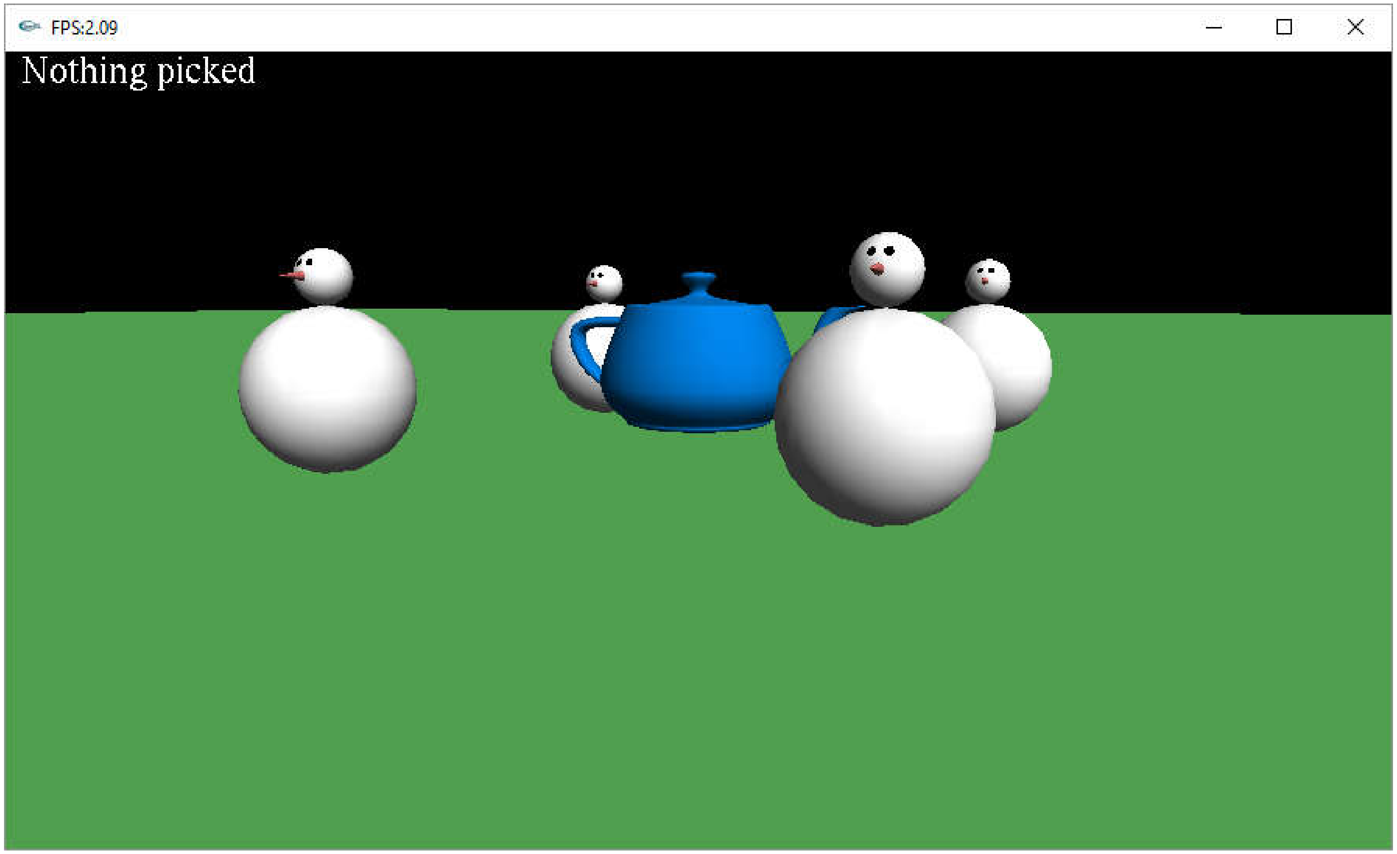
Picking and writing text

Picking

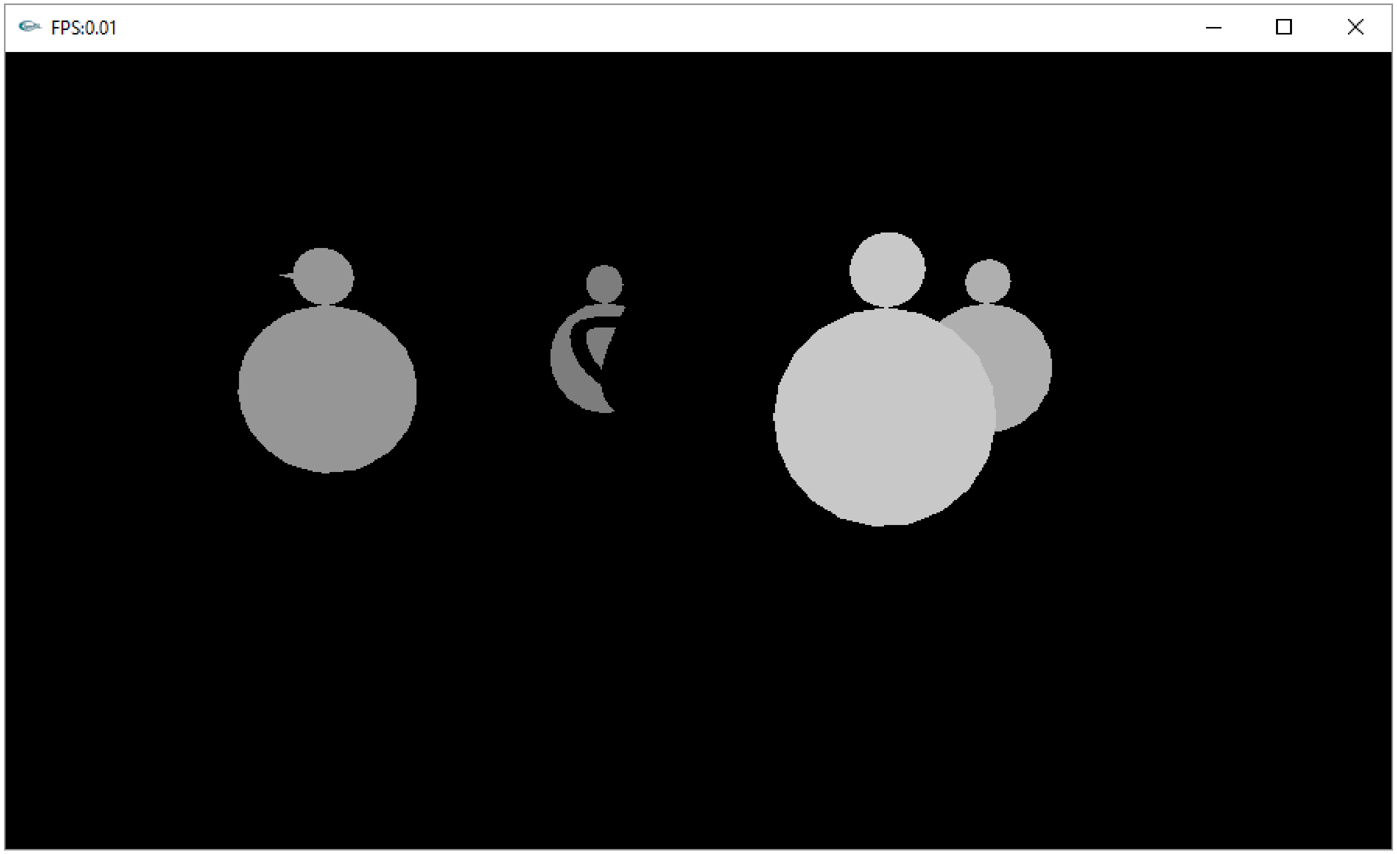
- Identifying objects on screen.



FPS:2.09

Nothing picked

Picking

- Approach:
  - 

Picking

- Algorithm
  - Turn off lighting and texturing.
  - Draw each object with a distinct color
    - Do not swap buffers
  - Read the pixel under the mouse
  - Turn on lighting and texturing.

Picking

- Mouse click starts the process

```
void processMouseButtons(int button, int state, int xx, int yy)
{
        if (state == GLUT_DOWN)  {
                if (button == GLUT_MIDLE_BUTTON) { // Middle button
                        unsigned char result = picking(xx,yy);
                        if (result)
                                printf("Picked Snowman %u\n", result);
                        else
                                printf("Nothing selected\n");
                }
        ...
}
```

Picking

- Picking function overview

  - Turn off lighting and texturing

    ```
    glDisable(GL_LIGHTING);
    glDisable(GL_TEXTURE_2D);
    ```

  - Clear the frame buffer and place the camera

    ```
    glClear(GL_COLOR_BUFFER_BIT);
    glLoadIdentity();
    gluLookAt(...);
    ```

  - Draw coded version of objects taking advantage of the values stored on the depth buffer

    ```
    glDepthFunc(GL_LEQUAL);
    .... draw
    glDepthFunc(GL_LESS);
    ```

Picking

- Picking function overview (2)

  - Read pixel under mouse position

```
GLint viewport[4];
unsigned char res[4];
glGetIntegerv(GL_VIEWPORT,viewport);
glReadPixels(x,viewport[3]- y,1,1,
