01 Course Introduction

Functional and Implementation Guidelines

Functional Guidelines

1. Install all needed development libraries
2. Install a C compiler and the cmake tool
3. Create your first “HelloWorld” graphical project
   1. Create a single window
   2. Display a single image on that window
   3. Sleep your application so you can see the image being displayed
   4. Destroy everything accordingly and quit the application gracefully

Implementation Guidelines

1. **Ubuntu Basics - Lecture**
2. Create a new C++ project
3. Create an empty main file
4. Copy the assets into the project
5. **CMake Tool Lecture**
6. Install needed packages
   1. build-essentials
   2. cmake
   3. libsdl2-dev
   4. libsdl2-image-dev
   5. libsdl2-ttf-dev
   6. libsdl2-mixer-dev
7. Utils CMake structure
   1. Create CMakeFileLists.txt file for the utils static library
   2. project(... LANGUAGES CXX)
   3. add\_library(... STATIC HEADERS SOURCES)
   4. target\_link\_libraries(... PRIVATE ...)
8. Copy the content of cmake\_helpers
9. Create root project CMakeLists.txt file
   1. project(... LANGUAGES CXX)
   2. include helpeprs\_c.cmake

| include(${CMAKE\_CURRENT\_SOURCE\_DIR}/cmake\_helpers/helpers.cmake) |
| --- |

* 1. append find\_modules folder to ${CMAKE\_MODULE\_PATH}

| set(CMAKE\_MODULE\_PATH  ${CMAKE\_MODULE\_PATH} ${CMAKE\_CURRENT\_SOURCE\_DIR}/cmake\_helpers/find\_modules) |
| --- |

* 1. Use find\_package

| find\_package(SDL2 REQUIRED)  find\_package(SDL2\_image REQUIRED)  find\_package(SDL2\_ttf REQUIRED)  find\_package(SDL2\_mixer REQUIRED) |
| --- |

* 1. add\_executable(... HEADERS SOURCES)
  2. set\_target\_cpp\_standard(), enable\_target\_cpp\_warnings()
  3. target\_include\_directories(..., SDL2\_INCLUDE\_DIR, SDL2\_IMAGE\_INCLUDE\_DIR, SDL2\_TTF\_INCLUDE\_DIR, SDL2\_MIXER\_INCLUDE\_DIR)
  4. target\_link\_libraries(... , SDL2\_LIBRARY, SDL2\_IMAGE\_LIBRARY, SDL2\_TTF\_LIBRARY, SDL2\_MIXER\_LIBRARY)

1. Implement the main file - Hello, World!
2. Global vars SDL\_Window\* gWindow, SDL\_Surface\* globalScreenSurface, SDL\_Surface\* globalImage
3. Use the <stdint.h> header everywhere and check for errors
4. Use the <SDL.h> header
5. Use the SDL\_GetError() method for every SDL related error checking
6. init()
   1. SDL\_Init(SDL\_INIT\_VIDEO)
   2. SDL\_CreateWindow() //use SDL\_WINDOW\_SHOWN as flags
   3. SDL\_GetWindowSurface()
7. deinit()
   1. SDL\_FreeSurface()
   2. SDL\_DestroyWindow()
   3. SDL\_Quit()
8. loadResources()
   1. SDL\_LoadBMP()
9. draw()
   1. SDL\_BlitSurface()
   2. SDL\_UpdateWindowSurface()
   3. SDL\_Delay()
10. BONUS: For homework try to not use global non-constant variables.