

CSED 232 Object-Oriented Programing (Spring 2021)

Programming Assignment # 3

- Inheritance & Polymorphism -

Due date : 4월 30일

담당 조교 : 김성태 (seongtae0205@postech.ac.kr)

주의사항

- string을 제외한 추가적인 STL의 사용은 허용하지 않습니다.
- 각 클래스마다 header file과 cpp file을 나누어 구현하여야 합니다. 추가적으로 필요한 멤버 함수(method)나 멤버 변수(member variable)가 있을 경우, 추가해도 무방합니다.
- 문제에 명시되어 있지 않더라도 각 클래스마다 생성자(Constructor), 소멸자(Destructor)는 필수입니다. 기본 생성자의 경우, 별다른 언급이 없다면 멤버 변수에 int, double은 0, string은 "", pointer는 NULL로 초기화해 주십시오.
- 클래스마다 클래스명과 파일명을 통일하여 주십시오.
- 모든 클래스의 멤버 변수는 private으로 선언하여야 합니다.

감점

- 제출 기한이 지나면 얻은 총점의 20% 감점
- 추가로 하루(24시간) 늦을 때마다 20%씩 감점
1일 이내 지연: 20% 감점, 2일 이내 지연: 40% 감점, 5일 이상 지연: 0점
- 컴파일이 정상적으로 이루어지지 않을 경우 0점

제출방식

채점은 Windows Visual Studio 2019 환경에서 이루어집니다. 작업하신 환경이 있는 프로젝트 폴더에 Report를 포함하여 zip파일로 압축 후 제출해 주시기 바랍니다. 폴더명은 '#학번'으로 만들어 주시고, Report는 pdf 형식으로 제출해주세요. 반드시 PLMS를 통해 제출해주시기 바랍니다. 이메일 제출은 인정되지 않습니다. 5일이 지날 경우 0점이므로 과제 제출 마감일로부터 5일후인 5월

5월 23시 59분 59초 이후는 PLMS를 통해 제출하실 수 없습니다.

예) 20209999.zip

채점 기준

1. 프로그램 기능

- 프로그램이 요구 사항을 모두 만족하면서 올바르게 실행되는가?

2. 프로그램 설계 및 구현

- 요구 사항을 만족하기 위한 변수 및 알고리즘 설계가 잘 되었는가?
- 설계된 내용이 요구된 언어를 이용하여 적절히 구현되었는가?

3. 프로그램 가독성

- 프로그램이 읽기 쉽고 이해하기 쉽게 작성되었는가?
- 변수 명이 무엇을 의미하는지 이해하기 쉬운가?
- 프로그램의 소스 코드를 이해하기 쉽도록 주석을 잘 붙였는가?

4. 보고서 구성 및 내용, 양식

- 보고서는 적절한 내용으로 이해하기 쉽고 보기 좋게 잘 작성되었는가?
- 보고서의 양식을 잘 따랐는가?

다른 사람의 프로그램이나 인터넷에 있는 프로그램을 복사(copy)하거나 간단히 수정해서 제출하면 학점은 무조건 'F'가 됩니다. 이러한 부정행위가 발견되면 학과에서 정한 기준에 따라 추가의 불이익이 있을 수 있습니다.

신뢰의 진화

본 과제에서는 '신뢰의 진화' 게임을 직접 구현해보면서 Inheritance와 Polymorphism 개념을 익혀보도록 한다.

1. 게임설명

'신뢰의 진화' 게임은 경제학 개념 중 하나인 게임이론을 누구나 쉽게 이해할 수 있도록 만들어진 인터넷 오픈소스 게임이다. 아래 링크를 들어가면 누구나 무료로 플레이 할 수 있으며, 과제를 진행하기 전에 직접 플레이 해보는 것을 권장한다.

<https://osori.github.io/trust-ko/>

1.1 신뢰 게임(Round)

