8주차 예비보고서

전공: 아트엔테크놀로지 학년: 3학년 학번: 20191172 이름: 함승우

1. 테트리스 게임의 flow chart 및 각 함수의 기능에 대한 설명

우선 테트리스 게임은 두 개의 코드가 독립적으로 진행된다. play()함수와 blockDown() 함수인데, play() 함수는 user의 입력에 의해 실행되는 코드로써, 이 프로그램에서는 keyboard input에 따라 실행된다.

1. play() 함수의 flow chart 및 play() 함수를 이루는 각 함수의 기능에 대한 설명

menu()함수에 의해서 1번(play), 2번(rank), 3번(recommended play), 4번(exit)에 따라서 동작이 수행되는데, 1이 입력됐을 때 play가 실행된다. play함수가 실행되면 전역변수들이 초기화되고, user로부터 입력을 받아서 키보드 상하좌우 키에 따라 블록의 이동 또는 회전 처리를 한다.

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

1) InitTetris() = void InitTetris()

테트리스 게임을 수행하기 전 변수를 초기화하는 함수이다.

게임이 실행될 때 블록이 위치하는 부분을 field라고 하는데 이 filed에 block의 유무는 1과 0으로 구분을 하게 되는데, 이 함수에서는 ‘0’으로 (채워지지 않은 상태)로 나타낸다. 그리고, 블록을 생성하는데, block은 tetris.h 파일에서 볼 수 있는데 7가지로 구성되어있다. 이를 randomly하게 설정한다. 초기 블록은 회전하지 않은 상태이기 때문에, rotate변수도 0으로 초기화한다. 블록의 위치 또한 처음 위치인 x= (Width/2) – 2, y = -1로 초기화한다.

2) GetCommand() = int Getcommand()

이 함수는 play 함수가 실행됐을 때 입력되는 방향키 및 ‘q’를 받는 함수이다.

방향키 상 = KEY\_UP

방향키 하 = KEY\_DOWN

방향키 좌 = KEY\_LEFT

방향키 우 = KEY\_RIGHT

로 받고,

‘Q’나 ‘q’를 받는다.

3) ProcessCommand() = int ProcessCommand()

이 함수는 앞서 Getcommand에서 받은 입력값들이 필드에서 가능한지 판단하는 함수이다.

Getcommand에서 받은 입력값들은 KEY\_UP, KEY\_DOWN, KEY\_LEFT, KEY\_RIGHT, ‘Q’, ‘q’이다.

이 입력값들을 CheckToMove()(추후 설명)함수를 활용해서 동작을 취한다.

KEY\_UP일 때, CheckToMove()를 활용해서 block을 시계 반대 방향으로 회전할 수 있다면, blockRotate 변수를 초기화한다.

KEY\_DOWN일 때, CheckToMove()를 활용해서 block을 아래로 이동시킬 수 있다면, blockY변수를 초기화한다.

KEY\_RIGHT일 때, CheckToMove()를 활용해서 block을 오른쪽으로 이동시킬 수 있다면, blockX변수를 초기화한다.

KEY\_LEFT일 때, CheckToMove()를 활용해서 block을 왼쪽으로 이동시킬 수 있다면, blockX변수를 초기화한다.

QUiT일 때, Quit한다.

2. BlockDown() 함수의 flow chart 및 BlockDown() 함수를 이루는 각 함수의 기능에 대한 설명

앞서 BlockDown() 함수는 사용자의 input에 관계없이 play() 함수와 독립적으로 실행되는 코드이다.

크게 말하자면, 현재의 block을 시간에 따라 한 칸 씩 아래로 떨어뜨리며, 점수를 갱신할 수 있다면 갱신하며, 더 이상 블록을 움직을 수 없을 때 gameOver을 동작시킨다.

