



JEO MC

Programming Languages & Translators
2021 Fall

Jeremy Lu

Emma Schwartz

Melody Hsu

Connie Zhou



Overview of JEoMC

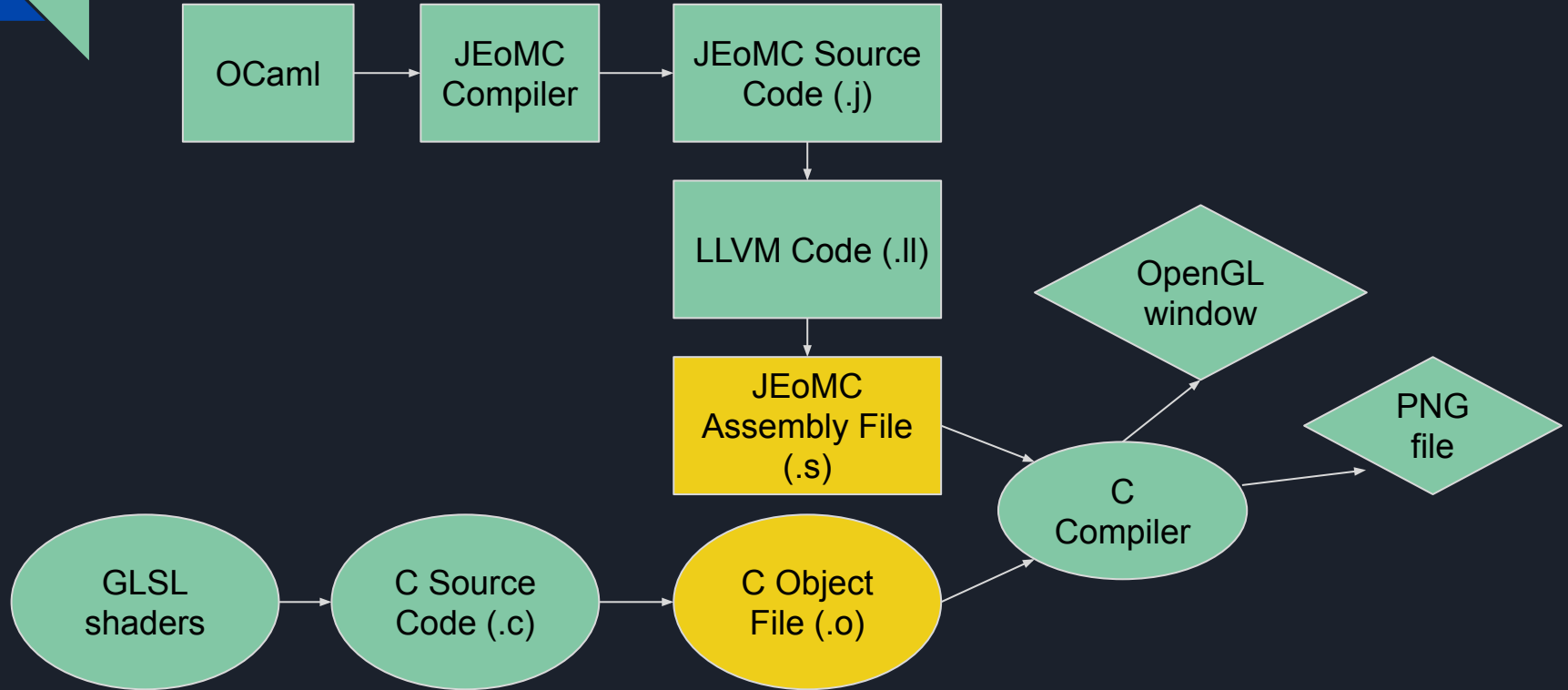
- A modular, geometric language that allows the end user to generate simple, customizable shape-based graphics, which can be viewed in an app window and saved in a PNG file
- Balance between detail and customizability, and efficiency and ease of use
- Highlights:
 - a. intuitive and easy to use
 - b. OS compatibility (MacOS & Ubuntu)
 - c. compoundable graphics



Language Structure

- Syntax: JEO MC looks like a stripped-down version of C
- Supported functionality
 - Primitives: bool, float, int, string
 - Loop control statements: for, while, break, continue
 - Simple arithmetic
 - User-defined functions
- Linkers: C, OpenGL, GLSL
- Drawing logic
 - Generate vertices with input parameters
 - Conversion of hex code strings to R,G,B floats
 - Use vertex arrays in OpenGL to draw

Building JEOCMC





OpenGL - A crash course

- OpenGL (Open Graphics Library) is an open-source library for rendering 2D/3D graphics that leverages GPU infrastructure to speed execution
- OpenGL is widely used, optimized for C/C++, available on many platforms



