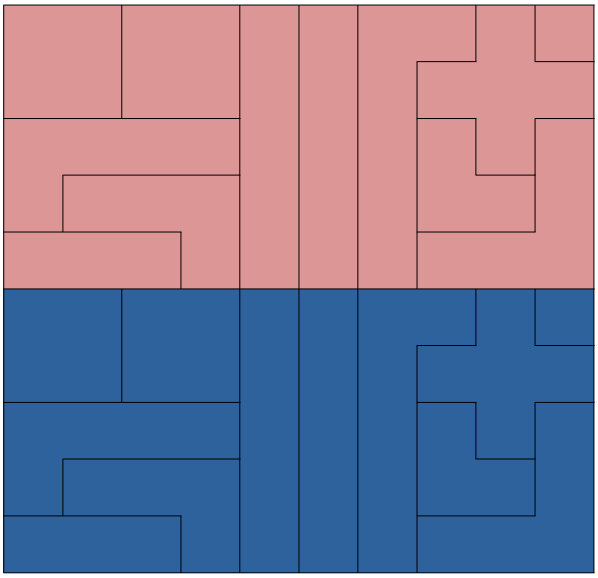
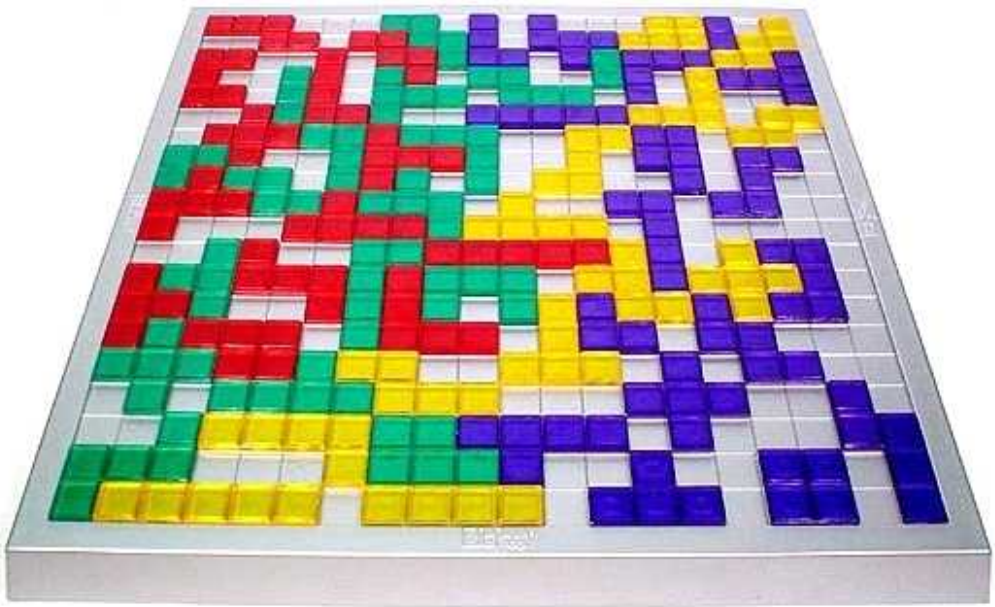
BLOKUS Champions League (v0.2)



1. Introduction

블로커스(BLOKUS)는 두 명 혹은 그 이상이 동시에 즐길 수 있는 동명의 보드게임을 의미한다.



게임의 목표는 간단하다. 일단, 각 플레이어는 동일한 개수로 이루어진 블록들을 받게 된다. (각 플레이어가 가진 블록의 종류는 모든 플레이어가 동일하다.) 플레이어는 자신의 차례에 판위에 블록을 놓게 되며, 블록을 놓은 뒤 다음 플레이어에게 차례를 넘긴다. 게임은 더 이상 놓을 블록이 없거나 더 이상 판위에 놓을 장소가 없을 때 끝나게 되며, 더 많은 칸을 차지한 플레이어가 승리하게 된다.

블록은 놓을 수 있다는 전제하에 어떤 변형도 가능하다. 예를 들어 돌리거나(rotate), 뒤집거나(flip) 할 수 있다. 물론 블록을 잘라내는 행위는 허용되지 않는다.

본 프로젝트에서 각 팀이 구현해야 하는 것은 바로 어느 위치에 블록을 놓을지 결정하는 알고리즘을 작성하는 것이다. 기본적인 API 및 규칙은 아래에 제공될 것이다.

2. 규칙

A. 판의 규칙

10 x 10의 칸으로 이루어진 가상의 판을 가정한다. 판은 2차원 배열로 제공되며, 배열의 각 위치에 따른 판의 지점은 다음과 같다.

