**Core API**

The core API allows to create widgets. Some widgets can contain subwidgets. Widgets are independent from the graphical backend, but feature some layout informations.

**Basic Layout**

A basic layout algorithm is available to position widgets according to their layout constraints.

**OpenGL backend**

A backend is implemented using OpenGL, to draw widgets.

---

**Level 2**

Level 1 must be unlocked to read this section

* **Developer Documentation**
  
  Document your project (not necessarily only in the source code) so that a new coming developer could understand and contribute to the code.

* **Release**
  
  Produce a release as a source archive or git tag. The release files should have up-to-date README and INSTALL files and more generally allow anyone to deploy the application.

**Fonts**

Text can be rendered inside some widgets, e.g. for button’s text.

**Layout**

A layout algorithm is available to position widgets according to their layout constraints.

**Events**

Events can be sent to a widget, propagating through the hierarchy.

**Unit tests**

The core mechanisms and the layout algorithm should be tested. Negative tests (e.g. impossible layouts) should not be omitted.

---

**PenGUI**

An immediate mode GUI API in Rust.

**Skills**

- **OpenGL**
  
  
  ![OpenGL Skill](skill_level)

- **GUI**
  
  
  ![GUI Skill](skill_level)

- **Rust**
  
  
  ![Rust Skill](skill_level)

(*)[The skill scale is from 0 (Fundamental Awareness) to 6 (Expert).]