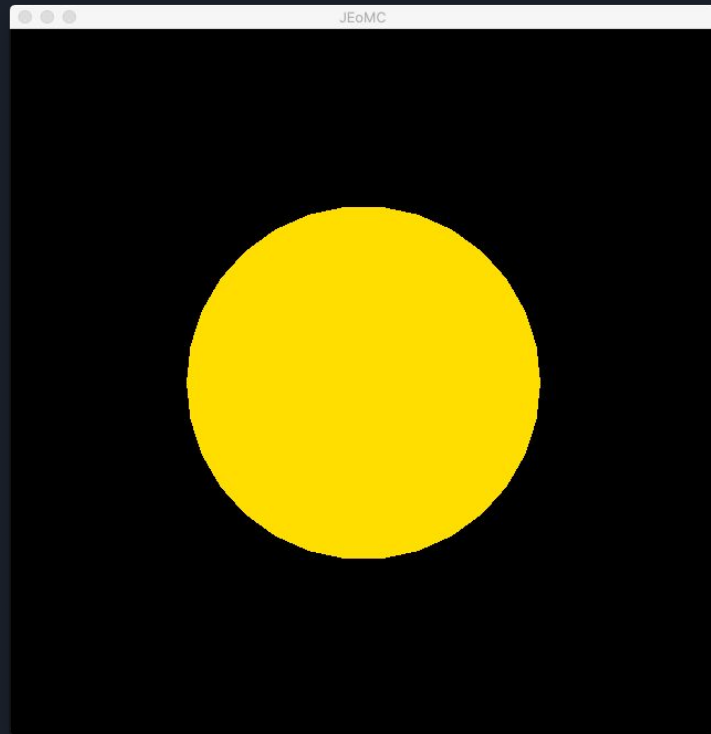
OpenGL - A crash course

- The main components of OpenGL are:
 - Array of points
 - Shaders - contribute to color
 - Buffers - control the flow
 - Windows - displays content



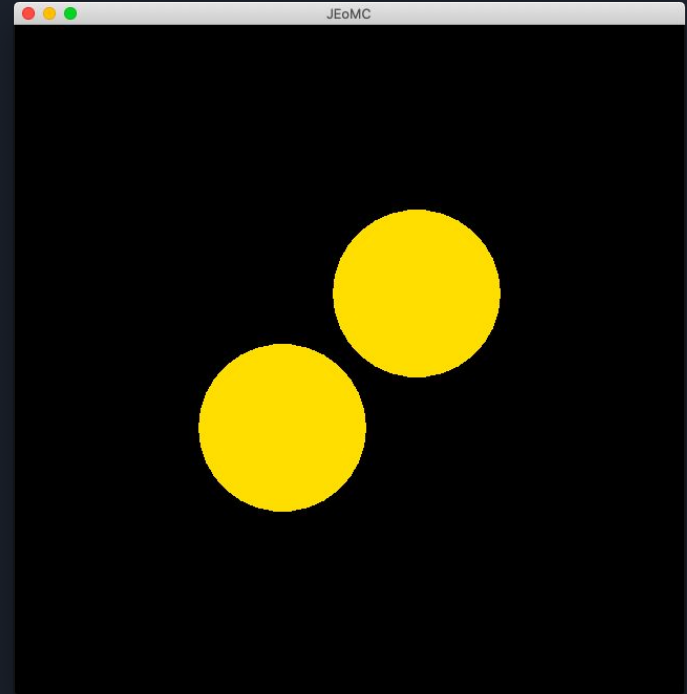
Just the basics first

```
int main()
{
    jeomcInit();
    drawCircle(0.0,0.0,0.5,"#FFDE00");
    jeomcRunAndSave();
    return 0;
}
```