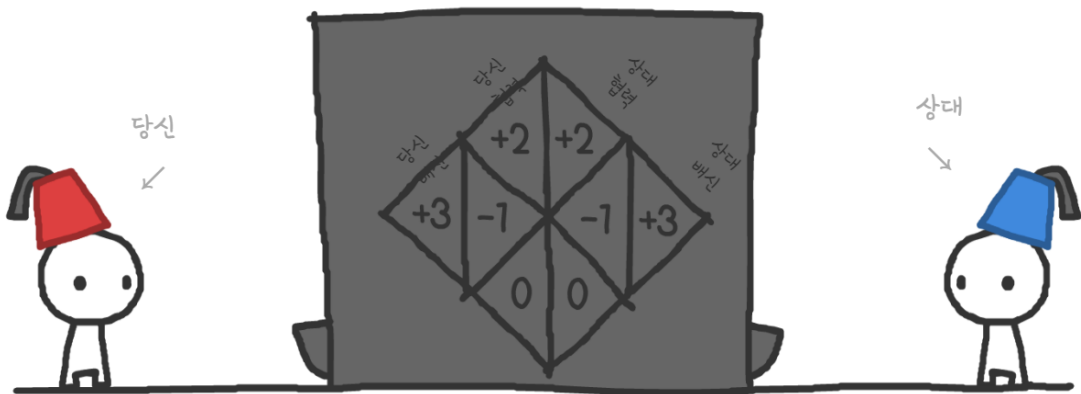


Figure 1. 신뢰 게임

Figure 1과 같은 상황이 있다고 해보자. 두 사람은 각자 협력 혹은 배신을 선택할 수 있다. 이는 동시에 이루어진다. 만약 두 사람 모두 협력하기로 하면, 두 사람에게 +2의 reward가 주어진다. 하지만 둘 중 한 사람만 협력하고 다른 한 사람은 배신을 하게 된다면, 협력을 선택한 사람은 -1, 배신을 한 사람은 +3의 reward를 얻는다. 둘 다 배신을 선택하게 된다면 모두 +0의 reward를 얻게 된다.

1.2 대결 (Match)

대결(Match)은 상대방과 여러 번의 Round를 진행하는 것을 의미한다. Figure 2는 총 5 Round로 구성된 Match의 진행 과정을 보여준다. 플레이어는 '배신-협력-협력-배신-배신' 전략을 취하였고, 상대방은 '협력-배신-협력-협력-배신' 전략을 취하였다. 5 Round를 진행한 후 최종 reward는 각자 7, 3이 되었다.

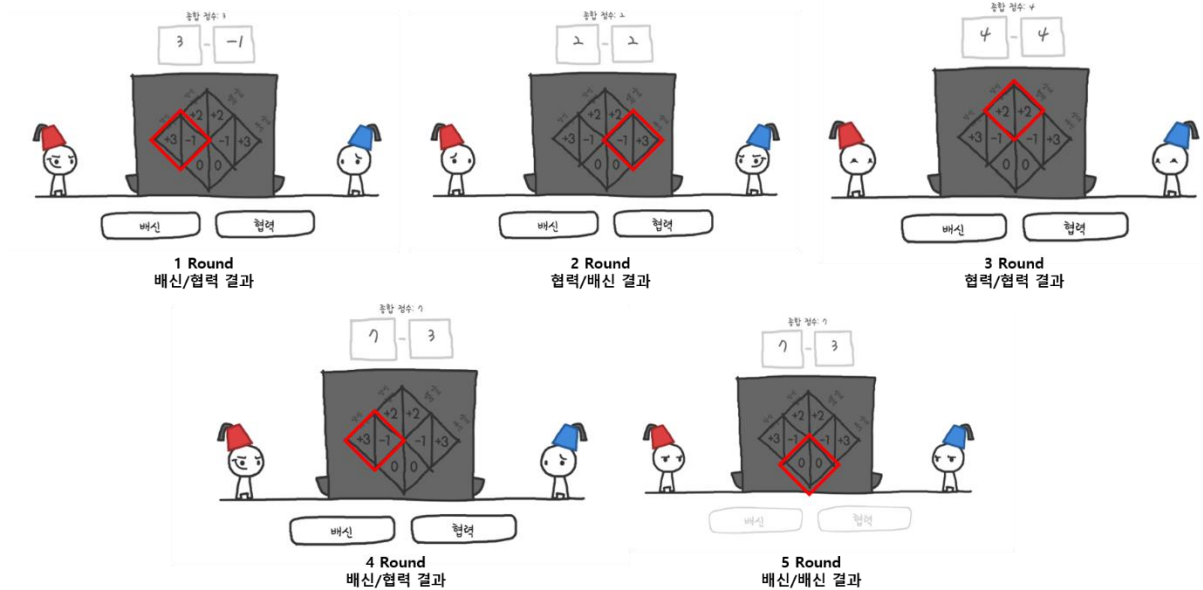


Figure 2. 5 Round로 구성된 Match 예시

1.3 직업

본 게임에는 총 6개의 직업이 있다. 각 직업 별로 전략이 상이하다.



플레이어(Player): 사용자의 입력을 받아 행동한다.

따라쟁이(Copycat): 협력으로 시작하며, 상대방의 이전 행동을 따라한다

항상배신자(Cheater): 항상 배신만 한다.

항상협력자(Cooperator): 항상 협력만 한다

원한을 가진 자(Grudger): 처음부터 계속 협력하다가, 상대방이 배신하는 순간 항상배신자처럼 행동한다.

탐정(Detective): 협력-배신-협력-협력으로 시작. 앞에 언급된 초반 4 Round 동안, 상대방이 한번이라도 배신을 했다면 이후에는 따라쟁이처럼 행동한다. 반면 초반 4 Round 동안 상대방이 항상 협력만 했다면 항상배신자처럼 행동한다.

