황 방지를 위한 가이드 좌표 표시

Judge::SetPlayers()

```
Judge.cpp > SetPlayers()
~/Documents/GitHub/Othello/Judge.cpp
184 std::cin >> tmp;
185 if ((tmp.front() == 'B') || (tmp.front() == 'W')) {
186     break;
187 } else {
188     std::cout << "You have to choose either B or W." << std::endl;
189 }
190 }
191 p1_color = tmp.front();
192 p1->SetColor(p1_color);
193
194 std::cout << "Enter Player2's name : ";
195 std::cin >> tmp;
196 p2->SetName(tmp);
197 if (p1_color == 'B') {
198     p2->SetColor('W');
199     std::cout << "Player2's color is W." << std::endl;
200 } else {
201     p2->SetColor('B');
202     std::cout << "Player2's color is B." << std::endl;
203 }
204 }
```

Judge::IsGameContinue()

Board::IsFullBoard()

Board::HasOnlyOneType()

위 함수들로 게임이 계속 진행될 수 있는지 확인

Judge::CheckValid(int row, int col, char color)

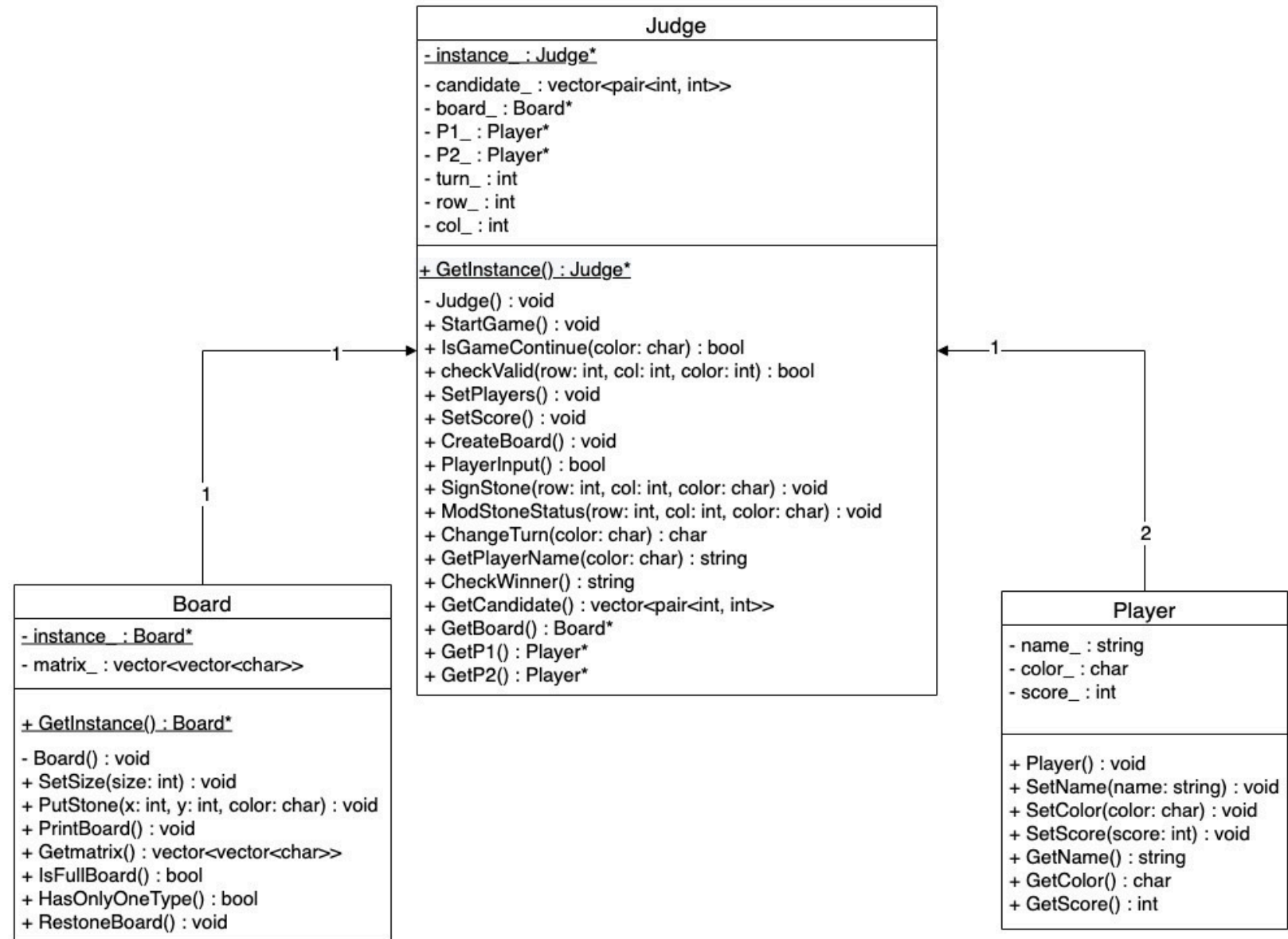
위 함수로 가이드 좌표 제작

```
Judge.cpp > IsGameContinue(char)
85  /* IsGameContinue : 게임을 진행할 수 있는지 여부를 확인하는 함수 */
86  bool Judge::IsGameContinue(char color) {
87      // 한 차례에 양쪽 모두 서로 차례를 넘겨야 하는 경우
88      std::vector<std::vector<char>> matrix = board_>Getmatrix();
89      int size = matrix.size();
90
91      if (board_>IsFullBoard()) { return false; }
92      // 둘을 놓을 수 있는 공간이 없을 경우
93      if (board_>HasOnlyOneType()) { return false; }
94      // 한가지 색의 돌만 존재하는 경우
95
96      (this->candidate_).clear();
97      std::vector<std::pair<int, int>> ().swap(this->candidate_);
98
99      for (int i = 0; i < size; i++) { /* 후보군들을 알아내기 위한 부분 */
100          for (int j = 0; j < size; j++) {
101              if (CheckValid(i, j, color)) {
102                  (this->candidate_).push_back(std::make_pair(i, j));
103              }
104          }
105      }
106  }
```