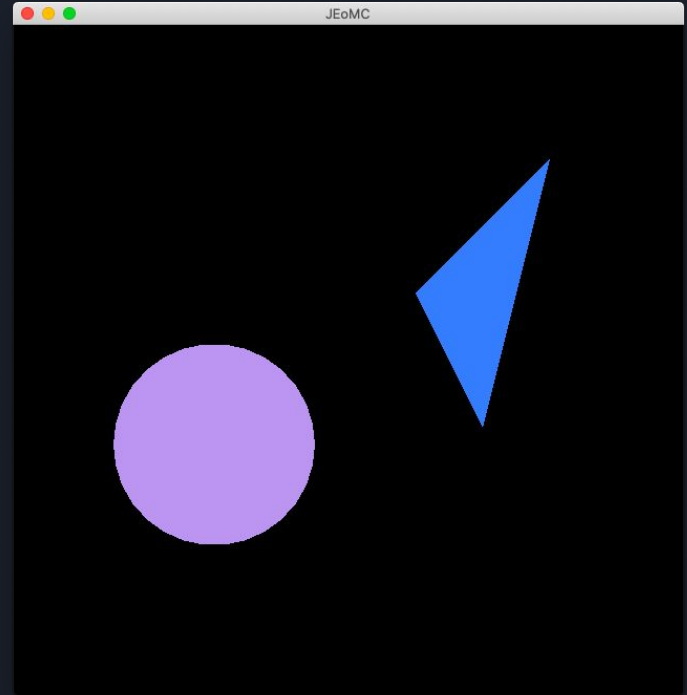
Just the basics first

```
int main()
{
    jeomcInit();
    drawCircle(-0.2,-0.2,0.25,"#FFDE00");
    drawCircle(0.2,0.2,0.25,"#FFDE00");
    jeomcRunAndSave();
    return 0;
}
```



Adding more complexity

```
int main()
{
    jeomcInit();
    drawCircle(-0.4,-0.25,0.3,"#BC94F2");
    drawTriangle(0.2,0.2,0.4,-0.2,0.6,0.6,"#347DFF");
    jeomcRunAndSave();
    return 0;
}
```



Adding more complexity

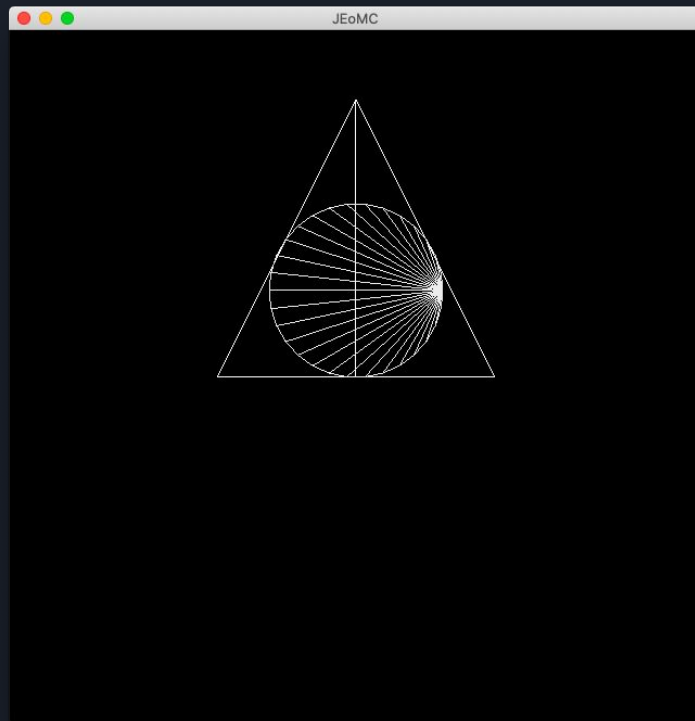
```
int main()
{
    jeomcInit();

    /* draw the cloak */
    drawLine(0.0,0.8,-0.4,0.0, "FFFFFF");
    drawLine(-0.4,0.0,0.4,0.0, "FFFFFF");
    drawLine(0.4,0.0,0.0,0.8, "FFFFFF");

    /* draw the wand */
    drawLine(0.0,0.8,0.0,0.0,"FFFFFF");

    /* draw the snitch */
    drawCircle(0.0,0.25,0.25,"#ECECEC");

    jeomcRunAndSave();
    return 0;
}
```