               GL_RGBA,GL_UNSIGNED_BYTE, res);
```

  - Reactivate lighting and texturing

```
glEnable(GL_LIGHTING);
glEnable(GL_TEXTURE_2D);
```

  - Return red color component

```
Return res[0];
```

Writing text on screen

- Bitmap fonts using orthographic projection

- Step 1: set orthographic projection

```
glMatrixMode(GL_PROJECTION);
glPushMatrix();
glLoadIdentity();
// set projection so that coordinates match window pixels
gluOrtho2D(0, w, h, 0);
glMatrixMode(GL_MODELVIEW);
```

- Step 2: disable depth test (assuming text is written in the end)

```
glDisable(GL_DEPTH_TEST);
```

Writing text on screen

- Step 3: set modelview matrix

```
glPushMatrix();
glLoadIdentity();
glRasterPos2d(10, 20); // text position in pixels
```

Writing text on screen

- Step 4: render text

```
// set text color – which color component to choose?

char text[64];
…
 for (char *c = text; *c != '\0'; c++) {
         glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, *c);
 }
```

Writing text on screen

- Step 5: Restore matrices and reenable depth test

```
glMatrixMode(GL_PROJECTION);
glPopMatrix();
glMatrixMode(GL_MODELVIEW);
glPopMatrix();

glEnable(GL_DEPTH_TEST);

// is it required to restore the color?
```

Assignment

- Add to the source code provided the ability to pick a snowman with the mouse.
  - write function picking

- Write text on screen displaying the number of the picked snowman
  - write function renderText

Questions

- Assume GLUT did not provide text functionality.
  - How could we implement it for bitmap fonts?