4- Windowing APIs.

- Open an OS window
- Init OpenGL canvas
- Capture input devices (keyboard, mouse, controllers, ...)

For OpenGL:
- GLUT (The OpenGL Utility Toolkit)
- FreeGLUT (Open-source GLUT) ← Recommended
- GLFW
- Qt ← Biggest, opinionated
- ...

Oldest

GLUT

```
#include <GL/glut.h>

int main(int argc, char** argv)
{
    // Inits
    ...

    // Call main loop
    glutMainLoop();
    return 0;
}
```

GLUT inits

```
// Init GLUT
glutInit(&argc, argv)

// Create a window
glutCreateInitWindowSize(1920, 1080);
glutInitWindowPosition(100, 100);
glutCreateWindow("W
glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA
                    | GLUT_DEPTH);
glutCreateWindow("Window Title");

// Register display callback function
glutDisplayFunc(myDisplay);
```

Double buffers

```
void myDisplay()
{
  // OpenGL draw calls here
}

// Register keyboard callback function
glutKeyboardFunc(myKeyboard);

void myKeyboard(unsigned char key, int x, int y)
{
  // Handle keyboard input here
}

// Register special keyboard callback
glutSpecialFunc(myKeyboard2);

void myKeyboard2(int key, int x, int y)
{
  // Handle keyboard input here
}

glutGetModifier(); // checks for modifier keys
                   (ctrl, alt, ...)

// Register mouse callback function
glutMouseFunc(myMouse);

void myMouse(int button, int state, int x, int y)
{
  // Handle mouse buttons here
}

// Register mouse motion callback
glutMotionFunc(myMouseMotion);

void myMouseMotion(int x, int y)
{
  // Handle mouse motion here while a button is down
}
```

```
// Register mouse "passive" motion
glutPassiveMotionFunc(myMousePassive);

void myMousePassive(int x, int y)
{
  // Handle mouse motion here while a button is
  // NOT down
}

// Register window reshape callback
glutReshapeFunc(myReshape);

void myReshape(int x, int y)
{
  // Do what you want when the window size changes
}

// Register idle callback function
glutIdleFunc(myIdle);

void myIdle()
{
  // Handle animations here
}
```

OpenGL inits

```
// Set the background color
glClearColor(0,0,0,0);

// Create buffers
// Create textures
// Compile shaders

void myDisplay()
{
  // Clear the viewport
  glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
  // OpenGL draw calls here
  // Swap buffers
  glutSwapBuffers();
}
```

Double buffering:
We have a front buffer with the image currently displayed. We make calculations to a back buffer. When the calculation is finished, we swap the two buffers. Then we clear the new back buffer and repeat. This is what glutSwapBuffers() does.

Worth noting is that all gl-commands sends a command to the GPU queue and then returns immediately. Then it calls the idle function.

```
void myIdle()
{
    // Example animation
    glClearColor(red, 0, 0, 0);
    red += 0.02;

    // Tell GLUT to redraw
    glutPostRedisplay();
}
```

Keyboard function:

```
void myKeyboard(unsigned char key, int x, int y)
{
    switch(key) {
        case 27: //ESC
            glutLeaveMainLoop();
            break;
        case 'a':
            // Do something
            break
    }
    glutPostRedisplay();
}
```