

CT4201/EC4215: Computer Graphics

# OpenGL: Interaction

---

BOCHANG MOON

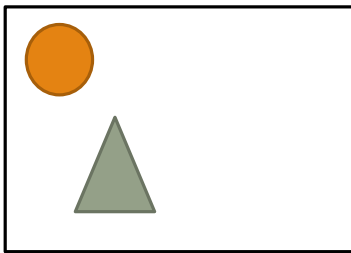
# User Interaction in Virtual World

---

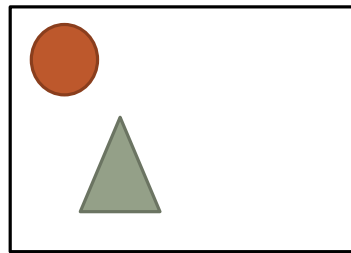
- Camera transformation
  - A user changes his/her eye position, direction, ...
- Object transformation
  - A user can move an object

# Target Scenario

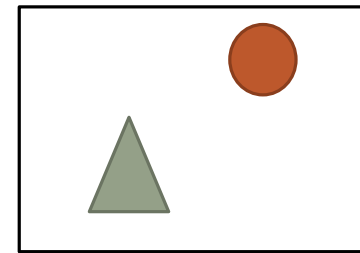
- Problem: a user may want to select an object in 3D world, and transform the chosen object by keyboard or mouse inputs
  - Click a mouse button at a pixel and drag a selected object
- Tasks
  - Catch the mouse event
  - Identify which objects are selected
  - Apply a transformation to the objects by the user input



input



User selects an  
object



Transform the  
object

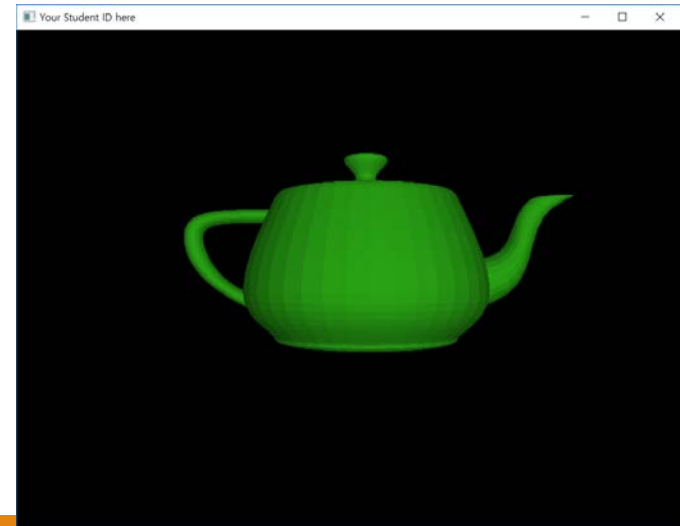
# Picking and Selection

---

- Ways to select an object
  - Build a ray (position and direction) and identify the intersection point between the ray and primitives
  - Employ OpenGL selection buffers
  - Utilize OpenGL a double buffer

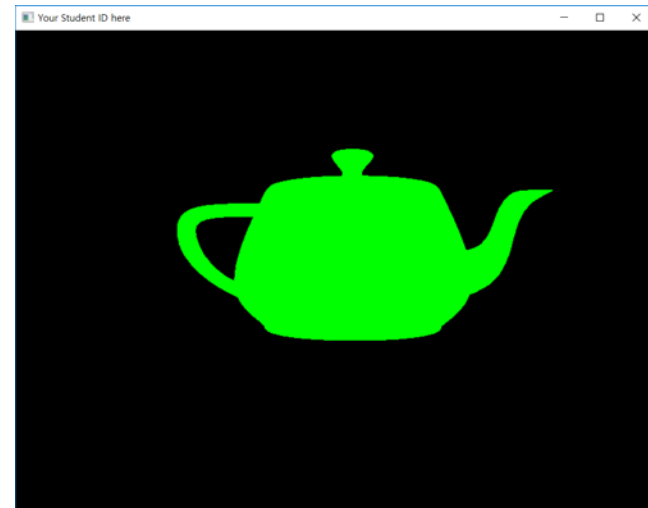
# A Naïve Idea 1

- Read the pixel color at the clicked position (x,y) from your screen (frame buffer)
  - unsigned char pixel[3];
  - glReadPixels(x, y, 1, 1, GL\_RGB, GL\_UNSIGNED\_BYTE, pixel)
- Compare the color with object colors
  - e.g., if the color is green, then the selected object is the teapot!
- Problems?
  - Each primitive can have different colors because of
    - Lighting, textures...



