How to Mow the Lawn

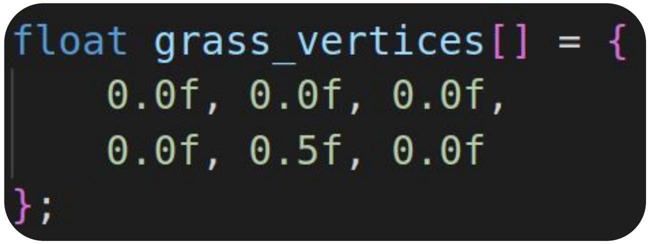
Coleby Kauffman
CS-392 GPU Programming
26 October, 2022

Overview

- Black void with a 50x50 field of grass
- Uncannily high definition lawn mower
- Configurable size and speed
- “Wind” physics, except when cut
- Inconsistent turning
- Three OpenGL “programs”:
  - Grass, Ground, Mower
- Terrible code

Part 1: The Grass

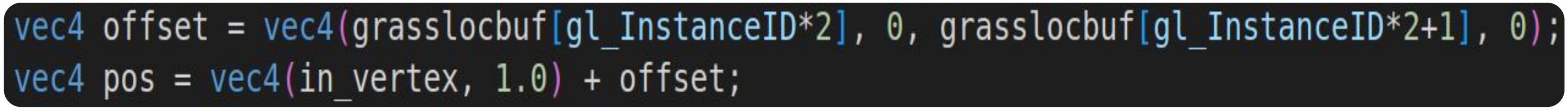
- To mow the lawn (grass), you need the lawn (grass)
- Two vertices
- GL_LINES
- Instanced rendering
    - 600,000 blades of grass
    - What if we do more? Fewer?
    - Break MLH 310 computer challenge?
    - 600,000 * sizeof(float) * 2 = 4,800,000 bytes = 4.8 MB
        - Two of these buffers
    - 100,000,000 * sizeof(float) * 2 = 800 MB
- grass_locations array
- Grass length (0.2 - 0.6)



```
float grass_vertices[] = {
    0.0f, 0.0f, 0.0f,
    0.0f, 0.5f, 0.0f
};
```

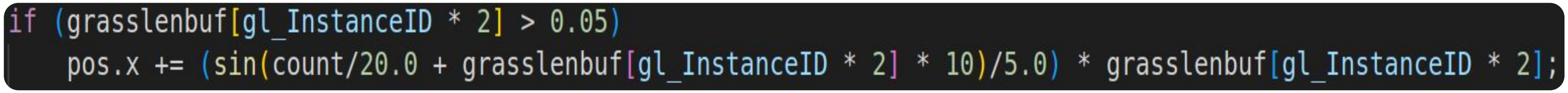
Part 1: The Grass (Part 2)

- Placing the grass



```
vec4 offset = vec4(grasslocbuf[gl_InstanceID*2], 0, grasslocbuf[gl_InstanceID*2+1], 0);
vec4 pos = vec4(in_vertex, 1.0) + offset;
```

- Wind (dancing?)

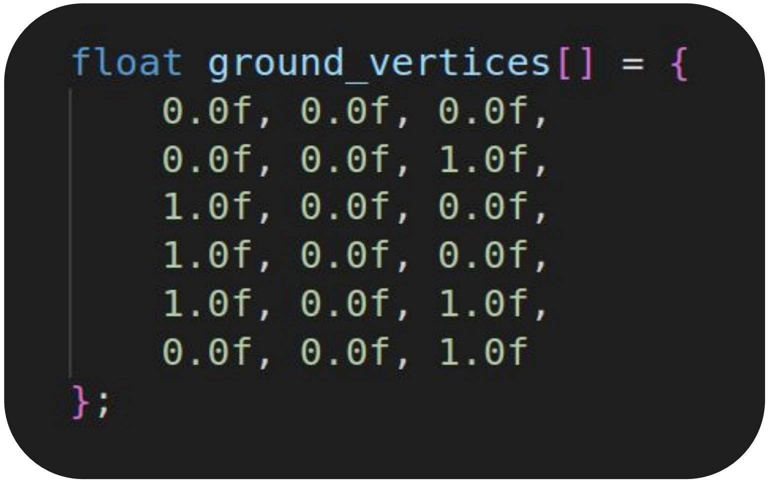


```
if (grasslenbuf[gl_InstanceID * 2] > 0.05)
    pos.x += (sin(count/20.0 + grasslenbuf[gl_InstanceID * 2] * 10)/5.0) * grasslenbuf[gl_InstanceID * 2];
```