1.4 토너먼트(Tournament)

Player를 제외한 직업들로 구성된 사람들 사이에 모든 Match를 진행시킨다. 각 Match는 총 10 Round로 구성되어 있다. Figure 3는 서로 다른 직업을 가진 5명의 사람들 사이에서 진행된 Tournament 결과를 보여준다. 5명으로 Tournament를 진행하게 된다면 총 ${}_5C_2$ 인 10번의 Match가 진행된다.

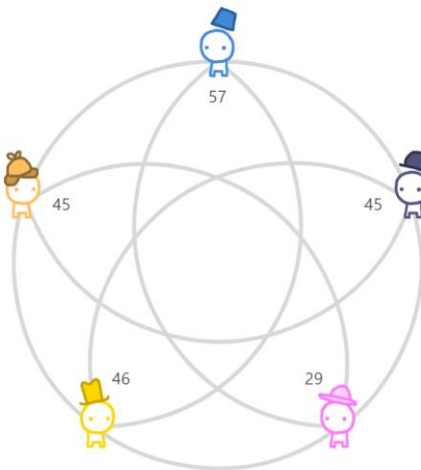


Figure 3. Tournament 결과

1.5 반복 토너먼트(Repeated Tournament)

반복 토너먼트는 Tournament를 여러 번 진행하는 것을 의미한다. 이 때의 Tournament는 두 개의 추가적인 과정이 추가된다. 모든 Match를 진행하였으면 reward 하위 5명을 탈락시킨다. 하위 5명을 탈락시켰으면, reward 1등 직업을 5명 재생산한다. 한번의 Tournament 과정(경기진행->패자탈락(Eliminate)->승자재생산(Reproduce))이 끝나면 모든 사람들의 reward는 다시 0으로 초기화 된다. Figure 4는 반복 토너먼트의 예시를 보여준다.



