

심화전공실습 (CGL)

HW12_Picking



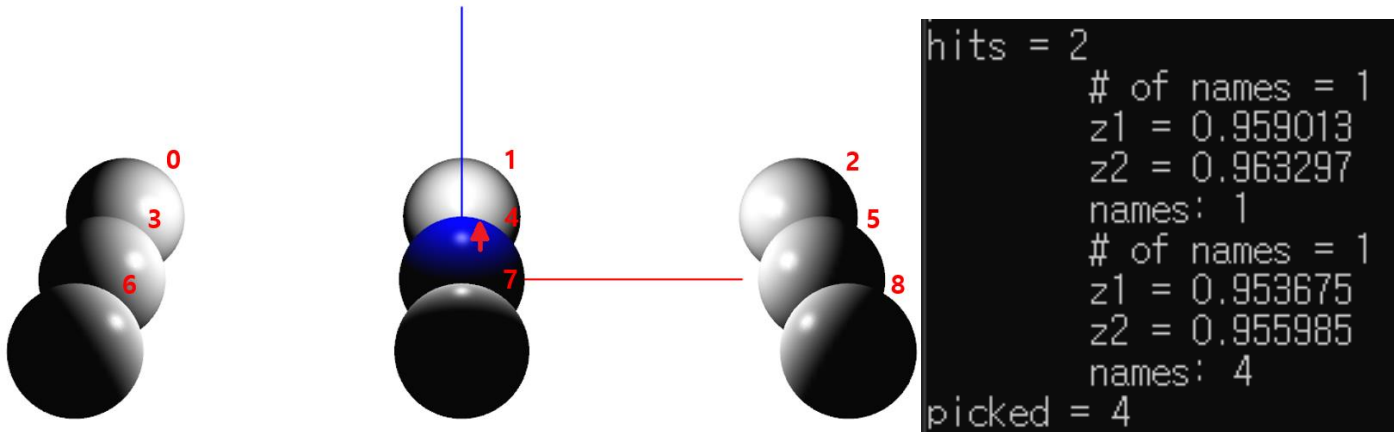
Self-scoring table

	P1	E1	E2	Total
Score	1	1	1	3

2018707068 김경환

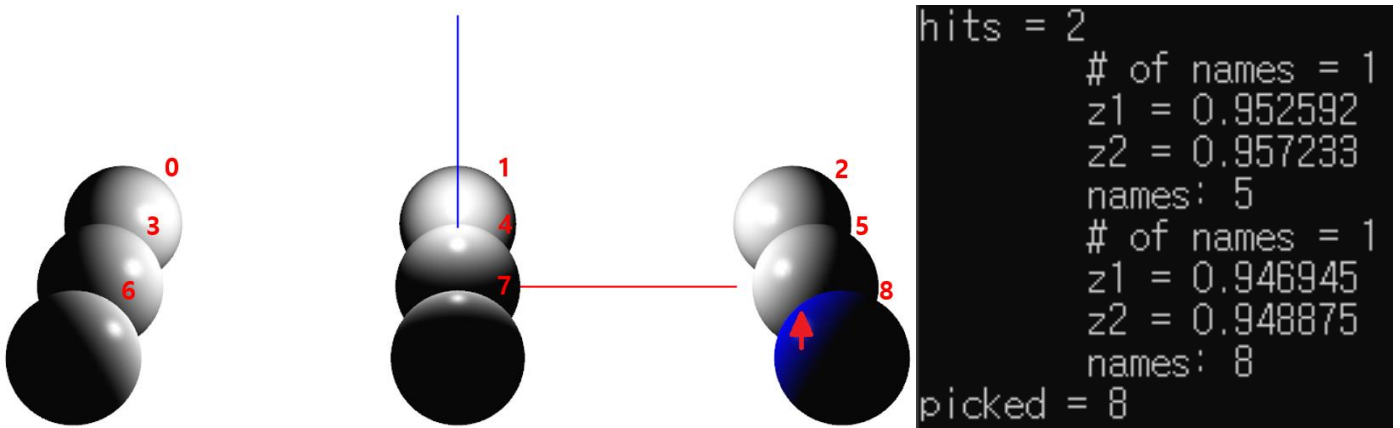
KwangWoon University

Practice 01. Picking an object using selection mode:



위 왼쪽 사진은 1번 구와 4번 구를 함께 selection되도록 4번 구의 경계를 클릭한 모습이다.