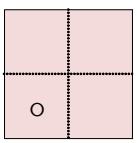
(0,0)	(0,1)	(0,2)	(0.3)	(0,4)	(0,5)	(0,6)	(0,7)	(0,8)	(0,9)
(1,0)	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	(1,7)	(1,8)	(1,9)
(2,0)	(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)	(2,7)	(2,8)	(2,9)
(3,0)	(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(3,6)	(3,7)	(3,8)	(3,9)
(4,0)	(4,1)	(4,2)	(4,3)	(4,4)	(4,5)	(4,6)	(4,7)	(4,8)	(4,9)
(5,0)	(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)	(5,7)	(5,8)	(5,9)
(6,0)	(6,1)	(6,2)	(6,3)	(6,4)	(6,5)	(6,6)	(6,7)	(6,8)	(6,9)
(7,0)	(7,1)	(7,2)	(7,3)	(7,4)	(7,5)	(7,6)	(7,7)	(7,8)	(7,9)
(8,0)	(8,1)	(8,2)	(8,3)	(8,4)	(8,5)	(8,6)	(8,7)	(8,8)	(8,9)
(9,0)	(9,1)	(9,2)	(9,3)	(9,4)	(9,5)	(9,6)	(9,7)	(9,8)	(9,9)

place_to() 함수를 통해서 블록을 놓을 때는 아래에 기술될 블록의 시작지점과 블록의 종류를 인자로 넘겨주면 된다.

B. 블록의 종류

블록의 종류는 다음과 같다.

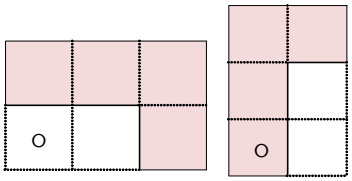
1) SQUARE



4개의 칸으로 이루어진 정사각형 모양의 블록이다. 이 형태의 블록은 총 2개 제공된다. 시작지점은 왼쪽 아래('o'로 표시되어 있는 부분)의 블록이다. 예를

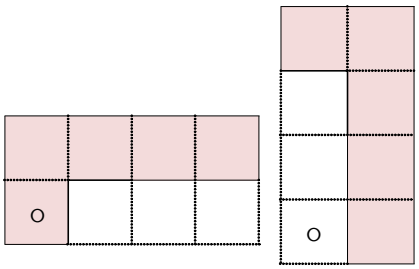
들어 시작지점이 (2,0) 지점에 있다면 전체 블록이 차지하는 좌표는 (2,0) (1,0) (2,1) (1,1)이 된다.

2) HANDGUN



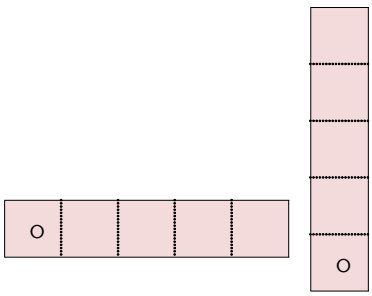
4개의 칸으로 이루어진 권총 모양의 블록이다. 이 형태의 블록은 총 1개 제공된다. 기본적으로 3x2 혹은 2x3의 공간을 차지하는 블록으로 가정한다. 시작 지점은 3x2 혹은 2x3 블록의 왼쪽 아래 지점이다.

3) SHOTGUN



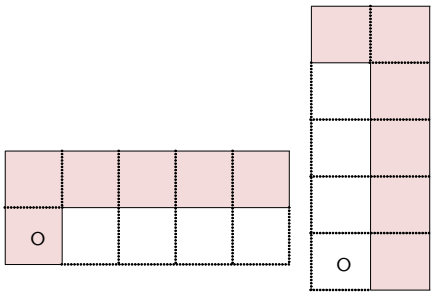
6개의 칸으로 이루어진 샷건 모양의 블록이다. 이 형태의 블록은 총 1개 제공된다. 전체 블록이 차지하는 공간은 4x2 혹은 2x4의 공간이다. 시작 지점은 4x2 혹은 2x4 블록의 왼쪽 아래 지점이다.

4) WORM



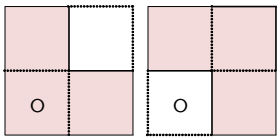
5개의 칸으로 이루어진 지렁이 모양의 블록이다. 이 형태의 블록은 총 2개 제공된다. 전체 블록이 차지하는 공간은 5x1 혹은 1x5의 공간이다. 시작지점은 가장 왼쪽 혹은 가장 아래쪽이다.