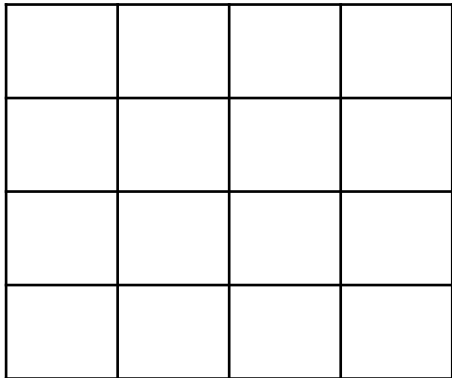
# A Naïve Idea 2

- Compare the color with object colors
  - e.g., if the color is green, then the selected object is the teapot!
- Disable the lighting (and texture)
  - `glDisable(GL_LIGHTING);`
- Assign a specific color (like object ID) to each object
  - `glColor4f(0.0, 1.0, 0.0, 1.0);`
  - `//` before drawing your primitives
- Problems?
  - This is not a realistic image...

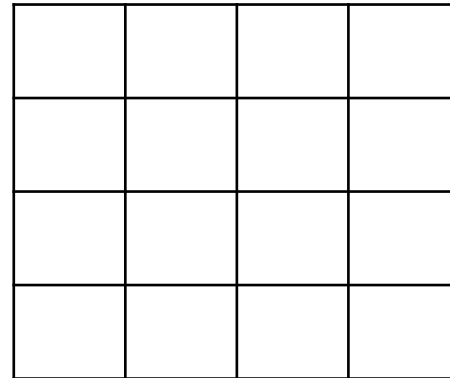


# OpenGL: Double Buffer

- Two buffers can be allocated with the following:
  - You can set a mode before creating an window.
  - `glfwWindowHint( GLFW_DOUBLEBUFFER, GL_FALSE )` // Single buffer mode
  - `glfwWindowHint( GLFW_DOUBLEBUFFER, GL_TRUE )` // Default option



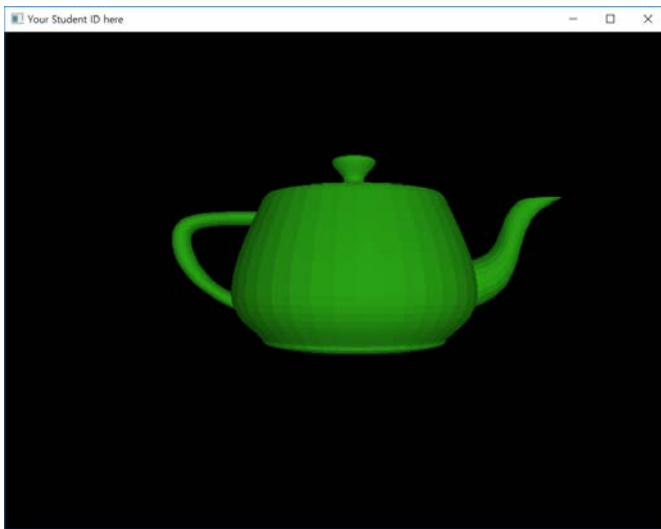
Front buffer



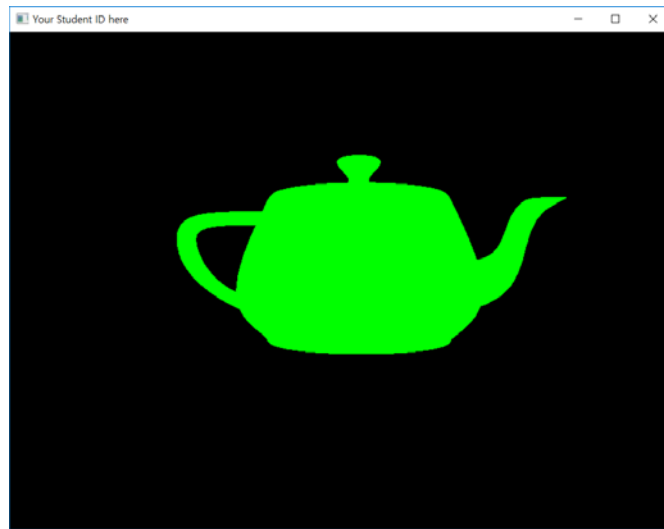
Back buffer

# OpenGL: Double Buffer

- Two buffers can be allocated with the following:
  - You can set a mode before creating an window.
  - `glfwWindowHint( GLFW_DOUBLEBUFFER, GL_FALSE )` // Single buffer mode
  - `glfwWindowHint( GLFW_DOUBLEBUFFER, GL_TRUE )` // Default option