디자인 패턴

- 싱글톤 패턴 사용

UML



실행 시나리오 진행

```
wanpile@MacBookPro Othello % ./main  
Welcome to Othello  
  
Enter the size of board : █
```

시작정보 입력

```
Welcome to Othello
```

```
Enter the size of board : 11
```

```
Enter an even number of 6 or more : 10
```

```
Enter Player1's name : player11
```

```
Player1 can choose color (B or W) : E
```

```
You have to choose either B or W.
```

```
Player1 can choose color (B or W) : W
```

```
Enter Player2's name : player22
```

```
Player2's color is B.
```

```
Possible Coordinates (*)
```

```
| 3 5 | 4 6 | 5 3 | 6 4 |
```

\	0	1	2	3	4	5	6	7	8	9
0
1
2
3	*
4	B	W	*	.	.	.
5	.	.	.	*	W	B
6	*
7
8
9

```
player11(W) 2 : 2 player22(B)
```

```
Player player22's Turn
```

```
Enter row and column ex) 2 4 : █
```

게임 진행

```
Possible Coordinates (*)
| 2 3 | 3 3 | 4 3 | 5 3 | 6 3 |
\ 0 1 2 3 4 5 6 7 8 9
0 . . . . . . . . .
1 . . . . . . . . .
2 . . . * . . . . .
3 . . . * W B . . .
4 . . . * W B . . .
5 . . . * W B . . .
6 . . . * . . . . .
7 . . . . . . . . .
8 . . . . . . . . .
9 . . . . . . . . .

player11(W) 3 : 3 player22(B)
Player player22's Turn
Enter row and column ex) 2 4 : 3 3

Possible Coordinates (*)
| 2 4 | 3 6 | 5 6 |
\ 0 1 2 3 4 5 6 7 8 9
0 . . . . . . . . .
1 . . . . . . . . .
2 . . . * . . . . .
3 . . . B B B * . .
4 . . . . B B . . .
5 . . . . W B * . .
6 . . . . . . . . .
7 . . . . . . . . .
8 . . . . . . . . .
9 . . . . . . . . .

player11(W) 1 : 6 player22(B)
Player player11's Turn
Enter row and column ex) 2 4 : 1 1
It's not a right choice.
Enter row and column ex) 2 4 : █
```

```
Possible Coordinates (*)
| 1 3 | 1 5 | 2 3 | 2 5 | 5 1 | 6 1 | 6 3 | 6 5 | 7 3 | 7 5 |
\ 0 1 2 3 4 5 6 7 8 9
0 . . . . . . . . .
1 . . . * . * . . .
2 . . . * W * . . .
3 . . . B W B . . .
4 . . . B W B . . .
5 . * W B W B . . .
6 . * . * W * . . .
7 . . . * . * . . .
8 . . . . . . . . .
9 . . . . . . . . .

player11(W) 6 : 6 player22(B)
Player player22's Turn
Enter row and column ex) 2 4 : -1 1
It's not a right choice.
Enter row and column ex) 2 4 : █
```


게임 종료

```
Possible Coordinates (*)
| 2 3 | 3 2 |

\ 0 1 2 3
0 W W W W
1 W W W B
2 W W B *
3 W B * W

p1(B) 3 : 11 p2(W)
Player p2's Turn
Enter row and column ex) 2 4 : 3 2

Possible Coordinates (*)
|
\ 0 1 2 3
0 W W W W
1 W W W B
2 W W W .
3 W W W W

Oops! Player p1 has no space to put stone.
It's Player p2's turn again.

Possible Coordinates (*)
| 2 3 |
\ 0 1 2 3
0 W W W W
1 W W W B
2 W W W *
3 W W W W

p1(B) 1 : 14 p2(W)
Player p2's Turn
Enter row and column ex) 2 4 : 2 3

p1(B) 1 : 14 p2(W)
Congratulations! Player p2 wins!
```