C언어로 테트리스 만들기 (리눅스, 콘솔)

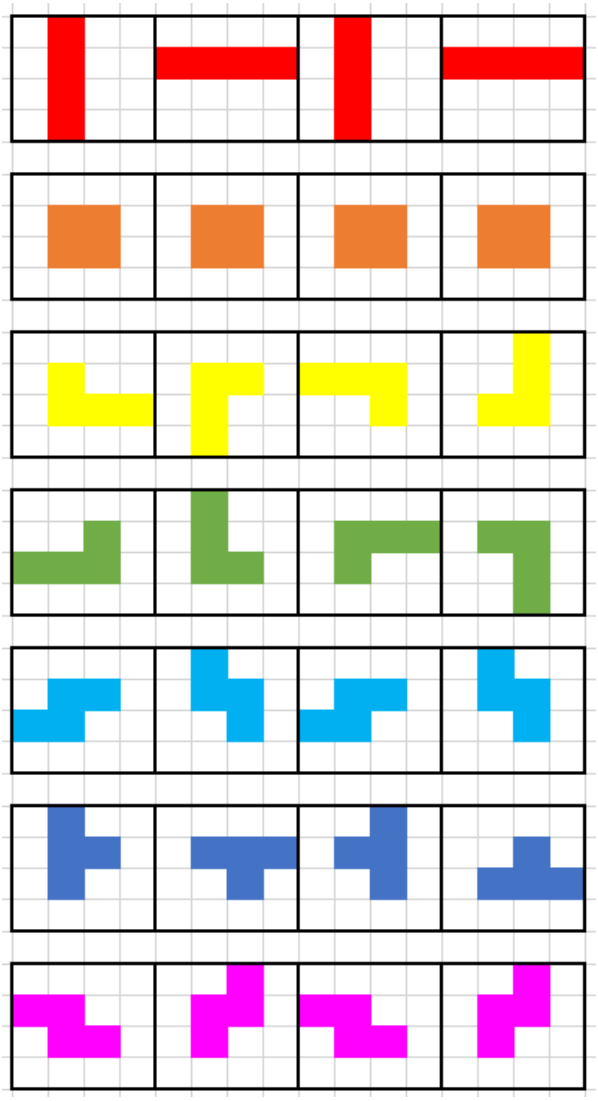
작성자: 변서연 (sybyeon1102@gmail.com)

사용자 시나리오

1. 프로그램을 실행시킨다.
2. 랜덤 블록이 위 중심에서부터 1초에 한 칸씩 내려온다.
3. ← , →를 사용하여 좌 우로 이동, ↓를 사용하여 빠른 드롭, space bar를 사용하여 블록 오른쪽으로 회전 기능들을 사용하여 블록을 적절한 위치에 떨어뜨린다.
4. (1) 만약 한 줄이 꽉 차면 그 줄이 사라진다.

   (2) 만약 새로 등장하는 블록과 기존 더미가 충돌하는 경우 종료된다.

7가지 블록 종류



해결하고 시작하기

1. 콘솔로 어떻게 화면을 지우고, 여러 요소들을 각각 출력할 수 있지? -> 콘솔 출력창의 커서를 이동한 후 해당 위치에서 출력할 수 있는 것 같다. 화면을 지우는 함수는 system("cls")이다. ->해당 코드가 작동하지 않아 system("clear")를 사용하였다.
2. 어떻게 Key를 입력 받기 위해 기다리는 동시에 화면을 재 출력할 수 있지? -> _kbhit() 함수를 사용하여 입력을 감지한다. -> _kbhit()함수가 존재하는 conio.h파일이 리눅스에 없음, curses.h도 없음.-> 구현해 놓은 함수 사용 https://indra17.tistory.com/entry/Linux-C%EC%97%90%EC%84%9C-%ED%82%A4%EB%B3%B4%EB%93%9C-%EC%9D%B4%EB%B2%A4%ED%8A%B8-%EB%B0%9B%EA%B8%B0

