Introduction to
GLEW, GLFW and SDL

What is GLEW?

- OpenGL Extension Wrangler
- Interface for OpenGL versions above 1.1
- Load OpenGL extensions
- Some extensions are platform specific, GLEW can check if they exist on that platform
- Alternatives: GL3W, glLoadGen, glad, glsdk, glbinding, libepoxy, Glee
- Best to use GLEW. Most people do.

Using GLEW

- #include <GL/glew.h>
- After initialisation OpenGL context:
  glewExperimental = GL_TRUE;
- glewInit();
- Should return GLEW_OK. If it fails, it returns the error.
- Can read error with glewGetErrorString(result);
- Check extensions exist:
  if (!GLEW_EXT_framebuffer_object){}
- wglew.h for Windows only functions

GLFW

- OpenGL FrameWork (...probably)
- Handles window creation and control
- Pick up and process input from the keyboard, mouse, joystick and gamepad
- Even allows multiple monitor support!
- Uses OpenGL context for windows

SDL

- Simple DirectMedia Layer
- Can do almost everything GLFW can do…
- and more! (Audio, Threading, Filesystems, etc.)
- Very popular, especially for Indie developers
- Used in: FTL, Amnesia, Starbound and Dying Light
- Even used in level editors for Source Engine and Cryengine!

Alternatives

- SFML (Simple and Fast Multimedia Library): Like SDL but with even more features
- ...but the OpenGL context is very weak. Based on 2D only graphics.
- GLUT (OpenGL Utility Toolkit): Is no longer maintained. Try to avoid it.
- Win32 API: For the purists. Lowest level for window creation. Only attempt if you know what you’re doing!

Summary

- GLEW (OpenGL Extension Wrangler) lets us interface with modern OpenGL and handle platform-specific extensions safely
- GLFW lets us create windows and OpenGL contexts, as well as handle user input
- SDL does all that GLFW does, and more
- The additional features of SDL are beyond the scope of this course, so we will be using GLFW

See you next video!