5) CROWBAR



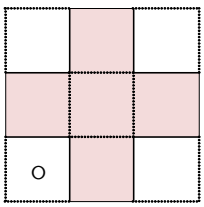
6개의 칸으로 이루어진 쇠지렛대 모양의 블록이다. 이 모양의 블록은 총 1개 제공되며, 5x2 혹은 2x5의 공간을 차지한다. 시작지점은 왼쪽 아래지점이다.

6) CHAIR



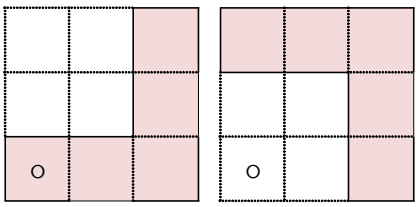
3개의 칸으로 이루어진 앉은뱅이 의자 모양의 블록이다. 이 모양의 블록은 총 1개 제공되며, 2x2의 공간을 차지한다. 시작지점은 왼쪽 아래지점이다.

7) CROSS



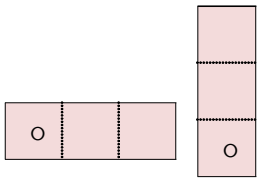
4개의 칸으로 이루어진 십자 모양의 블록이다. 이 모양의 블록은 총 1개 제공되며, 3x3의 공간을 차지한다. 시작지점은 왼쪽 아래지점이다.

8) BOOMERANG



5개의 칸으로 이루어진 부메랑 모양의 블록이다. 이 모양의 블록은 총 1개 제공되며, 3x3의 공간을 차지한다. 시작지점은 왼쪽 아래지점이다.

9) STICK



3개의 칸으로 이루어진 막대기 모양의 블록이다. 이 모양의 블록은 총 1개 제공되며, 3x1 혹은 1x3의 공간을 차지한다. 시작지점은 왼쪽 지점 혹은 가장 아래 지점이다.

10) DOT

1개의 칸으로 이루어진 블록이다. 이 모양의 블록은 총 1개 제공되며, 1x1의 공간을 차지한다.

C. 게임의 규칙

각 플레이어는 5초 내에 모든 작업을 끝내야한다. 시간을 측정하여 5초 이상 걸릴 경우, 실격패로 한다.

모든 경우에 타 플레이어의 블록을 침범할 경우, 실격패로 처리한다. 매 차례 종료 후, 판을 검사하여 이를 적발한다. 블록이 판의 경계를 벗어나도 실격패로 처리한다.

게임은 양쪽 플레이어 모두가 더 이상 놓을 블록이 없을 때 종료된다. 종료된 시점에서 각 플레이어가 차지한 말판의 칸 수를 세어 더 많은 칸을 차지한 플레이어가 승리하게 된다.

가능한 모든 플레이어의 조합에 대해서 3회씩 게임을 진행한다. 그리고 가장 승률이 높은 플레이어부터 차례대로 순위를 매긴다.

같은 색의 블록은 반드시 꼭짓점으로 이어져야 한다. 다음 그림을 보라.

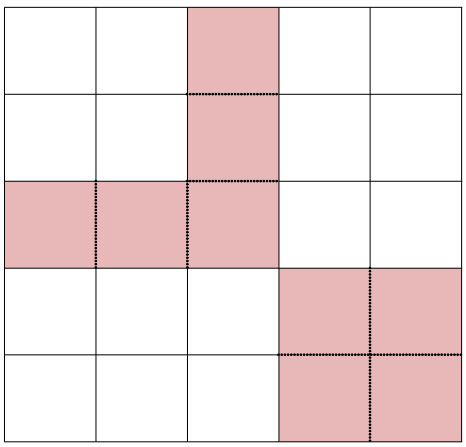


표 20 올바른 경우

위 그림은 같은 색의 블록이 꼭짓점으로 잇닿아 있으므로 규칙에 어긋나지 않는다.

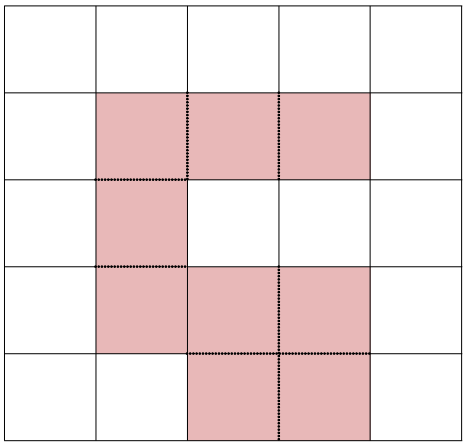


표 21 잘못된 경우

위 그림은 서로 같은 색의 블록이 모서리로 연결되어 있으므로, 규칙에 어긋난다.