Figure 4. Repeated Tournament에서의 Tournament 과정

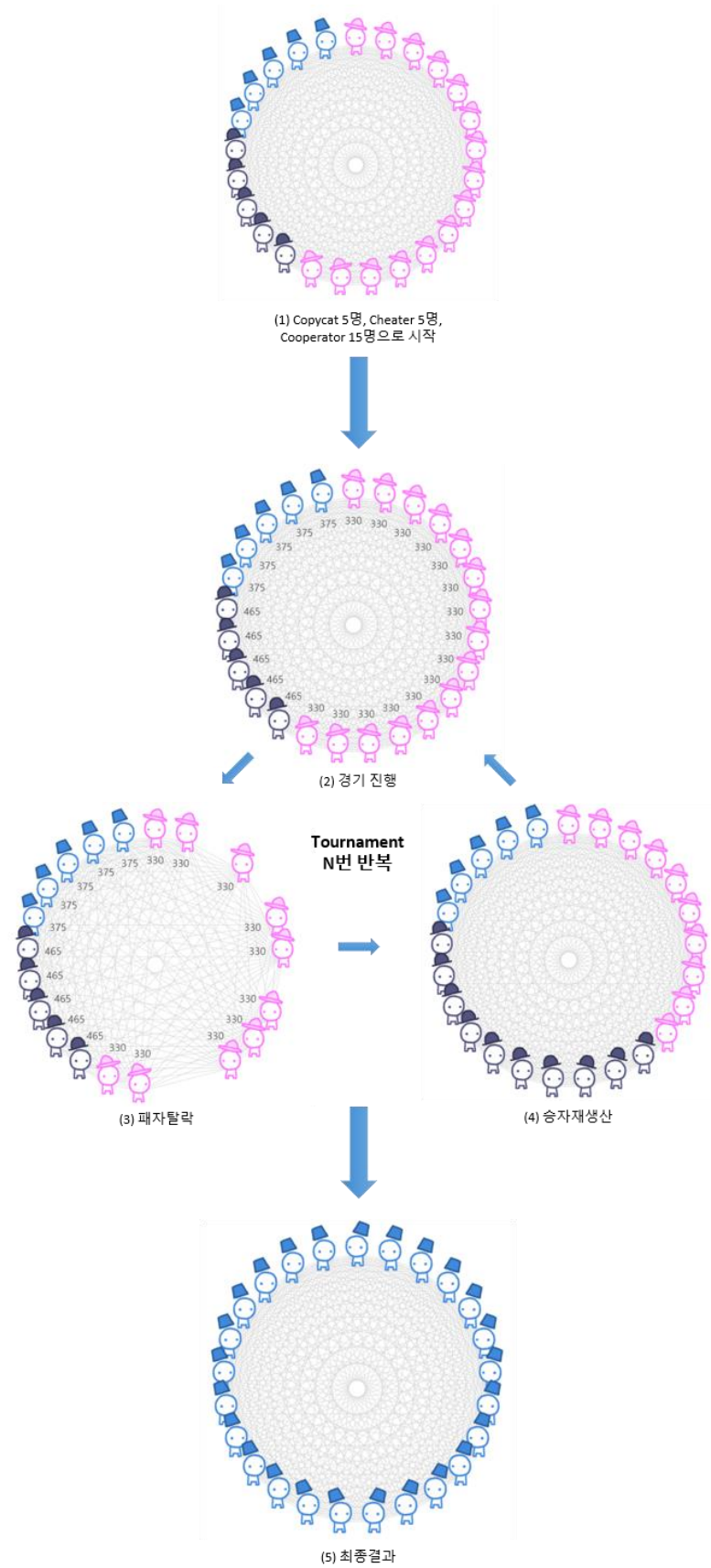


Figure 5. Copycat 5명, Cheater 5명, Cooperator 15명으로 시작한 반복 토너먼트 예시

2. 게임구현

2.1 시작 메뉴

사용자는 시작 메뉴로부터 어떤 게임을 할 지 선택할 수 있으며 프로그램을 종료할 수 있다. 사용자가 선택할 수 있는 메뉴의 종류는 총 4가지이다.

① Single Match

- Player가 직접 참여하여 여러 Round를 진행하는 게임. Total Round Number를 입력 받아 진행할 Round 개수를 설정한다. 사용자는 상대방의 직업을 선택할 수 있다. 사용자가 직접 Player의 행동을 제어하고 매 Round 결과를 확인하며 게임을 진행한다.

② Single Tournament

- 사용자가 원하는 인원과 직업 구성으로 단일 Tournament를 진행한다. 이 때 Player는 제외된다. 각 Match는 10 Round로 고정한다. 사용자는 매 Match 결과를 확인하며 게임을 진행한다. 사용자는 중간에 게임을 종료하고 시작 메뉴로 돌아갈 수 있다.

③ Repeated Tournament

- 사용자가 원하는 인원과 직업 구성으로 Repeated Tournament를 진행한다. 이 때 Player는 제외된다. 이 경우에도 각 Match는 10 Round로 고정한다. Total Tournament Number를 입력 받아 진행할 Tournament 개수를 설정한다. 매 Tournament에는 패자 탈락(Eliminate), 승자재생산(Reproduce) 과정이 추가적으로 이루어진다. 사용자는 매 Tournament 결과를 확인하며 게임을 진행한다. 사용자는 중간에 게임을 종료하고 시작 메뉴로 돌아갈 수 있다.

④ Exit

- 프로그램을 종료한다.

```
=====
Select Menu
=====
1. Single Match
2. Single Tournament
3. Repeated Tournament
4. Exit
=====
Command:
```

Figure 6. 시작 메뉴

2.2 Single Match

시작 메뉴에서 1을 입력 받으면 실행한다. Total Round Number를 입력 받아 총 진행할 Round 수를 정한다 (Figure 7).

```
=====
Select Menu
=====
1. Single Match
2. Single Tournament
3. Repeated Tournament
4. Exit
=====
Command: 1
Total Round Number: 5
Select an Opponent! Copycat(1), Cheater(2), Cooperator(3), Grudger(4), Detective(5)
```

Figure 7. Total Round Number 입력 후

진행할 Round 입력을 받았으면, 상대방을 선택한다 (Figure 8).

```
=====
Select Menu
=====
1. Single Match
2. Single Tournament
3. Repeated Tournament
4. Exit
=====
Command: 1
Total Round Number: 5
Select an Opponent! Copycat(1), Cheater(2), Cooperator(3), Grudger(4), Detective(5)
1
=====Round 1=====
Cooperating(1) or Cheating (0)?
_
```

Figure 8. 상대방 선택 후

Round를 진행하면서 Player의 행동 Cooperating(1) 또는 Cheating(0)을 입력 받는다. Player의 행동을 입력 받으면 Player와 상대방의 행동을 출력하고, 현재 Total Reward를 출력한다. 행동 입력은 숫자로 받는다. 그 후 다음 Round를 진행한다 (Figure 9).


```

=====
Select Menu
=====
1. Single Match
2. Single Tournament
3. Repeated Tournament
4. Exit
=====
Command:1
Total Round Number: 5
Select an Opponent! Copycat(1), Cheater(2), Cooperator(3), Grudger(4), Detective(5)
1
=====Round 1=====
Cooperating(1) or Cheating (0)?
0
Player: Cheating!
Copycat: Cooperating!
-----
Total Reward
1. Player: 3
2. Copycat: -1
=====Round 2=====
Cooperating(1) or Cheating (0)?