필요한 기능

1. 어떤 모양의 블록이 나올지 결정한다.
   - 랜덤 함수를 사용하여 7가지 블록 중 한가지를 결정한다.
2. 블록의 좌표를 이 맨 위, 중앙에서부터 0.8초에 한칸씩 아래로 내려오도록 한다.
   - 블록의 위치를 바꾼 지 0.8초가 지나면 블록의 좌표를 바꾸고 새로 출력 되도록 한다.
   - 만약 내려온 좌표에서 바닥 또는 더미와의 충돌이 있다고 판별되면 블록이 더미에 포함되도록 한다. -> 만약 가득 찬 한 줄이 있는 경우 삭제한다. -> 새 블록을 결정하여 출발시킨다.
3. 벽과 바닥과 더미와 블록을 출력한다.
4. 키의 입력이 있는 경우에는 해당 키의 기능을 수행한다.
   - ←, → : 블록을 좌, 우로 움직인다. 만약 벽과 부딪힌다면 더 이상 이동할 수 없게 한다.
   - ↓: 블록을 바닥과 충돌 전까지 이동시킨다.
   - SPACE BAR: 블록을 회전시킨다. 벽과 부딪히는 경우 좌우로 이동시켜 부딪히지 않도록 한다.

구현 순서

1. 벽과 바닥과 더미를 출력한다.

   - 2차원 배열 space, 12*(21+3)를 이용

   - 벽과 바닥과 더미는 1로 표시

2. 네모 블록 4*4 하나가 0.8초에 하나씩 아래로 떨어지도록

   - space에서는 2로 표시 (**블록의 등장** 참고)

   - 블록의 아래쪽 첫번째 줄부터 1줄씩 나오도록

   - 다른 블록에도 적용될 수 있게

   - 벽보다 위인 블록은 출력되지 않도록

3. 바닥과 더미를 감지하고 닿았을 때 쌓이도록

   - 한 칸 내려갔을 때 충돌하는 경우(값이 3이상 되는 space의 원소가 있는 경우) 좌표를 다시 한 칸 위로 올리고, 1을 빼 준다. 그리고 블록 좌표를 맨 위, 중앙으로 재설정한다. (새 블록을 출발시킴) (**충돌 판별과 더미화** 참고)

4. ←, → 키를 눌렀을 때 좌, 우로 움직이도록 하되 벽과 충돌했을 때 더 이상 가지 않도록 (3 참고)

5. ↓ 키를 눌렀을 때 충돌 전까지 한 칸 씩 아래로 떨어지도록

6. 줄이 가득 찼을 경우 삭제되고 삭제된 줄 윗줄을 아래로 삭제된 줄만큼 이동

7. 7가지 블록들이 순서대로 떨어지도록 (회전 테스트용)

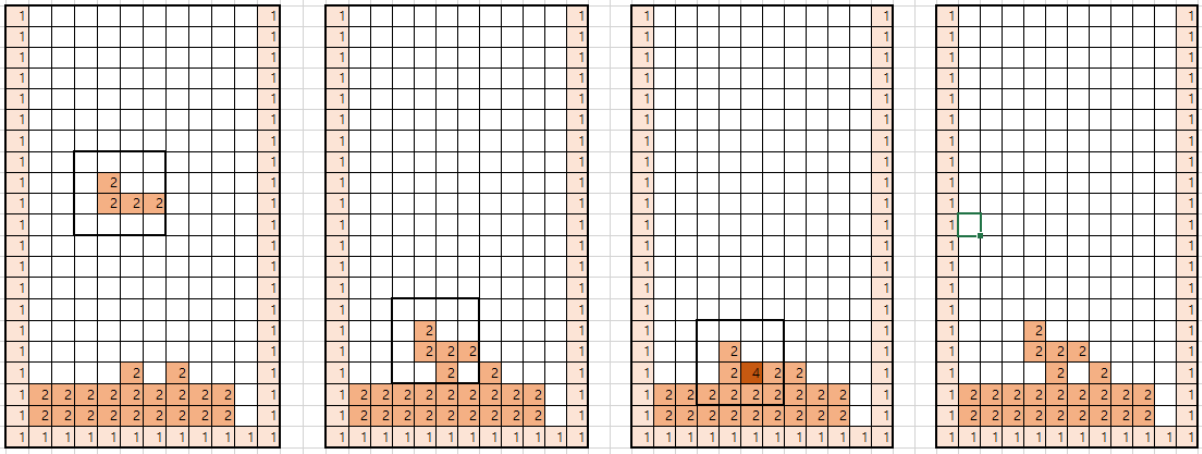
8. SPACE BAR를 눌렀을 때 블록이 회전하도록

   - 회전했을 때 벽과 충돌하는 경우 좌, 우로 블록을 이동시킨다.

