

OpenGL을 이용한 공간지각능력 게임 구현

21300112 김성실, 21500243 문주혜

2018-2 인터랙티브 비주얼 미디어

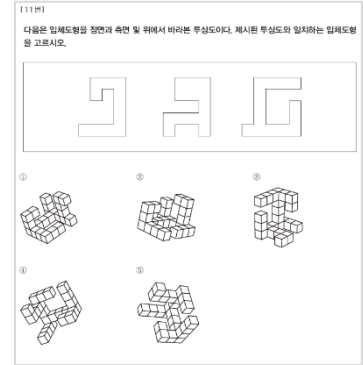
신성용 교수님, 한다성 교수님

[목차]

1. 주제_공간지각능력
2. 아이디어 게임화
3. 실현 기능
 - 3-1. 초기 화면 구성
 - 3-2. View Customization
 - 3-3. 선택하기
 - 3-4. Cube Regeneration
 - 3-5. 완성 확인
 - 3-6. 완성 Effect
4. 프로그램 사용 방안
5. role of team members
6. References

1. 주제_공간지각능력

컴퓨터 그래픽을 통하여 반응하는 감각 기관은 바로 시각이다. 사람이 얻는 대부분의 정보는 눈을 통해서 들어온다. 그 시각적 데이터를 이용하여 머릿속으로 문제 해결을 위한 정보를 구성하고 조합한다. 이번 학기 OpenGL을 활용하여 시각적 데이터를 어떻게 하면 잘 전달할지를 공부하였다. 그렇다면 그 시각적 데이터를 어떻게 잘 활용할 수 있는지에 대한 부분도 고민하는 시간을 가져보았다. 그 때 나온 키워드가 바로 공간지각능력이다. 공간지각능력은 상하, 좌우, 전후의 공간 관계나 위치를 감각을 통하여 파악하는 시각 능력이다. 그리고 경험했던 공간지각능력 향상 활동 중 가장 먼저 떠올랐던 것은 투상도 문제이다. 입체도형의 정면과 측면 및 위에서 바라본 투상도를 보고 입체도형을 추측하는 문제이다. 우리는 이 문제를 2차원이 아닌 3차원으로 구현하여 3차원에서 생각하는 입체도형을 좀 더 직관적으로 봄으로 더 효과적인 학습이 되도록 주제를 정해 보았다.



2. 아이디어 게임화

우리는 3D를 통하여 입체 도형을 다각도로 보는 것을 실현함으로 더 이해하기 쉽고 어렵지 않게 연습할 수 있는 공간지각능력 게임을 구현하고자 한다. 본 아이디어를 구체적으로 게임화 하기 위하여 생각한 단계는 크게 세가지이다. 첫 번째는 초기 화면 구축으로, 정면과 측면, 위에서 바라본 투상도와 큐브 1000(10*10*10)개를 제공한다. 세 시야에서 바라본 투상도는 사용자에게 공간지각능력을 활용하여 입체도형을 유추하기 위한 정보로 제공된다. 큐브는 가로, 세로, 높이 각각 10개씩 총 1000개의 큐브를 미리 생성한다. 두 번째는 게임 진행 단계이다. 사용자가 제공된 투상도를 고려하여, 필요 없는 큐브를 클릭하여 삭제함으로 입체도형을 완성시킨다. 마지막 단계는 완성단계이다. 입체도형이 알맞게 완성되면 사용자에게 완성했다는 알림으로 이벤트를 주어 성취감을 갖게 한다.

3. 실현 기능

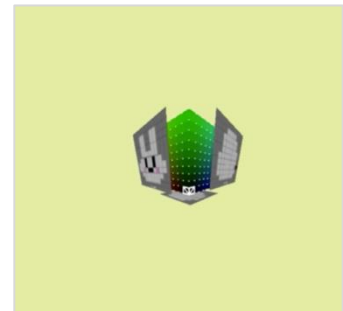
본 게임은 Visual Studio와 OpenGL 3.0을 활용한 게임으로 게임에 사용된 파일은 obj파일과 mtl, jpg, png 파일을 이용하여 구현되었다. 더욱 세세한 기능은 아래와 같다.

3-1. 초기 화면 구성

게임 시작과 함께 크게 두 가지의 오브젝트가 화면에 나타난다. 첫 번째는 사용자가 공간지각 능력을 펼쳐 추측해야 할 입체도형을 만들 큐브 1000개와 정면과 측면, 위에서 바라본 투명도를 보여주는 세 개의 면이다.