```

Figure 9. Player 행동 입력 후

모든 Round를 진행하였으면 최종 승자를 출력하고 시작 메뉴로 돌아간다(Figure 10).

```

2. Copycat: 4
=====Round 4=====
Cooperating(1) or Cheating (0)?
0
Player: Cheating!
Copycat: Cooperating!
-----
Total Reward
1. Player: 7
2. Copycat: 3
=====Round 5=====
Cooperating(1) or Cheating (0)?
0
Player: Cheating!
Copycat: Cheating!
-----
Total Reward
1. Player: 7
2. Copycat: 3
=====
The winner is Player
=====
Select Menu
=====
1. Single Match
2. Single Tournament
3. Repeated Tournament
4. Exit
=====
Command:

```

Figure 10. 최종 Round 진행 후, 승자 출력 및 시작 메뉴 복귀

2.3 Single Tournament

시작 메뉴에서 2를 입력 받으면 실행한다. Single Tournament가 시작되면 "Decide Population"을 출력하고 각 직업별 인원을 차례대로 입력 받는다(Figure 11).

```
=====
Select Menu
=====
1. Single Match
2. Single Tournament
3. Repeated Tournament
4. Exit
=====
Command:2
Decide Population
Copycat:
```

Figure 11. Single Tournament 시작

직업별 인원을 입력 받았으면, 현재 Total Reward를 출력하고 Continue(1)과 Stop(0) 중 하나를 입력으로 받는다(Figure 12). 입력은 숫자로 받는다.

```
=====
Select Menu
=====
1. Single Match
2. Single Tournament
3. Repeated Tournament
4. Exit
=====
Command:2
Decide Population
Copycat: 1
Cheater: 1
Cooperator: 1
Grudger: 1
Detective: 1
-----
1. Copycat: 0
2. Cheater: 0
3. Cooperator: 0
4. Grudger: 0
5. Detective: 0
-----
Continue(1) or Stop(0)
```

Figure 12. Population 입력 받은 후

Continue(1)를 입력 받으면 다음 Match를 진행한다. Match를 진행한 사람의 index를 "1 vs 2"의 바 형식에 맞추어 출력한다. 그리고 현재 Total Reward를 출력한다(Figure 13).

```
1. Single Match
2. Single Tournament
3. Repeated Tournament
4. Exit
=====
Command:2
Decide Population
Copycat: 1
Cheater: 1
Cooperator: 1
Grudger: 1
Detective: 1
-----
1. Copycat: 0
2. Cheater: 0
3. Cooperator: 0
4. Grudger: 0
5. Detective: 0
-----
Continue(1) or Stop(0)
1
=====1 vs 2=====
1. Copycat: -1
2. Cheater: 3
3. Cooperator: 0
4. Grudger: 0
5. Detective: 0
-----
Continue(1) or Stop(0)
```

Figure 13. Match 진행

모든 Match가 끝나면 최종 Total Reward를 **내림차순 정렬**하여 보여주고 시작 메뉴로 돌아간다 (Figure 14).

```

3. Cooperator: 29
4. Grudger: 39
5. Detective: 42
=====
Continue(1) or Stop(0)
1
=====4 vs 5=====
1. Copycat: 57
2. Cheater: 45
3. Cooperator: 29
4. Grudger: 46
5. Detective: 45
=====
1. Copycat: 57
2. Grudger: 46
3. Cheater: 45
4. Detective: 45
5. Cooperator: 29
=====
Select Menu
=====
1. Single Match
2. Single Tournament
3. Repeated Tournament
4. Exit
=====
Command:

```

Figure 14. 최종 Match 진행 후, 최종 Total Reward 정렬 출력 및 시작 메뉴 복귀

Tournament 진행 도중 Stop(0)을 입력 받으면 시작 메뉴로 돌아간다.

```

=====1 vs 3=====
1. Copycat: 19
2. Cheater: 3
3. Cooperator: 20
4. Grudger: 0
5. Detective: 0
=====
Continue(1) or Stop(0)
0
=====
Select Menu
=====
1. Single Match
2. Single Tournament
3. Repeated Tournament
4. Exit
=====
Command:

```

Figure 15. Single Tournament 중단

2.4 Repeated Tournament

시작 메뉴에서 3을 입력 받으면 실행한다. Repeated Tournament가 시작되면 Total Tournament Number를 입력 받는다. 그 후 "Decide Population"을 출력하고 각 직업별 인원을 차례대로 입력 받는다. 입력을 다 받았으면 Single Tournament와 동일하게 현재 Total Reward를 한 번 출력해준다(Figure 16).

```
=====
Select Menu
=====
1. Single Match
2. Single Tournament
3. Repeated Tournament
4. Exit
=====
Command:3
Total Tournament Number: 10
Decide Population
Copycat: 5
Cheater: 5
Cooperator: 15
Grudger: 0
Detective: 0
-----
1. Copycat: 0
2. Copycat: 0
3. Copycat: 0
4. Copycat: 0
5. Copycat: 0
6. Cheater: 0
7. Cheater: 0
8. Cheater: 0
9. Cheater: 0
10. Cheater: 0
11. Cooperator: 0
12. Cooperator: 0
13. Cooperator: 0
14. Cooperator: 0
15. Cooperator: 0
16. Cooperator: 0
17. Cooperator: 0
18. Cooperator: 0
19. Cooperator: 0
20. Cooperator: 0
21. Cooperator: 0
22. Cooperator: 0
23. Cooperator: 0
24. Cooperator: 0
25. Cooperator: 0
-----
Continue(1) or Stop(0)
_
```