9. 7가지 블록들이 랜덤으로 떨어지도록

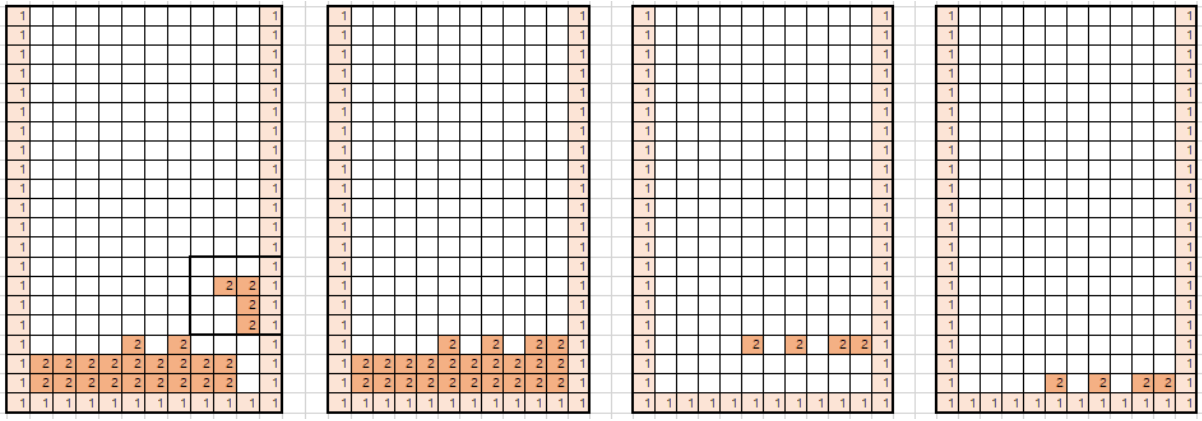
10. 맨 윗줄에 더미가 닿은 경우 종료되도록

충돌 판별과 더미화

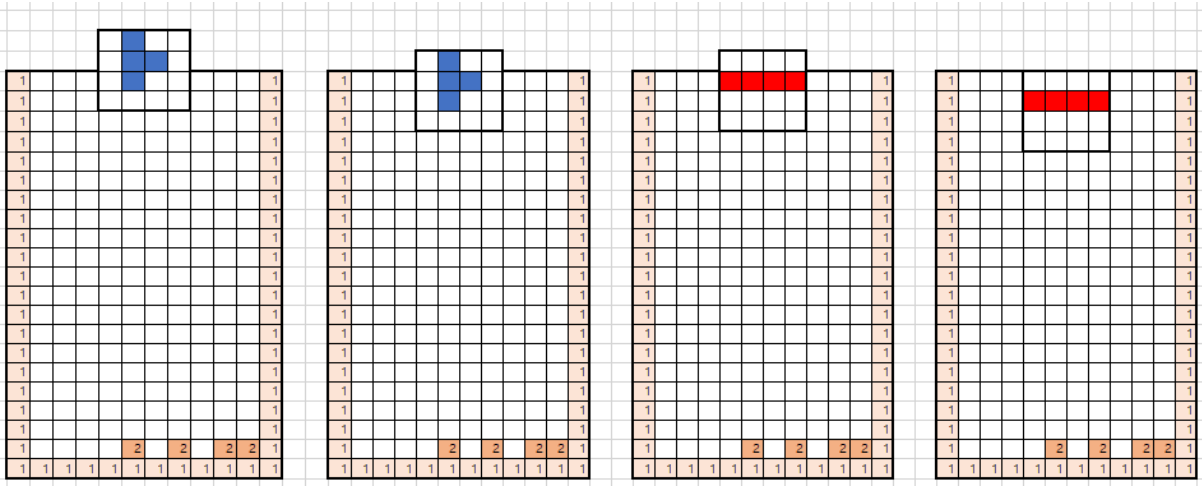


- 4x4블록 배열 안에서만 충돌 검사

줄 삭제



블록의 등장



- space 배열에 위쪽으로 3행 더 필요

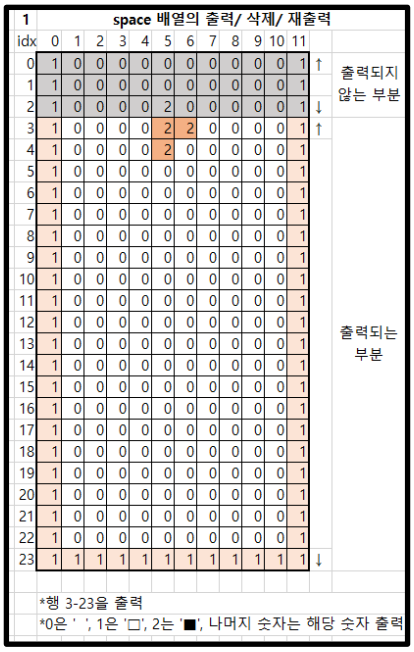
일정표

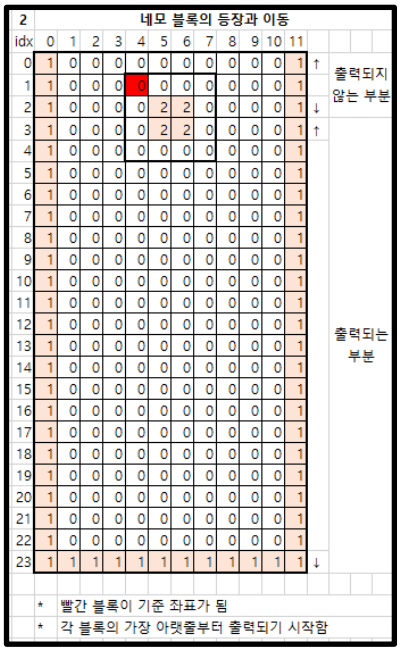
#	추가할 기능	예정 완료 날짜	완료 날짜	커밋 번호
1	space 배열의 출력/ 삭제/ 재출력		3/2	v1 commit
2	네모 블록의 등장과 이동	수(3/3)	3/3	v2 commit
3	충돌판별과 더미화	목(3/4)	3/3	v3 commit
4	←, → 키 기능		3/4	v4 commit
5	↓키 기능		3/4	v4 commit
6	줄 삭제	금(3/5)	3/4	v5 commit
7	블록 순차 등장		3/4	v6 commit
8	SPACE BAR 키 기능		3/4	v6 commit
9	블록 랜덤 등장		3/5	v7 commit
10	더미가 쌓여 게임 종료	토(3/6)	3/5	v7 commit
11	기능별 테스트 체크 리스트 만들기 테스트 및 개선	일-월(3/7-8)	3/10	v8 commit (6-3, 8-1 개선)
12	문서 정리	화-수(3/9-10)	3/11	v9 commit