- Issue: How much to sway?
  - Scaling wind effect with the length
- Vertex shader

Part 2: The Ground

- The ground is just two brown triangles
- GL_TRIANGLES
- Flickering geometry
  - Professional tip: do not be stupid
    - I have personal experience with this



```
float ground_vertices[] = {
    0.0f, 0.0f, 0.0f,
    0.0f, 0.0f, 1.0f,
    1.0f, 0.0f, 0.0f,
    1.0f, 0.0f, 0.0f,
    1.0f, 0.0f, 1.0f,
    0.0f, 0.0f, 1.0f
};
```

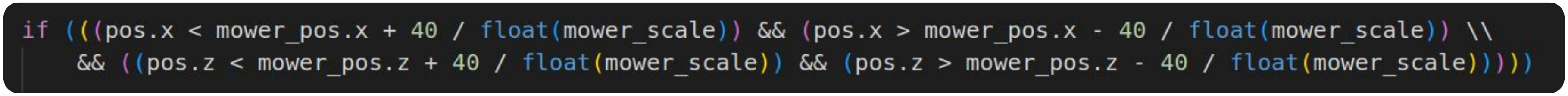
Part 3: The Mower

- One of the requirements for mowing the lawn is a lawnmower
- 3D model from free3d.com
- tinyobjloader
- Calculating the mower’s position
    - Things to consider:
        - When to turn (annoying problem)
        - Distance to next turn (annoying problem)
        - How big is the mower??? (annoying problem)
        - Model vs World space (this one’s on me...)
    - Candidate for most poorly written code of all time? Let’s look

Part 3 again: The Mower

- Collision with grass
  - Things to consider:
    - Mower “hitbox” and scaling it – `40 / float(mower_scale)` formula derivation
    - Cutting the grass and storing each blade’s state
- Mower scale and speed
  - Changing the scale was a pain…
    - Position, hitbox, mvp calculation, where to turn
    - Reset mower upon scale change
  - Speed is simple enough

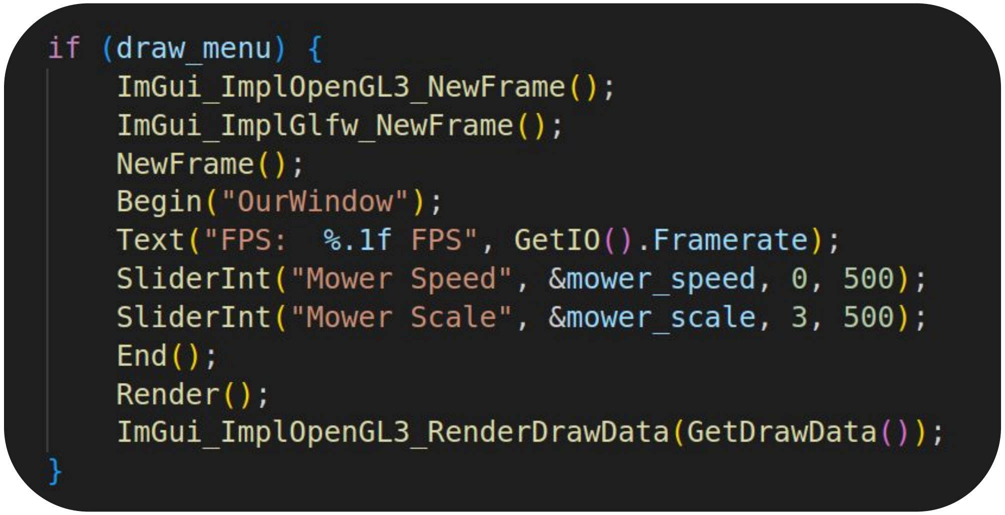
incomprehensible code



```
if (((pos.x < mower_pos.x + 40 / float(mower_scale)) && (pos.x > mower_pos.x - 40 / float(mower_scale)) \\
    && ((pos.z < mower_pos.z + 40 / float(mower_scale)) && (pos.z > mower_pos.z - 40 / float(mower_scale))))
```