Adding more complexity

```
/*...*/
float i;
float j;

jeomcInit();

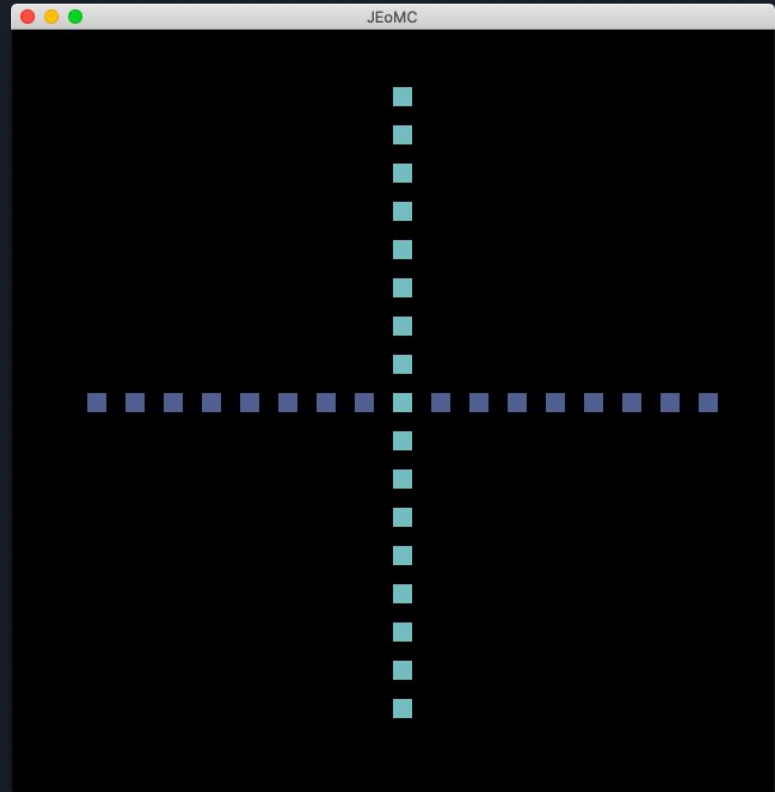
i = -0.8;
j = 0.0;

while (i < 0.8) {
    drawRectangle(i, j+0.05, i+0.05, j+0.05, i+0.05, j, i, j, "#516091");
    i = i + 0.1;
}

i = 0.0;
j = -0.8;

while (j < 0.8) {
    drawRectangle(i, j+0.05, i+0.05, j+0.05, i+0.05, j, i, j, "#74bec1");
    j = j + 0.1;
}

jeomcRunAndSave();
/*...*/
```



Putting it all together

```
int main()
{
    jeomcInit();

    drawCircle(0.0, -0.5, 0.4, "#FFFFFF");
    drawCircle(0.0, 0.1, 0.30, "#FFFFFF");
    drawCircle(0.0, 0.6, 0.20, "#FFFFFF");

    drawRectangle(-0.25, 0.32, -0.25, 0.45, 0.25, 0.45, 0.25, 0.32, "B49EC9");

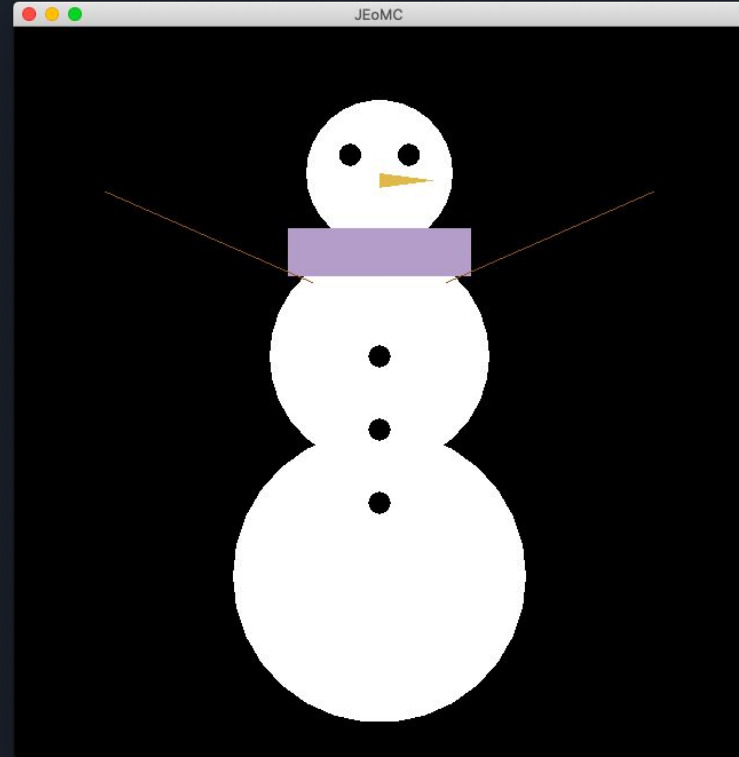
    drawCircle(-0.08, 0.65, 0.03, "#000000");
    drawCircle(0.08, 0.65, 0.03, "#000000");

    drawTriangle(0.0, 0.6, 0.15, 0.58, 0.0, 0.56, "E0B949");

    drawCircle(0.0, 0.1, 0.03, "#000000");
    drawCircle(0.0, -0.1, 0.03, "#000000");
    drawCircle(0.0, -0.3, 0.03, "#000000");

    drawLine(-0.75, 0.55, -0.18, 0.3, "9A5415");
    drawLine(0.75, 0.55, 0.18, 0.3, "9A5415");

    jeomcRunAndSave();
    return 0;
}
```





Challenges along the way

- One of the big challenges - cross-compatibility
- Library installation and version control
- Incompatibility with Docker
- OpenGL - not beginner-friendly!



LIVE DEMO!

