

이번 기말고사에서는 최종 제출 파일이 컴파일에러가 발생할 경우 0점입니다.
controller.cpp의 생성자에 **srand**부분에 자신의 학번을 입력으로 넣어주세요.

Refactoring (Easy)

1. Int color -> enum Color를 만들어서 처리한다.
 - Color 는 GREY, RED, GREEN, BLUE의 값을 가진다.
 - Color는 block의 public에 선언한다.
 - Block의 색상을 저장하는 color 변수는 private에 둔다.
2. Block 클래스에서 block::Color num_to_color(int num) 을 만들어 Color를 얻을 수 있게 함
0 -> GREY, 1 -> RED, 2 -> GREEN, 3 -> BLUE
Num_to_color는 static function
 - Hint : Controller.cpp 에서 함수를 사용
3. block_group의 불필요한 Color 관련 함수 제거
 - 사용되지 않는 color 관련 함수 제거
4. Shape_block 클래스에서 enum shape 선언
 - Shape의 값은 FOLD, TREE, CROSS의 값을 가짐
 - Shape_block에서 static 함수로 num_to_shape 정의
5. Make 함수를 shape_block으로 이동 (virtual -> 일반함수) (cross_block, fold_block, tree_block)
6. Drop 함수를 shape_block으로 이동 (virtual -> 일반함수) (cross_block, fold_block, tree_block)
7. Merge 함수를 shape_block으로 이동 (virtual -> 일반함수) (cross_block, fold_block, tree_block)
 - 4-6 함수의 구현은 shape_block.cpp 파일을 생성하여 구현할 것
 - 현재와 같은 기능을 하게 만듦

추가 기능

8. Shape_block 에서 static 함수를 통해 create 함수 생성
 - Controller.cpp에서 사용되는 make_block함수에서 current_block 값을 shape_block::create로 생성하도록 변경
 - Shape_block::create에서는 4에서 정의한 shape을 받도록 함 (num_to_shape 이용)
9. Grey 블록이 터질 때 연결된 grey 블록 모두 터지게 기능 변경
10. 블록이 터질 때 블록 개수의 2의 n승으로 점수 얻게 변경 + 회색 블록도 점수 얻게 변경

block.h, block.cpp

1x1 크기의 블록에 대한 위치와 색상 정보를 가지고 있는 클래스.

color: grey, red, green, blue의 색상을 저장하는 변수 (기존 코드는 int 값으로 저장)

x, y: x 좌표와 y 좌표를 저장하는 변수

Getter함수

get_color: private 필드인 색상(color) 값을 얻는 함수 (기존 코드는 int type return)

get_x: private 변수인 x 값을 얻는 함수

get_y: private 변수인 y 값을 얻는 함수

can_left: left를 실행할 수 있는 지 확인하는 함수 (왼쪽으로 블록이 더 이동할 수 있는지)

can_right: right를 실행할 수 있는 지 확인하는 함수 (오른쪽으로 블록이 더 이동할 수 있는지)

can_down: down을 실행할 수 있는 지 확인하는 함수 (아래로 블록이 더 내려갈 수 있는지)

move(x, y): x, y 좌표로 이동

change(block*): 입력받은 블록과 색상을 교체함

make: 현재 블록정보(x, y 좌표, 색상)와 같은 블록을 board의 puyo_board에 생성하는 함수

left: 왼쪽으로 한칸 이동

right: 오른쪽으로 한칸 이동

down: 아래로 한칸 이동

same_x(block*): 입력받은 블록이 같은 x축에 있는지

same_y(block*): 입력받은 블록이 같은 y축에 있는지

block_group.h, block_group.cpp

바닥에 쌓인 블록 중 같은 색상으로 연결된 블록을 관리하는 블록
size가 4가 될 경우 폭발

block_set: 같은 색상인 연결된 블록을 저장하는 set

color: grey, red, green, blue의 색상을 저장하는 변수 (기존 코드는 int 값으로 저장)

insert(block*): 입력받은 블록을 block_set에 추가

remove: 블록 set에서 임의로 하나의 블록을 제거해 return

Getter

get_size: block_set의 크기를 return

get_color: color 값을 return

explosion: 그룹에 포함된 모든 블록을 puyo_board에서 폭발시키는 함수

shape_block.h

특정모양의 블록에 대한 abstract 클래스

cross_block, fold_block, tree_block (shape_block 상속)

