**C++ Console Tetris Personal Project**

1. **Early Approach**

* 모양 별 4\*4 배열을 만들어 Block모양 표현(블록모양일 경우 2, 공백일 경우 0으로 표현)
* Map의 범위를 설정(가로20, 세로20)
* block이 들어갈 빈 공간은 0, 벽일 경우 1로 설정하여 벡터 멤버변수인 canvas 초기화
* keyboard 입력 값에 따라서 block 모양 배열을 canvas에 저장
* 일정시간(2초)에 따라 block이 내려가는 것을 고려하여 canvas에 저장
* 움직이는 것을 표현하기 위해 canvas를 초기화시킨 후, 위치가 변경된 block을 저장하고, 다시 출력하는 과정 반복
* block이 벽 또는 다른 block을 만났을 때 == Map의 canvas벡터값이 3 || 4가 되는 경우

이 정도의 Early Approach 후에 바로 personal project 진행

1. **Problem & Solution**

**모양 별 4\*4 배열을 만들어 Block모양 표현 시 block rotation issue**

* block rotation의 경우 새로 4\*4 배열을 4개 더 고려해야 하기 때문에 4\*4\*4 배열로 변경

**map에 block의 위치를 표현할 때, 매번 배열을 가져와서 저장시키는 issue**

* Block class를 생성하여 멤버변수로 x,y 위치 값 저장(모양 별 block[0][0]의 값으로 설정), 메소드 함수로 x,y값에 따라 원하는 block 값을 map에 저장할 수 있도록 설정
* Block class 생성 시, 모든 block이 공통적으로 지니는 x,y 멤버변수, x,y값이 이동하는 메소드 함수를 가진 Block 부모클래스 생성 후, 유도클래스에 각 block들의 4\*4\*4 모양배열을 저장하는 메소드 함수 추가

**block이 벽에 붙어있고, rotation 할때 범위 초과 issue**

* 벽에 붙어있을 때, tempx, tempy 변수를 통해 벗어나는 범위만큼 다시 밀어내는 방식 사용

**block이 다른 block들 사이에 있을 때, rotation 시 block간 겹치는 issue**

* Block을 움직였을 때 canvas에 4가되는 지점이 있다면 rotation 불가능하도록 설정

**map의 위치에 block 값을 저장할 때 block 모양값(2)은 canvas범위에 벗어나지 않는데 공백값(0)이 벗어나는 index compile issue**

* (초기) tempx, tempy변수를 통해 block 모양 별 x,y값의 시작과 끝의 값을 저장 후 tempx, tempy 변수가 map의 범위를 벗어날 경우 움직이지 않게 설정

> 코드의 복잡성 문제

* (최종) 배열의 공백값이 범위를 넘어가면 continue, 배열값을 map에 저장 후 3 || 4 인 값이 있으면 움직이지 않게 설정

**block 저장에 관한 issue**

* block의 움직임을 표현하기 위해 map을 초기화시키고 block의 좌표 변경 후 다시 출력시키는 과정에서 이미 내려간 block들을 저장하기 위해 canvas2 벡터 변수 생성.

내려간 변수들을 canvas2에 저장 후 canvas를 초기화시킨 후 다시 저장하며 출력

1. **Additional Implementations**

block이 움직일 때, block이 떨어질 위치를 미리 표시해주는 preview 함수

* map이 초기화되고 다시 출력될 때마다 canvas의 y좌표를 처음부터 끝까지 내려보내며 block배열의 값과 map의 y+1 합의 3 || 4가 되는 순간 map에 출력시키게 함.

1. **Explanation about File**

<map.h File>

* 블록들을 표현할 수 있는 공간을 마련하는 Map class를 담은 파일

<baseblock.h File>

* 세부적인 Block class들로 나누기 전 Block class의 기본적인 위치 변수와 메소드들을 가지고 있는 부모 클래스를 담은 파일

<block.h File>

* Block들을 모양별로 나눈 개별적인 class를 담은 파일

<keyboard.h File>

* 키보드의 입력에 따라 블록들을 움직일 수 있는 keyboard 함수를 담은 파일

<game.h File>

* Block들이 조건에 맞게 움직이며 게임을 진행할 수 있게 하는 game 함수, save 함수, preview 함수를 담은 파일

<main.cpp File>

* 5개의 HeaderFile에 속한 class와 함수들을 이용해 게임을 실행하는 cpp파일