

2016년도 1학기

동국대학교

심화 프로그래밍(CSE2022)

2016.05.26. | 조교 송치원

# 1 우리가 아는 '짝 맞추기' 게임

- 같은 그림의 패 2개를 맞추는 게임



## 2 우리가 할 '짝 맞추기' 게임

- 목표 : 하나의 게임 보드에서 카드의 짝을 많이 맞추자!
- 총 플레이어의 수 : 2명
- 플레이 방식 : 턴

## 2 우리가 할 '짝 맞추기' 게임

- 한 턴에 이루어지는 행동

1.  $(x_1, y_1)$  위치의 패를 뒤집는다.
2.  $(x_1, y_1)$  위치의 패가 무엇인지 두 플레이어에게 알린다.
3.  $(x_2, y_2)$  위치의 값을 확인한다.
4.  $(x_2, y_2)$  위치의 패가 무엇인지 두 플레이어에게 알린다.
5. 두 위치의 패가 동일할 경우
  - ① 현재 턴의 플레이어의 점수를 올린다.
  - ② 두 위치가 어떤 값으로 매칭되었는지 두 플레이어에게 알린다.
  - ③ 현재 턴의 플레이어가 1번부터 시작한다.
6. 두 위치의 패가 동일하지 않을 경우
  - ① 다음 턴의 플레이어가 1번부터 시작한다.

### 3 무엇을 해야 하는 걸까요?

- 지뢰 찾기와 마찬가지로 전체적인 틀은 구현되어있습니다.
- 앞에서의 룰에 따라 플레이 했을 때 상대방을 이길 수 있는 Logic 구현
- **본인의 생각을 구현해내는 능력을 길러봅시다.**
- 간단한(심플한) AI를 구현하여 챔피언에 도전해보세요.
- 실제로 제출한 과제로 대회가 치러집니다.

## 4

## 코드를 알아보기에 앞서

- 전체 프로그램 코드를 이해해야만 할 수 있는 과제가 절대 아닙니다.
  - 전체를 몰라도 어떤 기능을 구현해야 하는지를 명확히 알면 됩니다.
  - **실제로 게임을 한다면 어떤 패를 뒤집을지에 대해서 잘 고려해보세요.**
  - 본인이 정리되지 않은 것을 코드로 옮기긴 어렵습니다.
  - 어떻게 하면 '**조금 더**' 효율적일지 생각해보세요.
- 
- 문제를 소문제로 쪼개고 각 소문제를 어떻게 해결하는 게 좋을지에 대해서 고민해보세요

- `#define SAME_CARD_COUNT 8`
  - 한 숫자당 몇 개의 짝이 존재하는지
  - 반드시  $\text{BOARD\_SIZE}^2$ 의 약수로 지정해주세요.
- `#define BOARD_SIZE 100`
  - 보드의 크기
- `#define LAST_CARD_NUMBER`  
`(BOARD_SIZE*BOARD_SIZE)/SAME_CARD_COUNT`
  - 1~LAST\_CARD\_NUMBER의 숫자가 나옴

# Point.h

- Point(int,int)
  - x, y좌표를 매개변수로 받아서 바로 값을 설정하는 생성자
- int getX();
- int getY();
  - x, y좌표를 각자 반환함
- Point& operator = (const Point &p);
- bool operator == (const Point &p);
- bool operator != (const Point &p);
  - 연산자 오버라이딩
- bool checkBoardRange();
  - 설정된 x, y좌표가 보드 범위 이내인지 아닌지 판별



# Game.h

- `int board[BOARD_SIZE][BOARD_SIZE]`
  - 실제 게임 보드
- `bool matched[BOARD_SIZE][BOARD_SIZE]`
  - 매칭된 카드인지 체크
- `int getCard(Point point)`
  - Point 위치에 어떤 카드가 있는지 반환
- `bool tryMatching(Point p1, Point p2)`
  - P1과 p2위치가 매칭인지 확인

