

B777030 정승훈

게임소프트웨어

1. 3차원 Cube오브젝트를 사용해서 테트리스를 만들었다.
2. Cube오브젝트를 연결하여 만든 게임보드와 다음에 나올 테트리스 블록, 스코어와 연속 콤보 횟수를 UI의 Text를 통해 구현했다.
3. Cube오브젝트를 연결하여 테트리스 블록 하나를 만들고 각각에 다른 색을 입혔다
4. WASD와 화살표키, Space bar를 사용한다. W와 위방향키는 블록을 90도 회전시키고 A, D, S 와 좌우, 아래 방향키는 블록을 좌우와 아래로 이동시킨다.
5. GameManager가 블록의 위치를 기억하는 방식으로 구현했기 때문에 충돌처리 기능은 사용하지 않았다(Trigger, Collision)
6. 신규 블록 생성시 Instantiate를 통해 생성하고 생성한 블록의 부모와 위치 등을 설정했다.
7. 추가기능은 구현하지 않았다.

충돌처리 방식 대신에 GameManager가 가지고 있는 bool배열과 GameObject배열을 사용했다.

게임보드 외곽을 포함한 배열로 true일 경우 그곳에 블록이 있다는 의미로 사용했다. 그래서 이동할 때 오브젝트가 다음에 갈 위치의 bool값을 GameManager를 통해서 받아와서 모두 없다고 판정이 나와 이동하게 만들었다. 좌우 이동 같은 경우에는 이동하지 못해도 블록을 멈추는 기능을 넣지 않았지만 아래로 이동하는 것은 더 이상 움직일 수 없을 때를 위해 stop함수를 사용한다.

확인만 하기 위해서는 bool 배열만 있으면 되지만 한 줄을 모두 채웠을 때 블록을 제거하기 위해서 GameObject배열을 추가했다.

BlockSpawner는 테트리스 블록을 생성해주는 스크립트로 7종류의 블록을 랜덤으로 생성한다. 자식객체(컨트롤 가능한 객체)가 없어지면 미리 생성해 놓은 블록을 가져오고 새로운 블록을 생성한다.

BlockController는 블록 이동, 블록 정지에 사용된다. GameManager는 독자적으로 돌아가는 것이 없이 변수를 가지고 그것을 사용하는 함수를 외부로 노출시킨 형태이기 때문에 BlockController에서 GameManager를 얻어와서 사용하고있다.

회전은 만약 회전했을 때 게임공간에서 벗어나면 회전을 무효화한다.