Part 3.5: `ImGui_ImplGlfw_InitForOpenGL()`

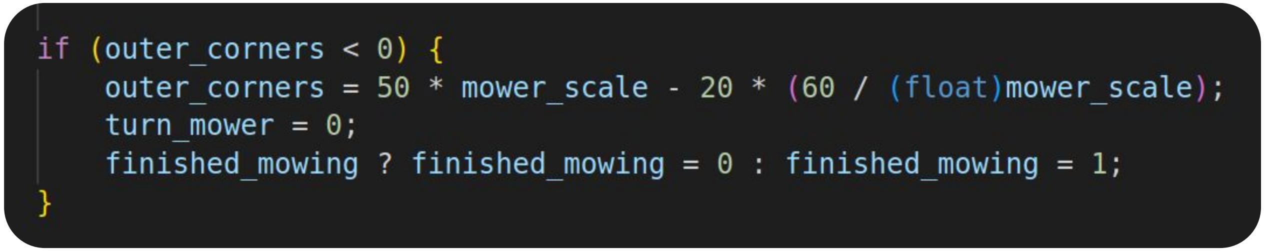
- What creates that interface box?
  - ImGui
  - Scale and Speed sliders
  - We can create abstract art
  - Comically small mower



```
if (draw_menu) {
    ImGui_ImplOpenGL3_NewFrame();
    ImGui_ImplGlfw_NewFrame();
    NewFrame();
    Begin("OurWindow");
    Text("FPS:  %.1f FPS", GetIO().Framerate);
    SliderInt("Mower Speed", &mower_speed, 0, 500);
    SliderInt("Mower Scale", &mower_scale, 3, 500);
    End();
    Render();
    ImGui_ImplOpenGL3_RenderDrawData(GetDrawData());
}
```

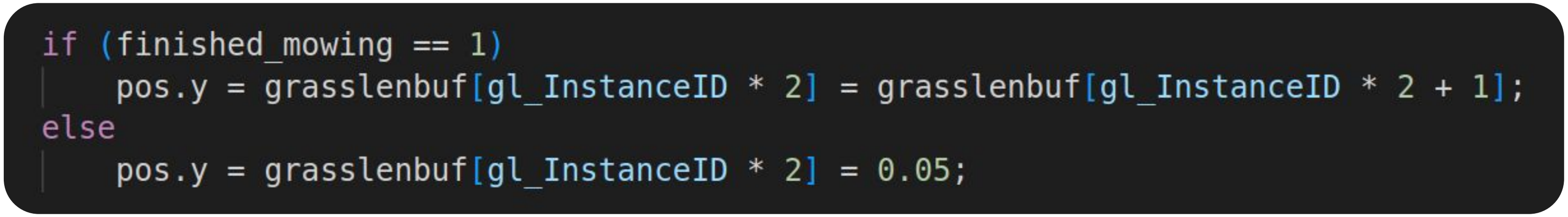
Part 4: When the Mowing is Complete 😔

- The fun has just begun
  - Un-mowing
  - finished_mowing uniform



```
if (outer_corners < 0) {
    outer_corners = 50 * mower_scale - 20 * (60 / (float)mower_scale);
    turn_mower = 0;
    finished_mowing ? finished_mowing = 0 : finished_mowing = 1;
}
```

  - Storing the original lengths



```
if (finished_mowing == 1)
    pos.y = grasslenbuf[gl_InstanceID * 2] = grasslenbuf[gl_InstanceID * 2 + 1];
else
    pos.y = grasslenbuf[gl_InstanceID * 2] = 0.05;
```