## Instructions for Circuit Land Installation

Circuit Land, the game is written in C++ using the SDL2 framework (<https://www.libsdl.org/>)

Other considerations for development platform for circuit were

1. Unity
2. Unreal
3. Godot

Considering flexibility, time constraints SDL2 based game development was chosen to focus on

1. Game Mechanics
2. Assets Creation
3. Prototyping timeline

### Game Development Environment

Circuit Land has been developed on windows platform with WSL. It has been independently tested on a separate Linux PC, so development and game play identified on Linux machine should work.

Mac based development and game testing has not been done

VS code has been used as code editor

### Dependency installation

SDL2 on Linux Debian based

$ sudo apt-get install libsdl2-dev libsdl2-image-dev libsdl2-ttf-dev

### Code Organization

CircuitLand.cpp : Main source file containing implementation of all helper functions and business logic of the game.

rsdl.cpp : Wrapper helper classes to bring in SDL2 functionality in a concise manner. Simplifying usage and referencing in CircuitLand.cpp

audio.cpp: SDL2 audio support abstracted with class to handle audio playback.

Image\_Assets Folder: All game assets created and referenced by the Circuit Land Game implementation

### Game Compilation

g++ -o CircuitLand audio.cpp CircuitLand.cpp rsdl.cpp -l SDL2 -l SDL2\_ttf -l SDL2\_image

### Game running

./CircuitLand

### Open Issues

1. Web Assembly Creation has not been completed. Usage of emscripten (<https://emscripten.org/>) and creation of Web Assembly has issues to deliver the game via web currently.
2. Docker instance has not been completed to play game via docker.

References

<https://www.libsdl.org/>

<https://www.youtube.com/watch?v=m4Pow9YsDyc&list=PLVotA8ycjnCs3DNWIbEIpBrjYkhJq11q->

<https://github.com/libsdl-org/SDL/releases>