

Fyrox Book

Dmitry Stepanov

2024-06-20

Contents

1	Introduction to Fyrox	5
1.1	What can the engine do?	5
1.2	How does the engine work?	6
1.3	Programming languages	6
1.4	Engine Features	7
2	Hello bookdown	11
2.1	A section	11
3	Cross-references	13
3.1	Chapters and sub-chapters	13
3.2	Captioned figures and tables	13
4	Parts	17
5	Footnotes and citations	19
5.1	Footnotes	19
5.2	Citations	19
6	Blocks	21
6.1	Equations	21
6.2	Theorems and proofs	21
6.3	Callout blocks	21
7	Sharing your book	23
7.1	Publishing	23
7.2	404 pages	23
7.3	Metadata for sharing	23

Chapter 1

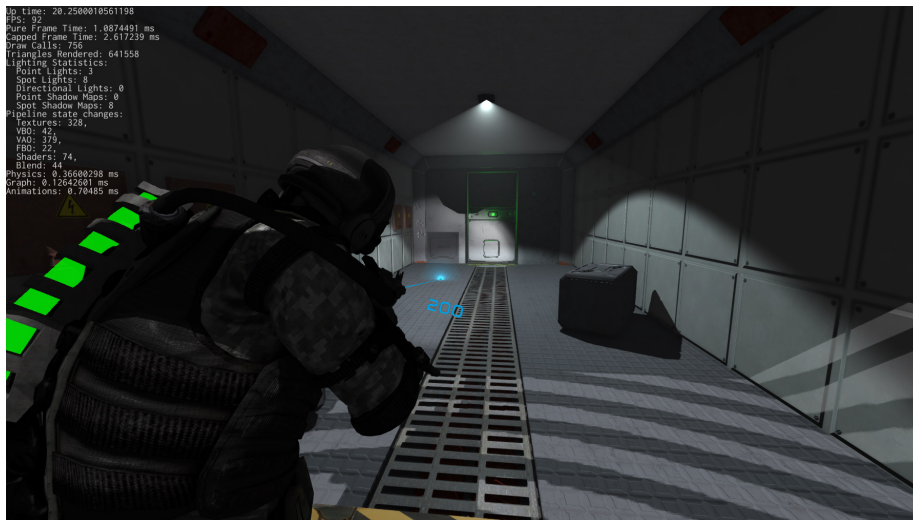
Introduction to Fyrox

Fyrox is a feature-rich, general purpose game engine that is suitable for any kind of games. It is capable to power games with small- or medium-sized worlds, large-sized world most likely will require some manual work.

Games made with the engine are capable to run on desktop platforms (PC, Mac, Linux) and Web (WebAssembly). Mobile is planned for future releases.

1.1 What can the engine do?

You can create pretty much any kind of game or interactive applications. Here's some examples of what the engine can do:





1.2 How does the engine work?

The engine consists of two parts that you'll be actively using: the framework and the editor. The framework is a foundation of the engine, it manages rendering, sound, scripts, plugins, etc. While the editor contains lots of tools that can be used to create game worlds, manage assets, edit game objects, scripts and more.

1.3 Programming languages

Everything of your game can be written entirely in Rust, utilizing its safety guarantees as well as speed. However, it is possible to use any scripting language you want, but that's have no built-in support, and you need to implement this

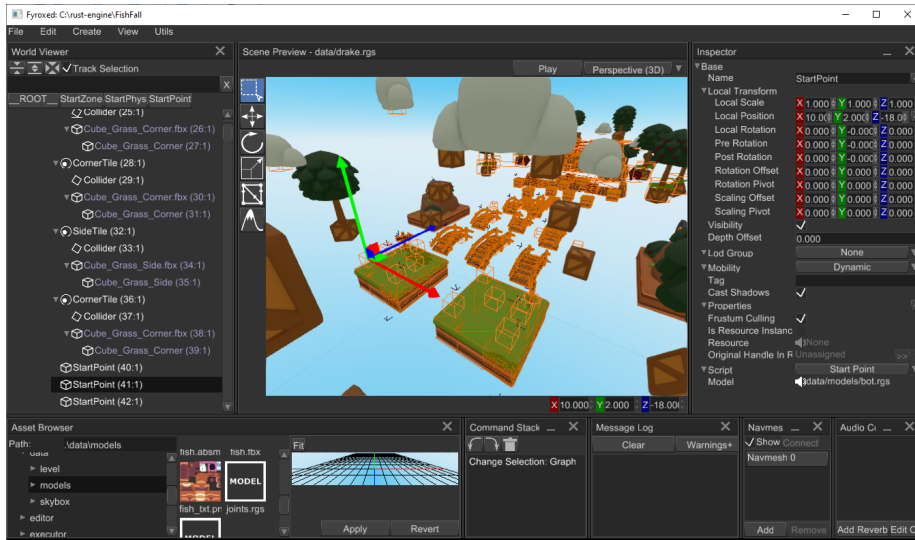


Figure 1.1: Fish Folly

manually.