오른쪽 사진을 보면 hits된 개수가 2개이고 names가 1과 4로 올바르게 출력되는 것을 볼 수 있다.



위 왼쪽 사진은 5번 구와 8번 구를 함께 selection되도록 8번 구의 경계를 클릭한 모습이다.

오른쪽 사진을 보면 hits된 개수가 2개이고 names가 5과 8로 올바르게 출력되는 것을 볼 수 있다.

1. selection mode를 사용해 picking을 수행할 때는 사용자가 지정한 pick volume에 교차하는 객체의 이름을 저장하기 위한 select buffer를 설정해줘야 한다. (`glSelectBuffer(GLsizei size, GLuint* buffer);`)
2. 기존 projection matrix를 backup하고 단위행렬로 초기해줄 필요가 있다. 이는 pick volum을 올바르게 설정하고 다시 이전 projection matrix를 복원하기 위함이다.
(`glMatrixMode(GL_PROJECTION); glPushMatrix(); glLoadIdentity();`)
3. 사용자가 지정한 교차영역(pick volum)을 지정한다. (`gluPickMatrix(x, y, delX, delY, viewport);`)
4. 렌더링을 위해 초기화한 projection matrix를 다시 설정할 필요가 있다. (`setupProjectionMatrix();`)
5. GL_RENDER에서 벗어나 GL_SELECT로 mode를 바꾸고 객체를 픽셀로 래스터화하는 대신 객체의 이름을 저장한다. (`glRenderMode(GL_SELECT);`)
6. 선택한 객체를 렌더링한다. (`render(window);`)
7. 다시 GL_RENDER로 돌아와 반환 값으로 선택된 객체를 얻어온다. (`GLuint hits = glRenderMode(GL_RENDER);`)
8. 선택된 객체 개수와 이름들로 가장 가까운 객체 이름을 얻는다. (`int name = findNearestHits(hits, selectBuffer);`)
9. 기존 backup해둔 projection matrix를 복원한다. (`glMatrixMode(GL_PROJECTION); glPopMatrix();`)

Select buffer									
n_1	m_1	M_1	i_1	...	i_{n_1}	n_2	m_2	M_2	...
# of names	Min depth value	Max depth value	1st name	...	n_1 -th name				

Draw 3 objects with names

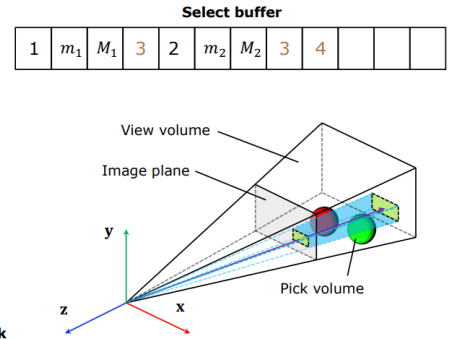
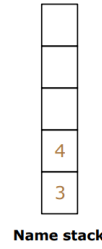
```

■ glInitNames();
■ glPushName(3);
■ drawSphere(red);

■ glPushName(4);
■ drawSphere(green);

■ glPopName();