Figure 16. Repeated Tournament에서 Total Tournament Number, Population 입력 후

Continue(1)을 입력 받으면 Tournament를 진행하고 Total Reward를 **내림차순 정렬**하여 보여준다(Figure 17).

```

23. Cooperator: 0
24. Cooperator: 0
25. Cooperator: 0
-----
Continue(1) or Stop(0)
1
=====Tournament 1=====
1. Cheater: 465
2. Cheater: 465
3. Cheater: 465
4. Cheater: 465
5. Cheater: 465
6. Copycat: 375
7. Copycat: 375
8. Copycat: 375
9. Copycat: 375
10. Copycat: 375
11. Cooperator: 330
12. Cooperator: 330
13. Cooperator: 330
14. Cooperator: 330
15. Cooperator: 330
16. Cooperator: 330
17. Cooperator: 330
18. Cooperator: 330
19. Cooperator: 330
20. Cooperator: 330
21. Cooperator: 330
22. Cooperator: 330
23. Cooperator: 330
24. Cooperator: 330
25. Cooperator: 330
-----
Eliminate(1) or Stop(0)

```

Figure 17. Tournament 진행 결과

Repeated Tournament에서는 Tournament 진행 후 패자탈락(Eliminate) 과정이 이어진다. 이 역시 숫자 1을 입력 받으면 진행하고 0을 입력 받으면 시작 메뉴로 돌아간다. Eliminate를 진행하였을 때는 별도의 출력이 존재하지 않는다. 이후 승자재생산(Reproduce)과정 또한 동일한 과정으로 입력을 받는다. Reproduce를 진행하면 모든 reward를 0으로 초기화하여 현재 Total Reward를 출력한다(Figure 19). Eliminate와 Reproduce 과정은 Reward를 기준으로 내림차순 정렬 후 Figure18처럼 구현하여야 한다.

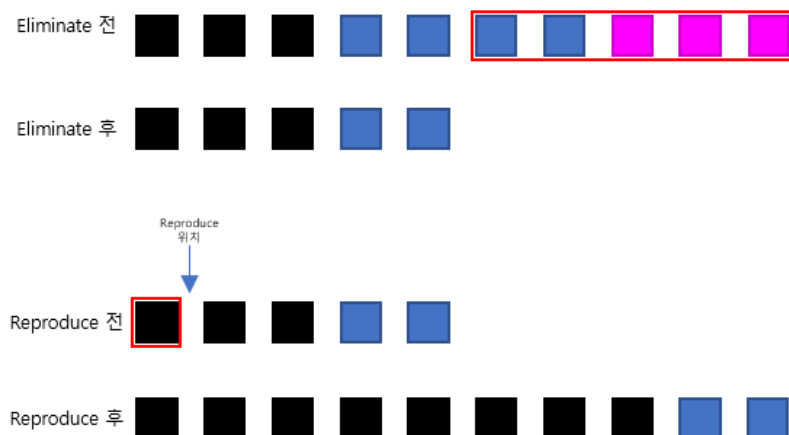


Figure 18. Eliminate, Reproduce 예시

```

19. Cooperator: 330
20. Cooperator: 330
21. Cooperator: 330
22. Cooperator: 330
23. Cooperator: 330
24. Cooperator: 330
25. Cooperator: 330
-----
Eliminate(1) or Stop(0)
1
-----
Reproduce(1) or Stop(0)
1
-----
1. Cheater: 0
2. Cheater: 0
3. Cheater: 0
4. Cheater: 0
5. Cheater: 0
6. Cheater: 0
7. Cheater: 0
8. Cheater: 0
9. Cheater: 0
10. Cheater: 0
11. Copycat: 0
12. Copycat: 0
13. Copycat: 0
14. Copycat: 0
15. Copycat: 0
16. Cooperator: 0
17. Cooperator: 0
18. Cooperator: 0
19. Cooperator: 0
20. Cooperator: 0
21. Cooperator: 0
22. Cooperator: 0
23. Cooperator: 0
24. Cooperator: 0
25. Cooperator: 0
-----
Continue(1) or Stop(0)
-