1.4 Engine Features

This is a more or less complete (yet, it can be outdated) list of engine features:

1.4.1 General

- Exceptional safety, reliability, and speed.
- PC (Windows, Linux, macOS), Android, Web (WebAssembly) support.
- Modern, PBR rendering pipeline.
- Comprehensive documentation.
- Guide book
- 2D support.
- Integrated editor.
- Fast iterative compilation.
- Classic object-oriented design.
- Lots of examples.

1.4.2 Rendering

- Custom shaders, materials, and rendering techniques.
- Physically-based rendering.
- Metallic workflow.
- High dynamic range (HDR) rendering.

- Tone mapping.
- Color grading.
- Auto-exposure.
- Gamma correction.
- Deferred shading.
- Directional light.
- Point lights + shadows.
- Spotlights + shadows.
- Screen-Space Ambient Occlusion (SSAO).
- Soft shadows.
- Volumetric light (spot, point).
- Batching.
- Instancing.
- Fast Approximate Anti-Aliasing (FXAA).
- Normal mapping.
- Parallax mapping.
- Render in texture.
- Forward rendering for transparent objects.
- Sky box.
- Deferred decals.
- Multi-camera rendering.
- Lightmapping.
- Soft particles.
- Fully customizable vertex format.
- Compressed textures support.
- High-quality mip-map on-demand generation.

1.4.3 Scene

- Multiple scenes.
- Full-featured scene graph.
- Level-of-detail (LOD) support.
- GPU Skinning.
- Various scene nodes:
 - Pivot.
 - Camera.
 - Decal.
 - Mesh.
 - Particle system.
 - Sprite.
 - Multilayer terrain.
 - Rectangle (2D Sprites)
 - Rigid body + Rigid Body 2D
 - Collider + Collider 2D
 - Joint + Joint 2D

1.4.4 Sound

- High quality binaural sound with HRTF support.
- Generic and spatial sound sources.
- Built-in streaming for large sounds.
- Raw samples playback support.
- WAV/OGG format support.
- HRTF support for excellent positioning and binaural effects.
- Reverb effect.

1.4.5 Serialization

- Powerful serialization system
- Almost every entity of the engine can be serialized
- No need to write your own serialization.

1.4.6 Animation

- Animation blending state machine - similar to Mecanim in Unity Engine.
- Animation retargetting - allows you to remap animation from one model to another.

1.4.7 Asset management

- Advanced asset manager.
- Fully asynchronous asset loading.
- PNG, JPG, TGA, DDS, etc. textures.
- FBX models loader.
- WAV, OGG sound formats.
- Compressed textures support (DXT1, DXT3, DTX5).

1.4.8 Artificial Intelligence (AI)

- A* pathfinder.
- Navmesh.
- Behavior trees.

1.4.9 User Interface (UI)

- Advanced node-based UI with lots of widgets.
- More than 32 widgets
- Powerful layout system.
- Full TTF/OTF fonts support.
- Based on message passing.
- Fully customizable.
- GAPI-agnostic.
- OS-agnostic.

- Button widget.
- Border widget.
- Canvas widget.
- Color picker widget.
- Color field widget.
- Check box widget.
- Decorator widget.
- Drop-down list widget.
- Grid widget.
- Image widget.
- List view widget.
- Popup widget.
- Progress bar widget.
- Scroll bar widget.
- Scroll panel widget.
- Scroll viewer widget.
- Stack panel widget.
- Tab control widget.
- Text widget.
- Text box widget.
- Tree widget.
- Window widget.
- File browser widget.
- File selector widget.
- Docking manager widget.
- NumericUpDown widget.
- `Vector3<f32>` editor widget.
- Menu widget.
- Menu item widget.
- Message box widget.
- Wrap panel widget.
- Curve editor widget.
- User defined widget.

1.4.10 Physics

- Advanced physics (thanks to the rapier physics engine)
- Rigid bodies.
- Rich set of various colliders.
- Joints.
- Ray cast.
- Many other useful features.
- 2D support.

Chapter 2

Hello bookdown

All chapters start with a first-level heading followed by your chapter title, like the line above. There should be only one first-level heading (#) per .Rmd file.

2.1 A section

All chapter sections start with a second-level (##) or higher heading followed by your section title, like the sections above and below here. You can have as many as you want within a chapter.

An unnumbered section

Chapters and sections are numbered by default. To un-number a heading, add a {.unnumbered} or the shorter {-} at the end of the heading, like in this section.

Chapter 3

Cross-references

Cross-references make it easier for your readers to find and link to elements in your book.