- virtual Point inputFirst()
  - 첫 번째 카드를 뒤집을 위치를 반환하면 됨.
- virtual Point inputSecond()
  - 두 번째 카드를 뒤집을 위치를 반환하면 됨.
- virtual void checkCardInfo(Point point, int card)
  - 뒤집힌 카드의 위치와 적힌 숫자를 매개변수로 넣어줌.
  - point 위치에 card가 쓰여져 있음
- virtual void matchedCard(Point p1, Point p2, int card)
  - 매칭이 된 경우 두 카드의 위치와 카드의 번호를 매개변수로 넣어줌.
  - P1, p2 위치의 패가 card란 숫자가 적혀서 매칭됨

# GameManager

- Assistant player1;
- M2012345678 player2;
  - 게임을 플레이할 player1과 player2
  - Class를 변경하여 본인이 구현한 코드의 성능을 비교하면 된다.
- Game game;
  - 실제 게임 보드
- int player1Score;
- int player2Score;
  - 플레이어들의 점수를 저장할 변수
- void play();
  - 게임을 플레이할 메소드

## 주의사항

- 입출력을 절대로 하지마세요.
  - cin, cout, scanf, printf 등 모든 입출력을 금지합니다.
- 클래스 명과 cpp파일 명과 header 파일명은 "M학번" 으로 합니다.
  - Ex) 학번이 2012345678일 경우 M2012345678
- 반드시 header 파일을 생성하여 class를 선언해주어야 합니다.
- cpp파일에는 본인이 구현한 class만이 존재하도록 합니다.
- Player.h를 상속받아서 구현합니다.

## 과제 일정

- 5월 26일 : 과제 출제
- 6월 02일 : 지뢰 찾기 대회!
  - 지뢰찾기 방법에 대해서 1분 스피칭 해주세요.
- 6월 09일 : 중간 확인 – 반드시 모두 제출하세요.
  - 조교의 AI와 대결
  - 조교의 AI에게 이기면 이번 과제에 만점!
- 6월 16일 : 최종 과제 마감
  - 1분 내외로 본인의 방법을 스피치
  - Ppt 준비하지 마세요. 간략하게 “이런 생각으로 했다.”를 전달해주세요.
  - 실제 제출 코드를 활용하여 대결 진행

# 주석과 보고서

- 주석은 절대 Line마다 달지 않는다.
- 메소드 위에 어떤 생각으로 구현한 것인지를 간략하게 적는다.
  - 해당 주석만 보고 어떤 식으로 구현된 것인지 이해 가능하도록.

```
/**  
이러이러해서 저러저러해서  
이러니 저러니 한 방법으로 함.  
*/
```

- 보고서는 각 메소드별로 위에서 작성한 주석을 그대로 적는다.

```
Point M2012345678::input()  
이러이러해서 저러저러해서 이러니 저러니 한 방법으로 함.
```

- 추가로 본인의 AI와 랜덤 AI의 대결 결과 6개를 캡처하여 붙인다.
- 과제 수행 소감을 적는다.

# 조교 vs 랜덤의 결과 내용

```
C:\Windows\system32\cmd.exe
Player1 Matched) [98,79] and [99,85] are 1234
Player1 Matched) [99,38] and [99,86] are 1237
Player1 Matched) [98,78] and [99,99] are 1217
Player1 total score : 4999
Player2 total score : 1
total turn count : 14264
계속하려면 아무 키나 누르십시오 . . .
```

```
C:\Windows\system32\cmd.exe
Player1 Matched) [99,77] and [99,90] are 1238
Player1 Matched) [99,99] and [99,93] are 1248
Player1 Matched) [99,39] and [99,94] are 1235
Player1 total score : 4998
Player2 total score : 2
total turn count : 14573
계속하려면 아무 키나 누르십시오 . . .
```

```
C:\Windows\system32\cmd.exe
Player1 Matched) [99,95] and [99,92] are 1249
Player1 Matched) [99,73] and [99,93] are 1238
Player1 Matched) [98,57] and [99,96] are 1231
Player1 total score : 4999
Player2 total score : 1
total turn count : 14422
계속하려면 아무 키나 누르십시오 . . .
```

- 조교(Player1)와 랜덤(Player2)의 3번의 경기
- 약 14,000번 정도의 확인 작업을 한다.
  - 랜덤의 경우 이미 확인한 곳도 중복으로 확인함
- 랜덤의 성능이 굉장히 나쁘다.
  - 이미 찾지 않은 2개의 위치를 맞춰야 하므로 확률이 당연히 떨어짐