① 초기 큐브 세팅

가로 세로 높이 방향으로 각각 10개씩 총 1000개의 큐브를 생성한다. 이때 각각의 큐브의 상태값(출력상태-is_visible, 정답 큐브 유무-is_true, 색-color)은 struct Cube[10][10][10]에서 관리된다. 이때 Cube[i][j][k]에서 (i,j,k)는 Cube[i][j][k]의 위치값(x,y,z)이다. 큐브의 위치 값(x, y, z)값에 비례해서 각 큐브의 색(r, g, b)값을 부여함으로서 Picking 과정에서 받아 오는 색(r, g, b)값을 통해 선택하는 cube의 위치 값(x, y, z) 값을 알 수 있다. 또한 큐브의 별텍스를 표시해줌으로 사용자가 큐브 간의 경계를 확실히 보여줄 수 있도록 한다.



② 세 방향에서 본 시야 plane 세팅 (Textured object(s))

정면과 측면, 위에서 바라본 투명도를 texturing한 plane 세 개를 불러와 각 위치에 맞게 위치시킨다. 또한 원활한 게임 진행을 위하여 'a'를 눌렀을 때 plane이 사라졌다 나타나도록 설정한다.

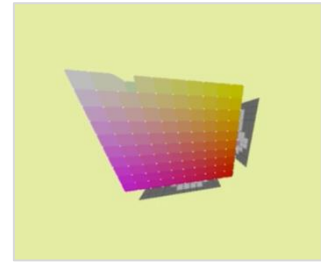
3-2. View Customization

사용자가 원하는 화면 뷰를 볼 수 있도록 화면 전환과 확대&축소, 이동할 수 있는 코드를 제공한다.

- ① 마우스 휠: 화면이 확대/축소 된다.
- ② 마우스 왼쪽 버튼: 카메라 뷰의 각도가 rotation된다.
- ③ Ctrl+마우스 왼쪽 버튼: 카메라 뷰의 위치가 마우스 왼쪽 버튼에 맞춰 상하좌우로 움직인다.

3-3. 선택하기 (*Interactively picking (or selecting) object(s) in 3D world*)

현재 클릭한 파트에 색을 확인하여 그에 해당하는 위치 값을 가진 블록을 삭제한다. 큐브의 색의 경우 초기 생성할 때 좌표 값의 따라 고유의 색 값을 가지고 있다. 그리고 그 값이 좌표 값과 이어져 있기 때문에 하나의 아이디로 사용한다. 또한 삭제 할 때 해당 cube의 위치값 $\text{vec3}(i,j,k)$ 을 $\text{vector idx_deleted}$ 에 pushback한다.



3-4. Cube Regeneration

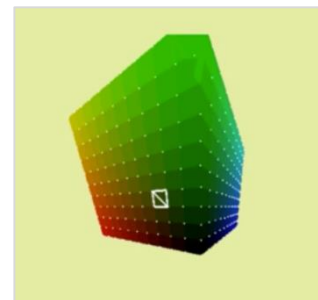
사용자가 큐브를 다시 생성하고 싶을 경우 두 가지 방법을 큐브를 생성할 수 있다. 큐브가 사라지거나 생성된 값을 텍스트로 출력해 줌으로 사용자가 직관적으로 진행 사항을 확인할 수 있다.

① 단축키 'z' 를 이용한 빠른 취소 (*Moving and/or rotating object(s)*)

'z'키를 누를 경우 저장된 idx_deleted 의 가장 마지막 값을 불러와 그 값의 해당하는 큐브를 생성한다. 그리고 그 값을 삭제함으로 그 다음 값이 최근 값이 되도록 설정한다. 이를 통하여 실행 취소 기능을 구현했다.

② 게임 패드와 키보드를 이용한 지정 생성 (*Interaction with a player by a gamepad*)

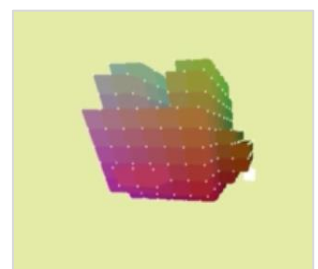
게임 패드 또는 키보드를 통하여 커서를 moving하여 원하는 위치에 지정하여 그 위치에 해당하는 큐브를 생성한다. 큐브의 위치값은 $\text{glm::vec3 detective_position}$ 에 저장되고 변경된다. '상', '하', '좌', '우', '앞', '뒤'를 키보드에서는 'I', 'K', 'J', 'L', 'Y', 'H'로 게임 패드에서는 'UP', 'DOWN', 'RIGHT', 'LEFT', 'HOME', 'END'버튼으로 조작한다. 그리고 원하는 위치에 커서를 옮긴 뒤 '+'와 '-'키를 이용하여 생성하거나 삭제한다. 이때도 또한 삭제 할 때 해당 cube의 위치값 $\text{vec3}(i,j,k)$ 을 $\text{vector idx_deleted}$ 에 pushback한다.