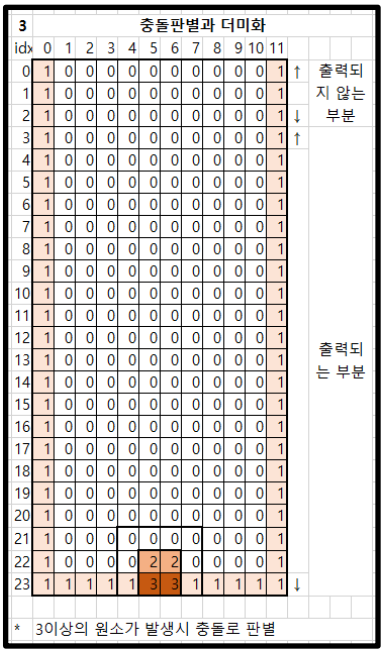
전역변수와 함수

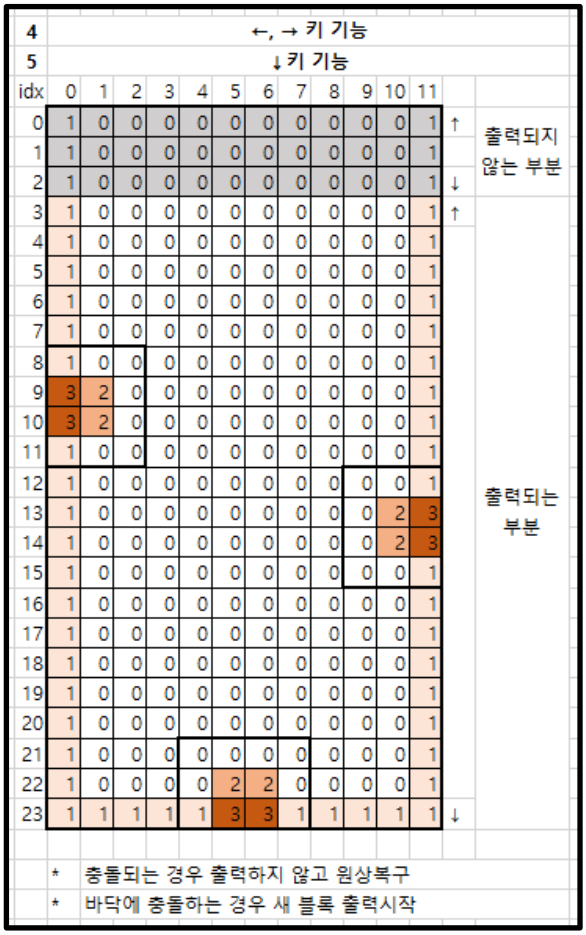
전역변수	int[24][12]	nSpace	화면에 출력되는 배열
전역변수	int[7][4][4][4]	nBlock	7가지블록의 4가지형태를 저장하는 배열
전역변수	int	nX	nSpace상에서의 블록의 가장 상단의 x좌표
전역변수	int	nY	nSpace상에서의 블록의 가장 좌측의 y좌표
전역변수	int	nBlockNo	7가지 블록 중 떨어지고 있는 블록의 번호
전역변수	time_t	tStart	블록의 좌표가 변경된 시각
전역변수	time_t	tEnd	현재시각
전역변수	int	nFalling	블록이 떨어지고 있으면 1, 바닥에 닿으면 0 반환
전역변수	int	nBlockRot	블록의 회전상태
전역변수	int	nGameOver	게임끝나면 1, 진행중이면 0 반환
함수	void	initSpace()	space를 초기화
함수	void	drawAll()	space를 화면에 출력
함수	void	addBlock()	nSpace에 블록의 값을 추가
함수	void	delBlock()	nSpace에서 블록의 값을 삭제
함수	int	isLineFull()	꽉찬 라인이 있는 경우 가장 아래쪽 꽉찬 라인의 인덱스반환, 꽉찬 라인이 없는 경우 0 반환
함수	void	delFullLine()	꽉찬 라인을 지움
함수	int	isCrushing()	충돌한 경우 1, 아닌 경우 0 반환
함수	void	setNewBlock()	어떤 블록 떨어뜨릴지 결정, 시작 nX, nY 좌표 설정
함수	void	moveToLeft()	충돌 피해 왼쪽으로 이동
함수	void	moveToRight()	충돌 피해 오른쪽으로 이동
함수	void	revertRot()	충돌 피해 회전취소
함수	void	avoidCrush()	회전후 충돌 피해 이동
함수	void	dropBlock()	블록의 좌표를 이동한지 1초가 지나면 블록을 한칸 떨어뜨림
함수	void	getInput()	키보드 인풋 받아 기능 수행