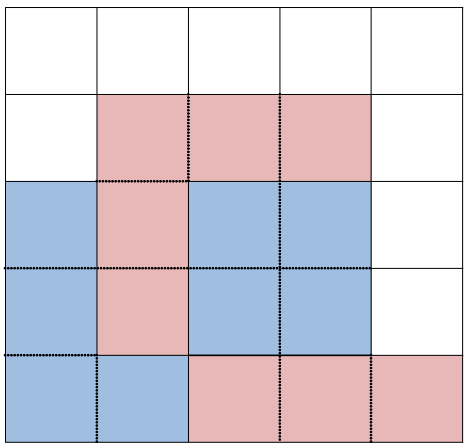


표 22 올바른 경우

위 그림은 서로 다른 색을 지닌 네 블록이 규칙을 준수한 형태로 놓여있는 것을 나타내고 있다. 각 플레이어는 이를 반드시 준수하여 블록을 놓아야 하며, 만약 닿을 수 있는 꼭짓점이 없을 경우 더 이상 둘 수 있는 공간이 없는 것으로 간주한다. 만약 규칙에 어긋난 형태로 블록을 놓을 경우 실격패로 판정된다.

D. 프로그램의 규칙

place_to() 함수를 사용할 경우, 다른 블록을 침범했을 때 에러코드(OVERLAP)를 리턴한다. place_to가 리턴하는 에러코드는 다음과 같다.

Error code	Value	Description
NOERROR	0	Normal exit
OVERLAP	1	Violate the enemy's block
OUT_OF_STAGE	2	Is out of stage

Table 1 Error code

각 팀은 블록을 놓을 자리를 결정하는 함수를 작성한다. 함수는 정해진 값을 리턴해야 하며, 리턴 코드는 다음과 같다.

Return code	Value	Description
PLAYING	0	Player in normal state
OUT_OF_SPACE	1	No one can be placed into the stage
OUT_OF_BLOCK	2	All blocks are placed

Table 2 Return code

3. API

A. place_to()

Function	place_to
Input	starting_point, block
Output	error_code
Description	입력으로 블록이 시작해야할 위치와 블록을 받는다. 해당 위치에 놓을 수 있으면, 위치시킨 후 NOERROR 코드를 리턴하고, 놓을 수 없다면 그에 해당하는 error code를 리턴한다.

B. rotate()

Function	rotate
Input	block, rotate_degree
Output	해당사항 없음
Description	블록을 회전시킨다. 입력으로 블록과 회전시킬 정도를 입력받는다. 정도는 미리 정의된 값인 SINGLE, DOUBLE, TRIPLE을 통해서 입력가능하며, SINGLE의 경우 90도, DOUBLE일 경우 180도, TRIPLE일 경우 270도를 회전하게 된다. 주의해야할 것은 2x4 블록의 경우 90도 회전시켰을 때 4x2 블록이 된다는 것이다.

C. flip()

블록을 뒤집는다. 입력으로 블록과 동작 지정 번호를 입력받으며, 1일 경우 좌우로, 2일 경우 상하로 뒤집게 된다.

Function	flip
Input	block, flip_directive
Output	해당사항 없음
Description	블록을 뒤집는다. HORIZONTAL 지시자를 넘겨주었을 경우 좌우로, VERTICAL 지시자를 넘겨주었을 경우 상하로 뒤집는다.

D. get_blocks()

Function	get_blocks
Input	team color
Output	block set
Description	블록을 가져온다.

E. get_stage()

Function	get_stage
Input	해당사항 없음
Output	stage
Description	게임이 진행되고 있는 판을 가져온다. 플레이어는 자신의 차례에서 이 함수를 통해 게임판을 가져올 수 있다.

F. print_blocks()

Function	print_blocks
Input	block set
Output	해당사항 없음
Description	현재 남아있는 블록들과 상태를 출력한다. 프로젝트 수행 중에만 쓸 수 있는 상태 확인용 함수이며, 제출 시에는 이를 삭제해야한다.

G. print_block()

Function	print_block
Input	block
Output	해당사항 없음
Description	현재 블록의 상태를 출력한다. 프로젝트 수행 중에만 쓸 수 있는 상태 확인용 함수이며, 제출 시에는 이를 삭제해야한다.

4. Revision history

연번	작성자	내용	날짜
1	김성훈	초안 작성	2013.05.26
2	김성훈	세부 규칙 추가	2013.05.29