

Project #3: 상속과 다형성을 이용한 테트리스 게임 완성 report

컴퓨터공학부 20163113 박희승

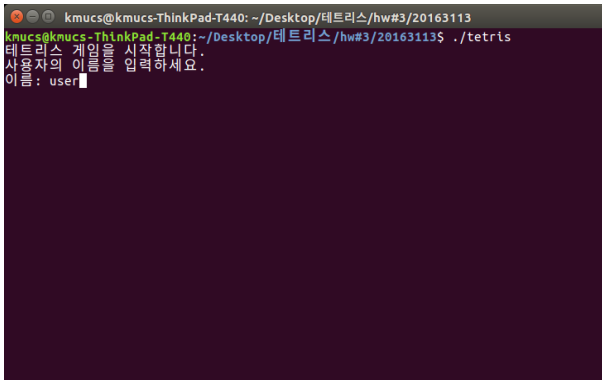
1. 과제설명

상속과 다형성을 이용한 객체지향 설계를 통해 총 7개의 테트로미노 블록이 무작위로 생성되는 테트리스 게임을 완성하였습니다.

2. 실행화면

입력 파일 없이 프로그램을 구동하는 경우

(1)키보드로 부터 사용자의 이름을 입력받는다.

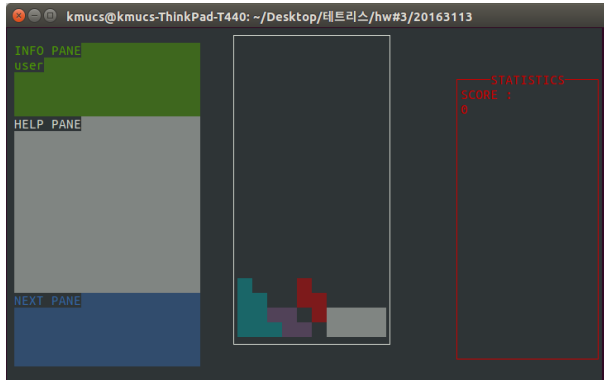


(2) 게임이 시작되면 무작위로 블록이 생성되어 일정시간 마다 자동으로 떨어진다. 키보드로 블록을 움직일 수 있다.



- ← (왼쪽 방향키) - 블록을 왼쪽으로 한 칸 이동
- (오른쪽 방향키) - 블록을 오른쪽으로 한 칸 이동
- ↓ (아래쪽 방향키) - 블록을 아래로 한 칸 이동
- ↑ (위쪽 방향키) - 블록을 반시계 방향으로 회전
- SpaceBar - 블록을 최하단으로 이동
- q (영문자 q키) - 게임 종료

(3) 블록이 쌓여 가로 줄이 빈틈 없이 채워지게 되면, 채워진 줄은 없어지고 그 위에 남은 블록은 아래 이동하고 없어진 줄 수만큼 점수가 올라간다.



(4) q (영문자 q키) 또는 블록이 쌓여 게임 화면의 가장 윗줄에 닿으면 게임은 강제 종료 된다.

q (영문자 q키)를 눌렀을 때



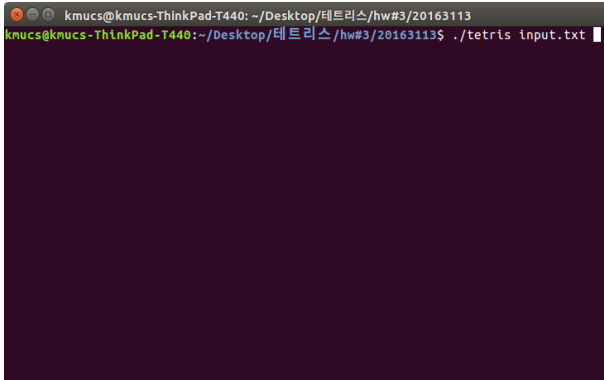
블록이 가장 윗줄에 닿았을 때



입력 파일로 프로그램을 구동하는 경우

(1) 입력값은 [사용자 이름] [seed 값] [블록 위치 조절 ...] 순으로 되어있다.

user 0 g t l l l l d l d r d r r r d q



r - 오른쪽으로 한 칸 이동
g - 아래로 한 칸 이동
t - 블록을 반시계 방향으로 회전
d - 최 하단으로 이동
q - 게임 종료

입력받은 사용자 이름과 seed로 게임이 설정된다.