```

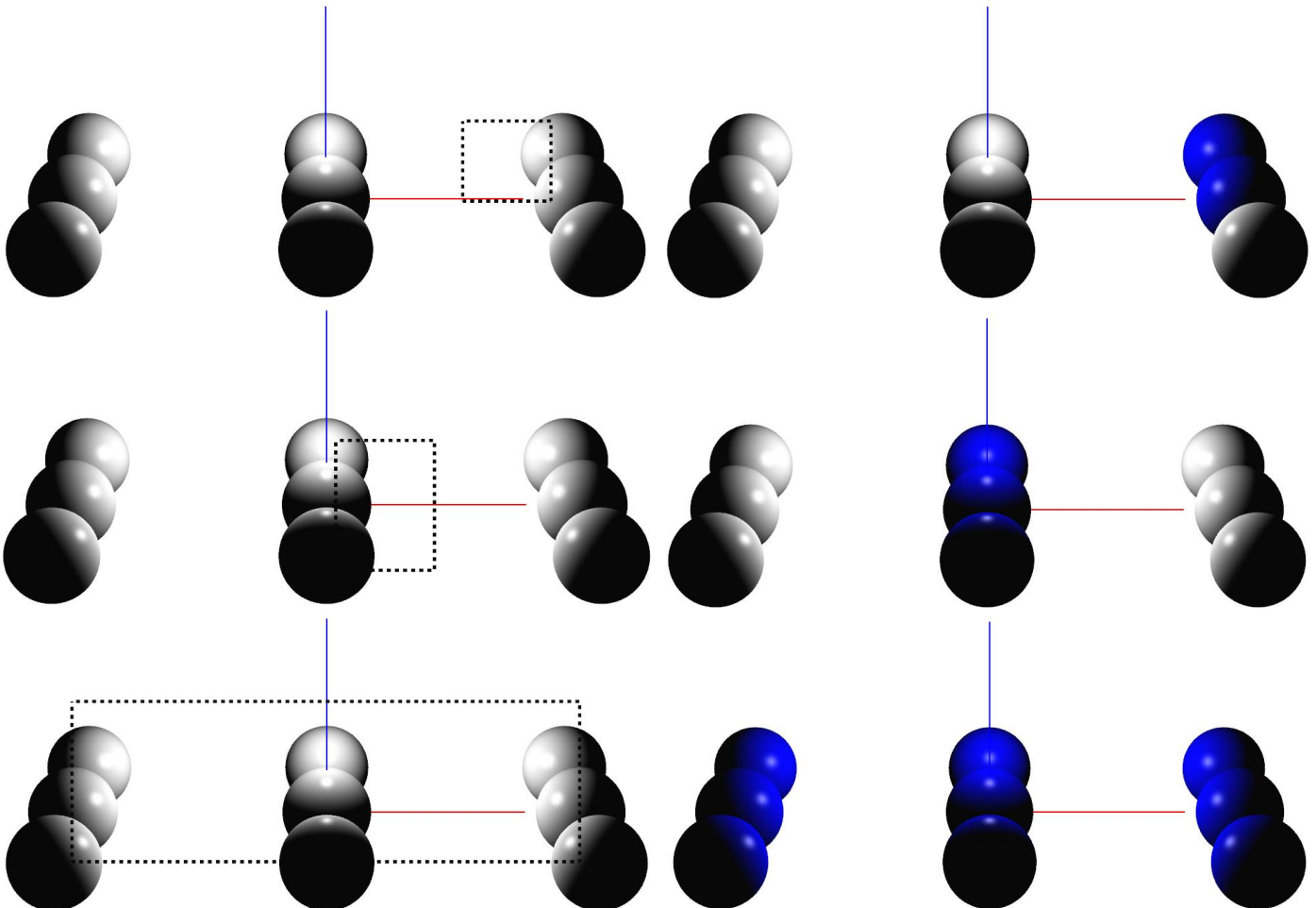


select buffer의 경우에는 위처럼 names의 개수, Min depth value, Max depth value, 1~n name으로 이루어져 있다. 따라서 findNerarestHits() 함수는 가장 가까운 객체를 찾기 위해 selectBuffer 배열의 index를 3 + n씩 증가시키는 것을 hits된 개수만큼 반복하며 배열 내의 원소들을 탐색한다.

이렇게 얻어낸 가장 가까운 객체의 이름을 통해서 렌더링 도중 같은 같은 객체의 이름을 가진 객체를 다른 색으로 표현하여 picking을 구현하였다.