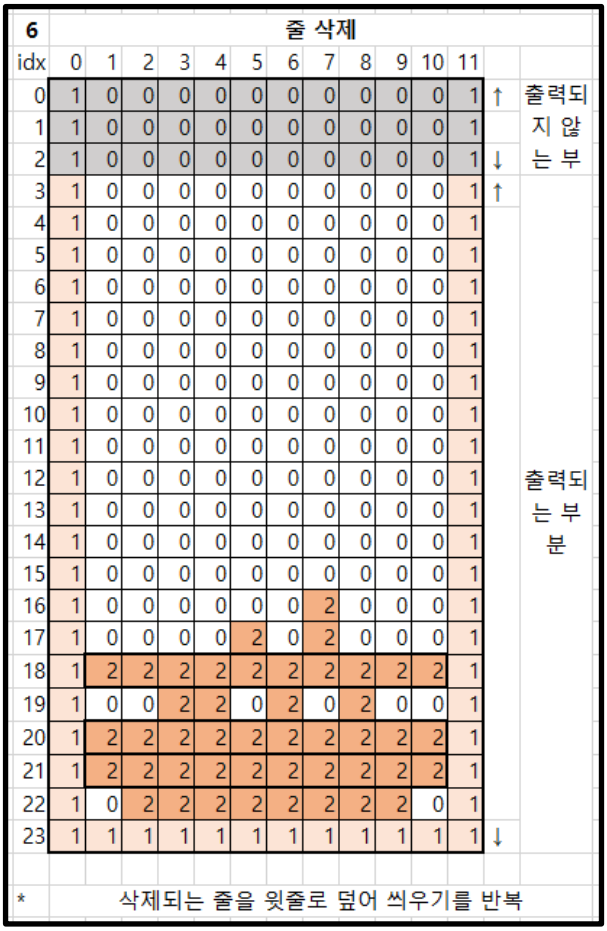
기능 별 구현 부가 설명

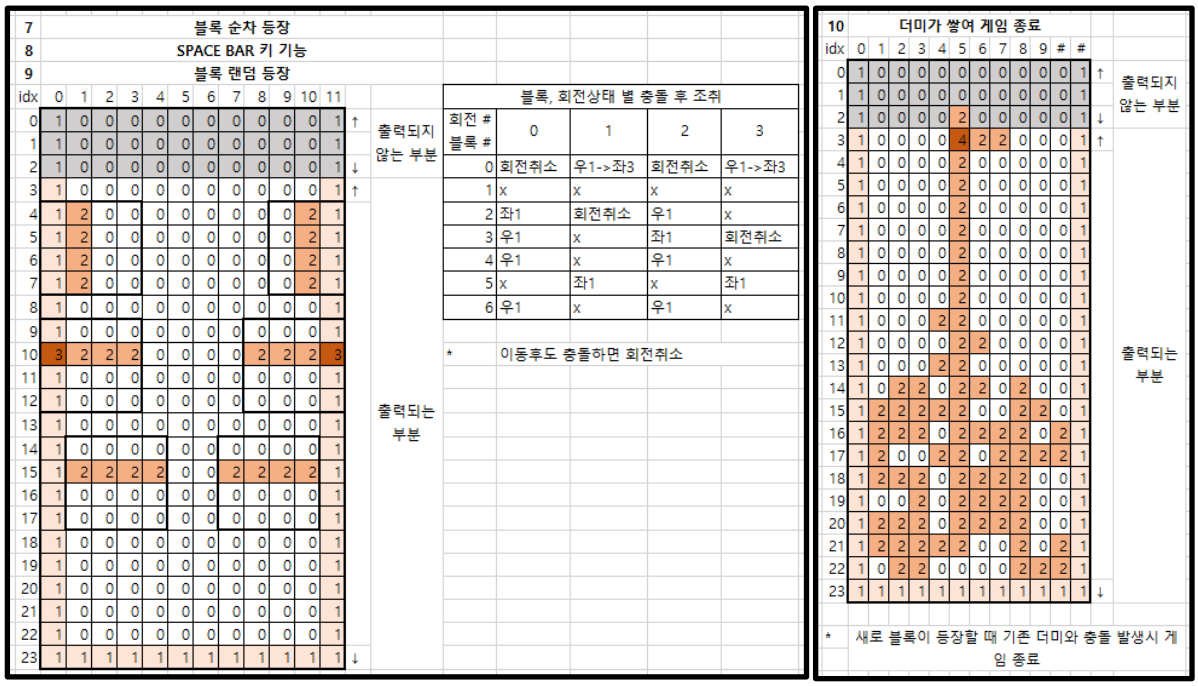

1
space 배열의 출력/ 삭제/ 재출력
idx 0 1 2 3 4 5 6 7 8 9 10 11
출력되지 않는 부분
출력되는 부분
*행 3-23을 출력
*0은 ' ', 1은 '□', 2는 '■', 나머지 숫자는 해당 숫자 출력


2
네모 블록의 등장과 이동
idx 0 1 2 3 4 5 6 7 8 9 10 11
출력되지 않는 부분
출력되는 부분
* 빨간 블록이 기준 좌표가 됨
* 각 블록의 가장 아랫줄부터 출력되기 시작함


3
충돌판별과 더미화
idx 0 1 2 3 4 5 6 7 8 9 10 11
출력되지 않는 부분
출력되는 부분
* 3이상의 원소가 발생시 충돌로 판별


4
←, → 키 기능
5
↓키 기능
idx 0 1 2 3 4 5 6 7 8 9 10 11
출력되지 않는 부분
출력되는 부분
* 충돌되는 경우 출력하지 않고 원상복구
* 바닥에 충돌하는 경우 새 블록 출력시작