십자, 꺾인 모양, 일자모양의 블록

block_vector

각 모양의 블록 정보(x, y좌표, 색상)를 가진 block*를 블록 벡터에 저장하여 가지고 있음

can_left, can_right, can_down, can_rotate

해당 모양의 블록이 왼쪽, 오른쪽, 아래로의 이동이나 회전이 가능한 지 확인하는 함수

left, right, down, rotate

해당 모양의 블록이 왼쪽, 오른쪽, 아래로의 이동이나 회전을 실행하는 함수

drop

모양의 블록에 포함된 block*를 더이상 아래로 이동할 수 없을 때까지 모두 아래로 이동시킴

make

block 클래스의 make를 이용해 block_vector의 모든 block*를 puyo_board에 생성

merge:

shape_block의 블록들이 모두 내려갔을 때 기존에 쌓여있던 다른 블록들과 같은 색상인지 확인해 block_group에 포함시키는 함수

board.h, board.cpp

block_array: 블록이 쌓일 2차원 배열

explosion_map<block, block_group>: block이 어떤 블록 그룹에 포함되었는지 확인하는 map

block_groups: block_group들을 관리하는 set

score: 현재 점수

starting_x, y : shape_block이 처음 생성될 때의 중심 위치 x, y 좌표 (cross, fold, tree block 참고)

static puyo_board: board를 static공간에 저장하는 변수, 게임에서 필요한 보드 정보를 모두 관리

function

can_make: type에 맞게 각 블록을 생성할 수 있는 지 확인하는 함수

can_move(x, y): 입력받은 x, y 좌표로 블록이 이동할 수 있는지 확인하는 함수

is_block(x, y): 입력받은 x, y 좌표에 블록이 있는지 확인하는 함수

make(x, y, block): 입력받은 x, y 좌표에 블록을 추가

merge_group(cur_block, adj_block)

현재 블록과 인접한 블록이 색상이 가고 각각 그룹을 가지고 있을 때 두 그룹을 합치는 함수

_merge(a_x, a_y, block)

x, y좌표의 블록과 block이 같은 색상일 경우 그룹을 merge

(merge_group함수 이용)

merge(block)

블록의 인접 블록을 모두 확인하여 같은 색상의 블록이 있을 경우 group을 합침

(_merge함수 이용)

move_block(old_x, old_y, new_x, new_y)

old_x, y에 있는 블록을 new_x, y로 이동

explode_adjacency_grey(x, y): x, y 좌표의 인접 회색블록을 폭발시키는 함수

explode_grey(x,y): x,y의 회색블록을 폭발시키는 함수

explode_block(x, y): x, y의 블록을 폭발시키는 함수

explode: block_groups에 포함된 group을 모두 확인해 size가 4 이상인 group을 폭발시킴

merge_all: explode가 일어난 후 groups, explosion_map을 reset후 다시 merge함

drop: explode 후 공중에 떠있는 모든 블록을 아래로 떨어뜨리는 함수

get_score: 점수를 반환하는 함수

to_string: 블록 보드의 모양을 문자열로 반환하는 함수

controller.h, controller.cpp

count: 현재 게임의 블록 count

current_block

현재 조작중인 블록

make_block

current_block 을 count 에 맞게 생성함

left, right, down, drop, rotate

현재 조작중인 블록에 대해 left, right, down, drop, rotate 를 실행할수 있으면 실행

drop 의 경우 drop 후 merge 를 실행함

can_move: current_block 을 더 움직일 수 있는 지 확인하는 함수 (down 이 불가능할 경우 false)

explode: current_block 을 free 시키고 explode 함수를 호출 (터질 수 있을 때만 터짐)

print: 블록 보드 상태를 출력

start:

current_block 을 생성하고 move 함수를 호출해 조작

더이상 블록을 생성할 수 없을 때까지 반복함

move:

더이상 current_block 을 움직일 수 없을 때 까지 키보드 입력을 받으며 current_block 을 이동

더이상 움직일 수 없으면 공중에 떠있던 블록도 더이상 내려올 수 없을 때 까지 down 해 공중에 떠있는 블록이 없게 함