(2) 텍스트 파일을 통해 리플레이 형태로 진행되는 게임 모드에서 게임의 최종 결과가 아니라 **게임이 진행되는 과정이 모두 보이도록 하는 기능**으로 구현하였다. (리플레이에서도 블록이 자동으로 떨어지는 설정으로 해놓음)



3. 코드설명

[Main.cpp]

- int num 점수를 저장하는 변수
- char username[8] username을 저장하는 변수
- int color 현재 이동중인 블록의 숫자를 저장하는 변수
- int check 블록이 쌓일 시 알려주는 변수
- int block[4][2] 현재 이동중인 블록의 좌표
- int board[21][12] 테트리스 보드판
- *[21][12]로 해준이유는 충돌검사를 위해 왼쪽,오른쪽과 아래쪽 테두리에 1을 넣어주었다.
- 입력값이 있을 경우 username과 seed를 받고 srand(seed)로 seed값을 설정한다.
게임 진행 과정이 모두 보이도록 키 값을 입력받고 usleep(400000)을 넣어준다.
- 키보드로 입력값을 받을 경우 username을 입력받은 후 키 값을 입력받는다.
seed값 설정은 srand(time(NULL))으로 해준다.
kbhit()라는 함수를 사용하여 키보드 입력이 없을 때도 일정시간 마다 블록이 떨어지도록 하였다.

[Block.cpp]

- draw() 전역변수에 있는 보드판과 현재 움직이고 있는 block을 그려주는 함수이다.
- moveblock(char c) 키보드 입력값을 받아 블록을 움직여준다.
- clear() 가로 줄이 가득 찰 경우 해당 보드판을 0으로 바꿔주고 한칸씩 내려준다.
- gameover() 보드판에 가장 위에 숫자가 생길 경우는 게임오버이므로 isGameOver을 true로 바꿔준다.
- tetromino() **가상함수**로 상속받을 각 블록 클래스들에게 오버라이딩된다. 게임이 시작할 때와 블록이 쌓였을 때 새로운 블록이 생성될 때 초기 좌표값을 설정해준다.
- rotate() **가상함수**로 상속받을 각 블록 클래스들에게 오버라이딩된다. 충돌 검사와 동시에 회전 기능을 한다.

[Tetris.cpp]

- 7가지의 블록 클래스를 동적 할당한다.
- play(char input) new_block_은 랜덤으로 얻은 한 개의 블록 주소값을 가진다. 포인터를 사용하여 블록을 움직여 주고 그려준다. StatPane_->draw()를 해주어 점수판을 계속 그려준다. 블록이 쌓일 경우 전역변수 check가 1이 된다. check가 1일 경우 0으로 바꿔주고 NEW_Block()함수로 새로운 블록을 만들어준다.
- NEW_Block() 난수를 이용해 7가지의 블록중에서 한개의 주소값을 new_block_에 넣어준다.
- result() 게임오버시 결과창을 draw해준다.
- updateScreen() 게임시작시 전체화면을 그려주고 처음시작할 블록 하나를 생성해준다.

[O_Block.cpp, I_Block.cpp, Z_Block.cpp, L_Block.cpp, J_Block.cpp, S_Block.cpp, T_Block.cpp]

- Block클래스를 상속받은 클래스들이다.
- tetromino() 블록 초기좌표를 설정해준다.
- rotate() 회전시킬 각각의 좌표들이 0인지 모두 판별 후 모두 0이면 회전해준다.

[Pane.cpp]

- draw() **가상함수**로 상속받을 각 Pane 클래스들에게 오버라이딩된다.

[InfoPane.cpp, HelpPane.cpp, NextPane.cpp, BoardPane.cpp, StatPane.cpp, ResultPane.cpp]

- draw() 각 win_좌표에 window를 그려준다.

4. 느낀점

생소한 상속과 다양성을 이용하여 테트리스를 구현하는데 많은 어려움이 있었지만 오히려 어려움을 해결하면서 많은 공부가 된 것 같습니다. 처음에는 코드들을 분할하여 구현하는 것에 “굳이 왜 불편하게 나눠서 코딩하지”라는 생각이 있었는데 코드가 점점 길어질수록 코드를 분할하여 작업하는 것이 보기도 쉽고 디버깅에도 편하다는 것을 알게 되었습니다. 또한 가상함수를 이용해 하나의 함수로 여러가지 기능을 할 수 있다는 장점을 알 수 있었고 c++에서 가장 중요한 다양성에 대해 많은 공부를 할 수 있었습니다.