현재 block이 CheckToMove() 함수에 따라서 움직일 수 있다면, 현재의 block의 blocky 변수를 1 더해준다. 움직일 수 없다면 blocky 변수가 -1인지 아닌지 체크한다. blocky 변수가 -1이라는 것은 테트리스 필드가 다 차서 움직일 수 없는 경우를 뜻한다. blocky 변수가 -1이라면 gameover시키고, blocky 변수가 -1이 아니라면 (단순히 block이 움직일 수 없는 경우, 땅바닥에 닿거나, 다른 block위에 위치했을 때) AddBLockFild()함수를 통해서 이 Block을 필드에 추가한다. 그 후, 라인을 지울 수 있으면 DeleteLIne() 함수를 작동시켜 score를 갱신한다. 그 후 block을 새로 설정해야 하고 초기화한 후, BlockDown함수를 계속 recursive하게 작동시킨다.

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

1) CheckToMove() = int CheckToMove()

블록이 해당 위치로 이동하거나 회전할 수 있으면 1을 아니면 0을 return 한다.

2) AddBLockToField() = void AddBlockToField()

블록의 위치 정보와 회전 정보, 모양 정보를 받아서 filed 배열에 추가한다.

이때 block의 위치인 blockX, blockY에 현재 block을 위치한다. 필드에 block이 채워져있는 경우를 1로 변환한다.

3) Deletline() = int DeleteLine()

바닥부터 완전히 채워진 라인을 찾아서 지우고, score를 갱신한다.

4) DrawChange() = void DrawChange()

이전 block 부분을 모두 0(‘.’)으로 초기화하고, 변경된 위치나 회전수에 따라서 블록을 다시 그린다.

3. 실습시간에 구현할 5가지 함수들에 대한 간단한 pseudo code를 제시하시오.

1)

|  |
| --- |
| int CheckToMove(char f[HEIGHT][WIDTH],int currentBlock,int blockRotate, int blockY, int blockX)  //블록이 해당 위치로 이동하거나 회전할 수 있으면 1, 아니면 0을 return한다.  for block의 가로 길이  for block의 세로 길이  if 현재 block(현재 회전수, 현재 block)이 존재할 때  if 변화한 block의 x 및 y 좌표가 필드 내에 없을 때  return 0  else if 변화한 block의 좌표에 field가 1일 때  return 0  위가 아니라면 return 1 |

2)

|  |
| --- |
| void DrawChange(char f[HEIGHT][WIDTH],int command,int currentBlock,int blockRotate, int blockY, int blockX)  //  ProcessCommand의 switch문을 참조하여 command 이전 블록 정보를 찾는다.  Switch(command)  1.방향키 위  이전 블록 정보 = (현재 블록의 회전수 -1)%4  2.방향키 아래  이전 블록 정보 = (현재 블록의 y좌표 - 1)  3. 방향키 왼쪽  이전 블록 정보 = (현재 블록의 x좌표 + 1)  4. 방향키 오른쪽  이전 블록 정보 = (현재 블록의 x좌표 - 1)  그 후 이전 블록 정보 부분에 ‘.’으로 채운다.  for 블록의 가로 길이  for 블록의 세로 길이  printw(“.”)  DrawBlock() 함수를 활용해서 그린다. |

3.

|  |
| --- |
| void BlockDown(int sig)  if 아래로 움직일 수 있다면  blocky 1 증가  else  gameover인 경우, block의 y좌표가 -1일 때  gameover  아니라면, 그 블록을 addblocktofield함수를 활용해서 그리고, 새로운 block을 설정하기 위해 초기화한다. initialize 함수 참조 |

4.

|  |
| --- |
| void AddBlockToField(char f[HEIGHT][WIDTH],int currentBlock,int blockRo tate, int blockY, int blockX)  //블록이 4X4이기 때문에  For 4  For 4  현재 block이 있다면  이 block이 추가된 영역의 필드를 1로 바꿔준다. |

5.

|  |
| --- |
| int DeleteLine(char f[HEIGHT][WIDTH])  // for문을 사용해서 바닥부터 정보를 읽는다.  For  For  필드의 값이 1로 가득찬 줄이 있을 때  이 줄을 없애고 위에 있는 줄의 정보들을 받아서 다시 새로 그린다. |