```

Figure 19. Eliminate와 Reproduce과정 후

Figure 20은 Repeated Tournament의 중간 결과를 보여준다. 본 예제에서 7번의 Tournament를 진행하면 결국 Copycat만 살아 남는 것을 확인할 수 있다. Figure 21은 최종 결과를 보여준다. 모든 Tournament가 끝나면 시작 메뉴로 돌아간다.

```

Continue(1) or Stop(0)
1
=====Tournament 7=====
1. Copycat: 375
2. Copycat: 375
3. Copycat: 375
4. Copycat: 375
5. Copycat: 375
6. Copycat: 375
7. Copycat: 375
8. Copycat: 375
9. Copycat: 375
10. Copycat: 375
11. Copycat: 375
12. Copycat: 375
13. Copycat: 375
14. Copycat: 375
15. Copycat: 375
16. Copycat: 375
17. Copycat: 375
18. Copycat: 375
19. Copycat: 375
20. Copycat: 375
21. Cheater: 60
22. Cheater: 60
23. Cheater: 60
24. Cheater: 60
25. Cheater: 60
-----
Eliminate(1) or Stop(0)
1
-----
Reproduce(1) or Stop(0)
1
-----
1. Copycat: 0
2. Copycat: 0
3. Copycat: 0
4. Copycat: 0
5. Copycat: 0
6. Copycat: 0
7. Copycat: 0
8. Copycat: 0
9. Copycat: 0
10. Copycat: 0
11. Copycat: 0
12. Copycat: 0
13. Copycat: 0
14. Copycat: 0
15. Copycat: 0
16. Copycat: 0
17. Copycat: 0
18. Copycat: 0
19. Copycat: 0
20. Copycat: 0
21. Copycat: 0
22. Copycat: 0
23. Copycat: 0
24. Copycat: 0
25. Copycat: 0
-----
Continue(1) or Stop(0)
1
=====Tournament 8=====
1. Copycat: 480
2. Copycat: 480
3. Copycat: 480
4. Copycat: 480
5. Copycat: 480

```

Figure 20. Tournament 7 결과


```
Continue(1) or Stop(0)
1
=====Tournament 10=====
1. Copycat: 480
2. Copycat: 480
3. Copycat: 480
4. Copycat: 480
5. Copycat: 480
6. Copycat: 480
7. Copycat: 480
8. Copycat: 480
9. Copycat: 480
10. Copycat: 480
11. Copycat: 480
12. Copycat: 480
13. Copycat: 480
14. Copycat: 480
15. Copycat: 480
16. Copycat: 480
17. Copycat: 480
18. Copycat: 480
19. Copycat: 480
20. Copycat: 480
21. Copycat: 480
22. Copycat: 480
23. Copycat: 480
24. Copycat: 480
25. Copycat: 480
-----
Eliminate(1) or Stop(0)
1
-----
Reproduce(1) or Stop(0)
1
-----
1. Copycat: 0
2. Copycat: 0
3. Copycat: 0
4. Copycat: 0
5. Copycat: 0
6. Copycat: 0
7. Copycat: 0
8. Copycat: 0
9. Copycat: 0
10. Copycat: 0
11. Copycat: 0
12. Copycat: 0
13. Copycat: 0
14. Copycat: 0
15. Copycat: 0
16. Copycat: 0
17. Copycat: 0
18. Copycat: 0
19. Copycat: 0
20. Copycat: 0
21. Copycat: 0
22. Copycat: 0
23. Copycat: 0
24. Copycat: 0
25. Copycat: 0
=====
Select Menu
=====
1. Single Match
2. Single Tournament
3. Repeated Tournament
4. Exit
=====
Command:
```

Figure 21. 최종 Tournament 진행 후 시작 메뉴 복귀

2.4 예외처리

* 다음과 같은 입력을 받았을 때에는 "Invalid Input!\n"을 출력하고 입력을 다시 받는다. (Figure 22)

시작 메뉴

- 1, 2, 3, 4 외 다른 입력 받았을 때

Single Match

- Total Round Number에서 1 이상, 50 이하 외 다른 입력을 받았을 때
- 유효한 Opponent번호(1, 2, 3, 4, 5) 외 다른 입력을 받았을 때
- 유효한 행동 번호(1, 0) 외 다른 입력을 입력 받았을 때

Single Tournament

- 각 직업별 인원 입력에서 0 이상, 50 이하 외 다른 입력을 받았을 때
- 유효한 명령 (1, 0) 외 다른 입력을 입력 받았을 때

Repeated Tournament

- Total Tournament Number에서 1 이상, 50 이하 외 다른 입력을 받았을 때
- 각 직업별 인원 입력에서 0 이상, 50 이하 외 다른 입력을 받았을 때
- 유효한 명령 (1, 0) 외 다른 입력을 입력 받았을 때

* 다음과 같은 경우에는 "Invalid Input!\n"을 출력하고 Population 입력을 처음부터 다시 받는다. (Figure 23) ("Decide Population" 출력 다시 하지 않음)

Single Tournament

- 전체 인원이 2 미만 또는 50 초과 일 경우

Repeated Tournament

- 전체 인원이 6 미만 또는 50 초과 일 경우

```

=====
Select Menu
=====
1. Single Match
2. Single Tournament
3. Repeated Tournament
4. Exit
=====
Command:1
Total Round Number: 200
Invalid Input!
10
Select an Opponent! Copycat(1), Cheater(2), Cooperator(3), Grudger(4), Detective(5)
_