Exercise 01. Implement click-and-drag using stippling:

Exercise 02. Select all objects inside a click-and-drag rectangle:



위 사진은 click and drag로 stippling rectangle을 렌더링하고, 내부의 모든 객체를 multi selection을 한 모습이다. click and drag로 stippling rectangle을 렌더링하기 위해서는 rectangle의 좌측상단과 우측하단의 좌표가 필요하다. 그리하여 mouseButton()에서 LEFT Button을 Press했을 때 dragging 모드로 바꾸고 glfwGetCusorPos를 사용하여 두 좌표를 모두 Cusor의 좌표로 저장한다. 그리고 dragging한 상태로 mouse를 move할 때는 rectangle의 모양이 바뀌어야 함으로 mouseMove()에서 dragging 모드일 때 우측하단의 좌표를 현재 Cusor의 좌표로 갱신한다. 그리고 mouse의 Left Button을 Release하여 rectangle의 크기를 확정지을 때는 complete mode로 바꾸고 현재 Cusor

의 좌표를 rectangle의 우측하단의 좌표로 저장한다.

이렇게 rectangle의 좌표를 얻었으면 이를 렌더링하기 위해서 좌표계를 조정할 필요가 있다.

mouse의 좌표는 window coordinate이기 때문에 9개의 구체가 perspective projection되어 있는 환경에 올바르게 렌더링할 수 없기 때문이다.

그리하여 render()에서 rectangle을 그리는 함수를 호출한다. 이때 rectangle은 mouse를 dragging할 때만 렌더링 되어야 함으로 if문을 사용한다. (if (inputMode == InputMode::DRAGGING) drawRectangle();)

drawRectangle()의 경우에는 위에서 말한 좌표계의 조정이 이루어지기 전에 screen 좌표를 world 좌표로 바꾸는 screen2world()를 사용한다.

그렇게 얻은 world coordinate를 렌더링하기 위해 우선 기존 projection matrix를 backup하고 단위행렬로 초기화한다. 이는 rectangle을 렌더링하고 다시 이전 projection matrix로 돌려놓기 위함이다.

그리고 modelview matrix의 경우에도 현재 2D world coordinate를 렌더링할 때는 영향을 주지 않아야 함으로 단위행렬로 초기화한다.

또한, perspectiveView에 false를 저장하고 setupProjectionMatrix()를 호출하여 orthogonal projection으로 바꾼다.

이제 이전에 얻은 world coordinate를 렌더링하고 다시 backup 해둔 projection matrix를 돌려놓는다.

마지막으로 기존에는 perspective projection이었기 때문에 perspectiveView를 true로 바꿔놓는다.

이제 rectangle 내부의 모든 객체를 multi selection하기 위해서는 practice 1에서 진행한 pick volume을 설정하는 gluPickMatrix(x, y, delX, delY, viewport)를 호출할 때 픽셀의 범위인 delX, delY를 rectangle의 너비와 높이로 설정하고 x, y의 좌표를 rectangle의 중점으로 줘야 한다. 또한, x, y 좌표의 경우 framebuffer coordinate에서의 좌표가 필요함으로 HiDPI를 위해 mouse cursor 좌표에 dpiScaling을 곱해준다.

이렇게 pick volume 설정은 끝났기 때문에 rectangle 내부의 모든 객체는 selection되게 되는데, 이를 시각화하기 위해서는 selection된 객체들의 이름이 필요하다.

그리하여 int 타입의 배열을 hits만큼 동적할당으로 선언하고 기존의 findNearestHits()의 selectBuffer에서 객체들의 이름에 접근할 때 해당 배열에 저장하도록 한다.

마지막으로 구체를 렌더링하는 2중 for문의 가장 안쪽에서 해당 배열을 전부 탐색하는 for문을 사용하여 저장된 이름과 같은 이름의 구체는 다른 색깔로 렌더링되도록 한다.