Front buffer



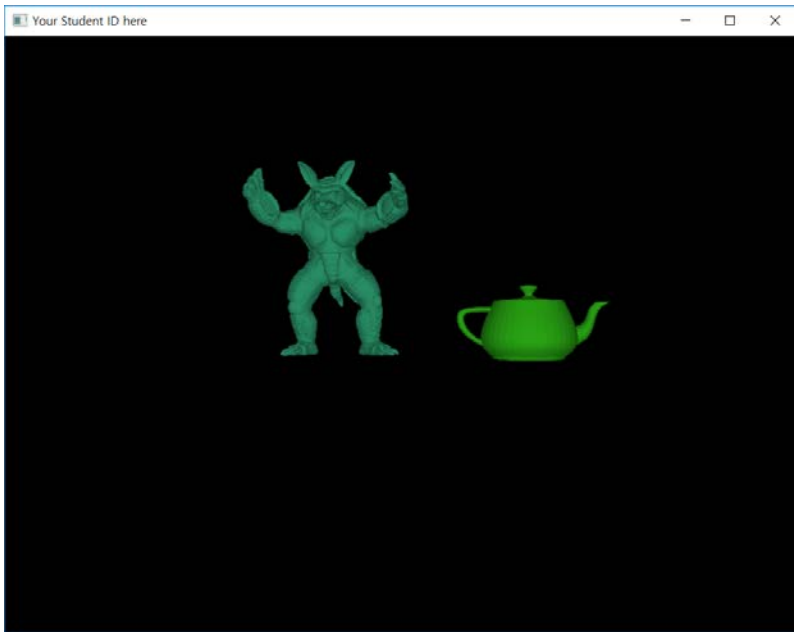
Back buffer



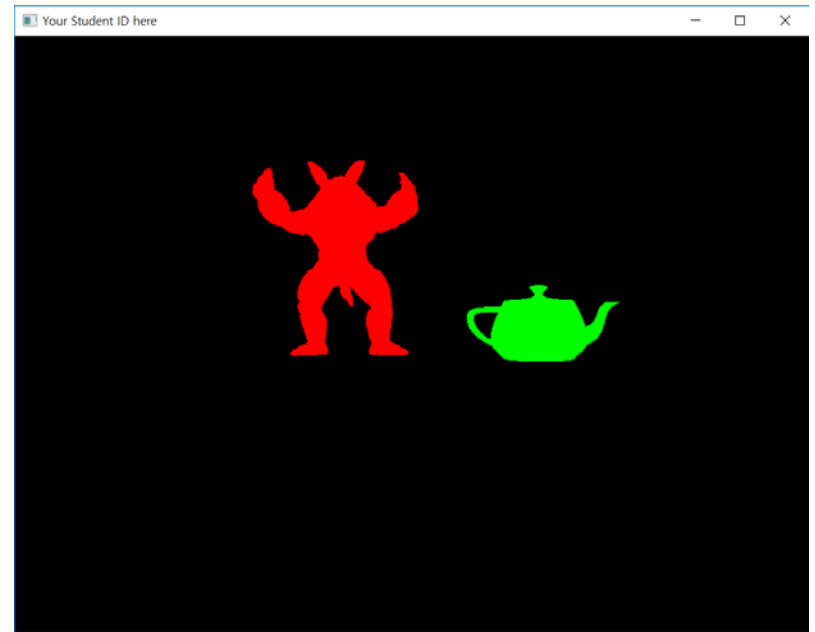
# OpenGL: Double Buffer

- Read the pixel color at the clicked position (x,y) from the back buffer
  - `glReadBuffer(mode)`
    - mode: `GL_FRONT`, `GL_BACK`
  - `unsigned char pixel[3];`
  - `glReadPixels(x, y, 1, 1, GL_RGB, GL_UNSIGNED_BYTE, pixel)`
  
- Check whether the mouse is being pressed:
  - Need to draw the back buffer (without lighting)
    - `glDrawBuffer(mode)`
      - mode: `GL_FRONT`, `GL_BACK`

# Example



Front buffer



Back buffer

# OpenGL: Double Buffer

- Pros.
  - Easy to implement it
  - Support for robust picking and selection compared to the naïve choices
- Cons.
  - Hard to handle flickering for animations, by swapping the two buffers
    - The original purpose of the double buffer is to prevent flickering.
- Selection buffers:
  - A better solution for the picking and selection
    - e.g., `glRenderMode(GL_SELECT)`
  - Require more completed implementations (will be not covered in this course)