```

Figure 22. 예외처리 예시 1

```

=====
Select Menu
=====
1. Single Match
2. Single Tournament
3. Repeated Tournament
4. Exit
=====
Command:2
Decide Population
Copycat: 0
Cheater: 0
Cooperator: 0
Grudger: 0
Detective: 0
Invalid Input!
Copycat: 2
Cheater: 1
Cooperator: 5
Grudger: 0
Detective: 0
-----
1. Copycat: 0
2. Copycat: 0
3. Cheater: 0
4. Cooperator: 0
5. Cooperator: 0
6. Cooperator: 0
7. Cooperator: 0
8. Cooperator: 0

```

Figure 23. 예외처리 예시 2

2.5 Class

Agent: 각 직업 Class의 Parent Class로 Abstract Class로 정의하여 사용한다. act(...) 함수를 Pure virtual 함수로 정의한다.

Player, Copycat, Cheater, Cooperator, Grudger, Detective: Agent를 Base Class로 하는 Derived Class. act(...) 함수는 상대방의 history를 인자로 받아 행동을 반환한다.

Population: 구현자가 정의한 자료구조 Class를 상속받아 Agent들을 관리하는 Class. Eliminate, Reproduce, sort 등을 담당한다. 또한 Total Reward 출력 등을 담당한다.

Round: Round를 담당한다.

Match: Match를 담당한다. Match에서 진행된 Round 결과를 history로 저장하고 관리한다.

Tournament: Tournament를 담당한다.

Game: 메뉴 화면을 포함하여 모든 게임을 제어한다. main함수에서 하나의 Object만 선언된다.

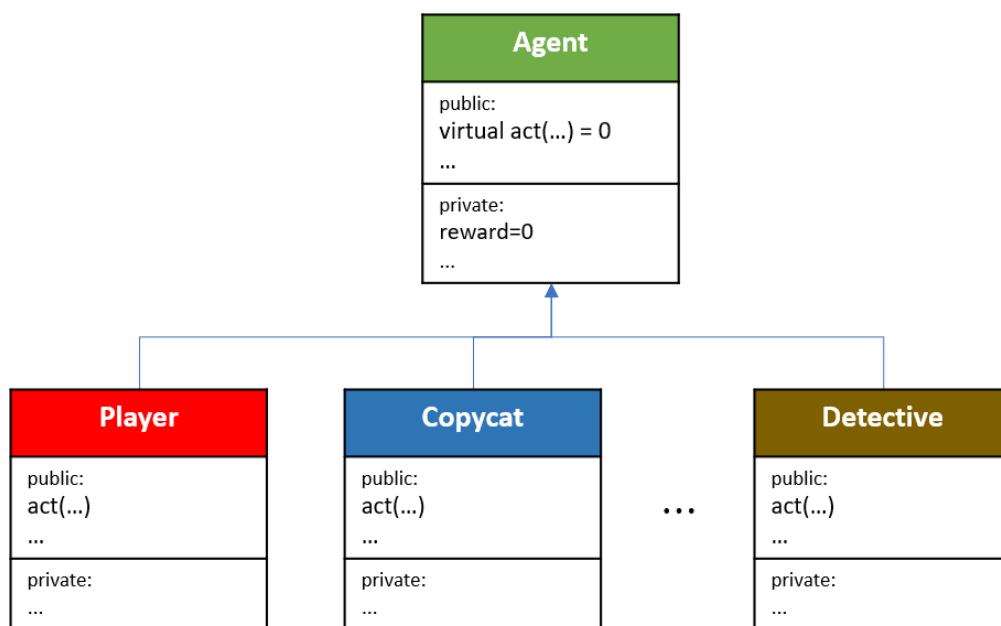


Figure 24. Agent Class 관계도

위에 명시된 Class는 반드시 선언하여야 하며, 추가적인 Class 정의도 가능하다. 특히 Population class는 구현자가 원하는 자료구조 Class를 정의하고 이를 상속받아 Agent를 관리하도록 구현하여야 한다. Assignment2에서 구현했던 DoublyLinkedList를 참고하는 것도 좋은 방법이다. LinkedList 계열의 자료구조를 사용하기 위해서는 Agent를 멤버로 가지는 별도의 Node class를 정의하여야 할 것이다. 자신이 사용한 자료구조에 대한 내용을 보고서에 필히 포함하도록 한다.

채점 기준

1. 프로그램 기능 – 60%
 - 시작 메뉴 5%
 - Single Match 10%
 - Single Tournament 15%
 - Repeated Tournament 30%
2. 프로그램 설계 및 구현 – 30%
 - Single Match 10%
 - Single Tournament 10%
 - Repeated Tournament 10%
3. 프로그램 가독성 – 5%
4. 보고서 구성 및 내용, 양식 – 10%