3-5. 완성 확인

사용자가 큐브를 정확한 모양으로 완성하였는지 확인하기 위하여 큐브를 누를 때 마다 각각의 완성된 결과값의 큐브 값(is_true)과 현재 존재하는 큐브 값(is_visible)이 일치하는지 확인하는 함수($\text{check}()$)를 넣는다. 이때 모든 값이 일치 하면 true를 아니면 false를 return한다. 완성된 입체 도형의 값은 아래와 같다.

$[x][y][z] = \{[0][\text{all}][\text{all}], [1][0,5,6,7,8,9][\text{all}], [2,7][8,9][\text{all}], [3,4,5,6][9][\text{all}], [8][0,6,7,8,9][\text{all}], [9][0,1,3,4,5,6,7,8,9][\text{all}], [\text{all}][0,1,4,5,6,7,8,9][0,9],$



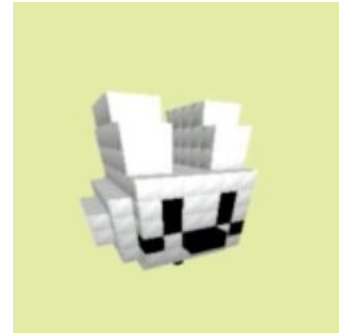
[all][0,5,6,7,8,9][1,8], [all][9][2,3,6,7], [all][6,7,8,9][4,5], [0,1,2,7,8,9][all][0,9], [0,1,7,8,9][all][1,8],
[0,9][all][2,3,6,7], [0][all][4,5] }

3-6. 완성 Effect

사용자가 알맞게 완성할 경우 완성을 축하하는 의미로 총 4가지의 효과가 나타난다.

① 입체도형 Texturing

입체도형을 만들기 위한 큐브들을 삭제하고, 텍스처링이 입혀진 또 다른 입체도형 파일을 불러옴으로 더욱 현실 같은 입체도형이 나타난다.



② 입체도형 Rotating

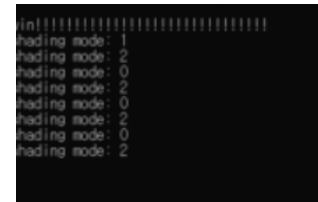
입체도형이 제자리에서 돌면서 조금씩 위로 올라간다. 이를 통해 완성된 입체 도형에 움직이는 효과를 준다. 이때 아래 보이는 검은색 cube는 (0,0,0)에 위치하고 있다.

③ Win 문구

완성했다는 의미로 Win 문구가 출력된다.

④ Shading Mode (*Switch between three shading modes: Phong, Gouraud, and Constant*)

1번 2번 3번을 누르면 완성된 입체 도형의 shading 모드가 바뀐다. 1번의 경우 phong 모드로 2번은 gouraud, 3번은 flat shading mode를 제공한다.



4. 프로그램 사용 방안

공간지각 능력을 키우기 원하는 모든 분야에서 사용이 가능하다. 또한 본 프로그램은 기존의 문제와 같이 2D가 아닌 3D로 볼 수 있어 더 쉽게 머릿속에서 구상한 결과를 직관적으로 적용시키고 활용할 수 있다는 장점이 있다. 이 장점을 이용하여 공간지각능력을 처음 공부하기 시작하는 초등 단계나 아직 공간지각능력이 발달이 필요한 학습 단계에서 응용하여 사용할 수 있다. 이후 원하는 입체 모형을 자동으로 계산하여 문제를 생성하는 프로그램으로 발전시켜 경제적으로 유리하게 사용한다.

5. role of your team members_팀 역할

감성실, 문주혜: 아이디어 게임 구체화, 주제 선정, 게임 기획, OBJ파일 제작, 초기 게임 단계 구축, TEXTURING, PICKING, 큐브 생성, z단축기, 게임패드 설정, 입체모델 완성시 효과 구축, 파일 합치기, 문제 완성 확인 과정, 기타 버그 수정, 보고서, 데모 동영상 촬영, 데모 동영상 편집

6. References

<https://blog.naver.com/kids1545/221115489469>