6
줄 삭제
idx 0 1 2 3 4 5 6 7 8 9 10 11
출력되지 않는 부분
출력되는 부분
* 삭제되는 줄을 윗줄로 덮어 씌우기를 반복



테스트 체크리스트

테스트 체크 리스트					
#	기능별 체크 필요한 사항			확인 결과	수정 방안
1	space 배열의 출력/ 삭제/ 재출력	1	게임 실행 중 출력상태가 전반적으로 양호한지 확인	1) ↓키를 길게 누르고 있을 경우 출력 화면이 깜빡임	1) ↑키에 즉각 드롭 기능을 추가하여 '사용자'가 ↓키를 길게 누르는 일이 없도록 유도. 2) ↓를 길게 누른 값이 지금 떨어지고 있는 블록에만 적용되도록 설정 3) 연속 입력의 간격에 한계가 있도록 설정
2	블록의 등장과 이동	2-1	블록이 떨어진 직 후 위에서 바로 다음 블록의 첫 줄이 출력되는지	정상 작동	-
		2-2	↓키를 누르지 않는 상태에서 일정한 시간 간격으로 블록이 떨어지는지	정상 작동	-
3	←, → 키 기능	3-1	←, →키를 눌렀을 때 각각 좌, 우로 한 칸씩 이동하는지	정상 작동	-
		3-2	벽 또는 더미와 충돌하였을 때 이동이 막히는지	정상 작동	-
4	↓키 기능	4-1	↓키를 눌렀을 때 한 칸씩 떨어지는 지	1) 1초 간격인 자동 드랍과 타이밍이 겹칠 경우 두칸씩 내려오는 경우가 있음	1) ↓키를 입력 할 때 tStart가 초기화되도록 설정
		4-2	바닥 또는 더미와 충돌하였을 때 새 블록이 바로 출력되기 시작하는지	정상 작동	-
5	줄 삭제	5	한 줄 또는 그 이상이 가득 찼을 때 적절한 시점에서 삭제 되는 지	정상 작동	-
6	SPACE BAR 키 기능 (sheet 11-1 참고)	6-1	SPACE BAR 키를 눌렀을 때 올바른 도형의 모양이 보여지는지	정상 작동	-
		6-2	벽과 가까이에서 회전 하였을 때 충돌하지 않는지	정상 작동	-
		6-3	벽과 더미사이 좁은 공간에서 회전하였을 때 충돌하지 않는지	1) 벽과 더미 사이로 들어가는 입구에서 회전하는 경우 2) 고르지 않은 벽과 더미 사이에서 회전하는 경우 충돌 발생	1) 회전과 충돌판별법을 블록별, 환경별로 경우의 수를 나눠 수정 2) 충돌 판별 후 충돌시 회전 취소 부분을 한단계 더 추가
7	블록 랜덤 등장	7	블록이 랜덤하게 등장하는지	정상 작동	-
8	더미가 쌓여 게임 종료	8-1	새로운 블록이 더미와 바로 충돌할때 게임이 정상 종료 되는지	정상 작동	-
		8-2	더미가 한쪽으로 쌓여 새로운 블록과 충돌하지 않는 경우 정상 플레이 되는지	1) 오른쪽으로 쌓은 경우 게임이 멈춤 (종료x)	1) 멈출만한 곳을 찾아 수정

- 중요도 순으로 빨, 주, 노, 초

개선 과정 및 결과

#	수정 방안	수정 과정 및 결과
6-3	1) 회전과 충돌판별법을 블록별, 환경별로 경우의 수를 나눠 수정 2) 충돌 판별 후 충돌시 회전 취소 부분을 한단계 더 추가	2)의 방법으로 수정 후 해결
8-2	1) 멈출만한 곳을 찾아 수정	함수 delFullLine()이 0을 return 하지 않아 0을 return 하도록 수정
1	1) ↑키에 즉각 드롭 기능을 추가하여 '사용자'가 ↓키를 길게 누르는 일이 없도록 유도. 2) ↓를 길게 누른 값이 지금 떨어지고 있는 블록에만 적용되도록 설정 3) 연속 입력의 간격에 한계가 있도록 설정	보류
4-1	1) ↓키를 입력 할 때 tStart가 초기화 되도록 설정	보류