3.1 Chapters and sub-chapters

There are two steps to cross-reference any heading:

1. Label the heading: `# Hello world {#nice-label}`.
 - Leave the label off if you like the automated heading generated based on your heading title: for example, `# Hello world = # Hello world {#hello-world}`.
 - To label an un-numbered heading, use: `# Hello world {-#nice-label}` or `{# Hello world .unnumbered}`.
2. Next, reference the labeled heading anywhere in the text using `\@ref(nice-label)`; for example, please see Chapter 3.
 - If you prefer text as the link instead of a numbered reference use: any text you want can go here.

3.2 Captioned figures and tables

Figures and tables *with captions* can also be cross-referenced from elsewhere in your book using `\@ref(fig:chunk-label)` and `\@ref(tab:chunk-label)`, respectively.

See Figure 3.1.

```
par(mar = c(4, 4, .1, .1))
plot(pressure, type = 'b', pch = 19)
```

Don't miss Table 3.1.

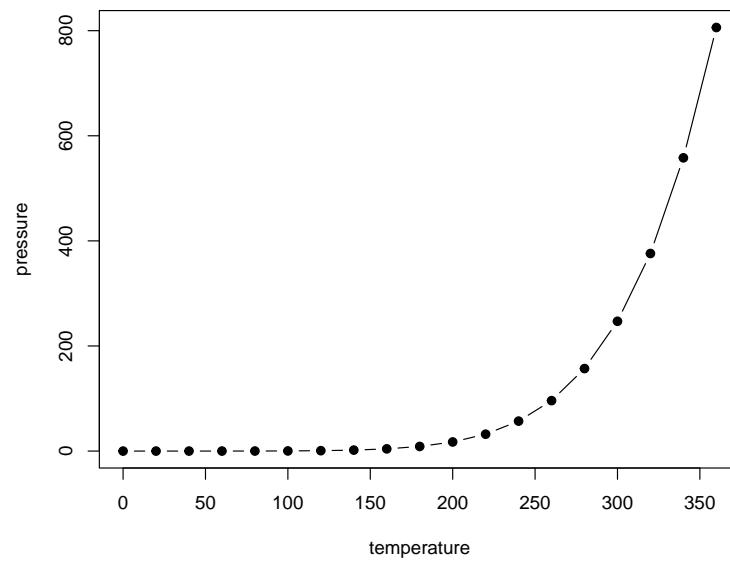


Figure 3.1: Here is a nice figure!

```
knitr::kable(  
  head(pressure, 10), caption = 'Here is a nice table!',  
  booktabs = TRUE  
)
```

Table 3.1: Here is a nice table!

temperature	pressure
0	0.0002
20	0.0012
40	0.0060
60	0.0300
80	0.0900
100	0.2700
120	0.7500
140	1.8500
160	4.2000
180	8.8000

Chapter 4

Parts

You can add parts to organize one or more book chapters together. Parts can be inserted at the top of an .Rmd file, before the first-level chapter heading in that same file.

Add a numbered part: `# (PART) Act one {-}` (followed by `# A chapter`)

Add an unnumbered part: `# (PART*) Act one {-}` (followed by `# A chapter`)

Add an appendix as a special kind of un-numbered part: `# (APPENDIX) Other stuff {-}` (followed by `# A chapter`). Chapters in an appendix are prepended with letters instead of numbers.

Chapter 5

Footnotes and citations

5.1 Footnotes

Footnotes are put inside the square brackets after a caret `^[]`. Like this one ¹.

5.2 Citations

Reference items in your bibliography file(s) using `@key`.

For example, we are using the **bookdown** package [Xie, 2024] (check out the last code chunk in `index.Rmd` to see how this citation key was added) in this sample book, which was built on top of R Markdown and **knitr** [Xie, 2015] (this citation was added manually in an external file `book.bib`). Note that the `.bib` files need to be listed in the `index.Rmd` with the YAML `bibliography` key.

The RStudio Visual Markdown Editor can also make it easier to insert citations: <https://rstudio.github.io/visual-markdown-editing/#/citations>

¹This is a footnote.

Chapter 6

Blocks

6.1 Equations

Here is an equation.

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k} \quad (6.1)$$

You may refer to using `\@ref{eq:binom}`, like see Equation (6.1).

6.2 Theorems and proofs

Labeled theorems can be referenced in text using `\@ref{thm:tri}`, for example, check out this smart theorem 6.1.

Theorem 6.1. *For a right triangle, if c denotes the length of the hypotenuse and a and b denote the lengths of the **other** two sides, we have*

$$a^2 + b^2 = c^2$$

Read more here <https://bookdown.org/yihui/bookdown/markdown-extensions-by-bookdown.html>.

6.3 Callout blocks

The R Markdown Cookbook provides more help on how to use custom blocks to design your own callouts: <https://bookdown.org/yihui/rmarkdown-cookbook/custom-blocks.html>

Chapter 7

Sharing your book

7.1 Publishing

HTML books can be published online, see: <https://bookdown.org/yihui/bookdown/publishing.html>

7.2 404 pages

By default, users will be directed to a 404 page if they try to access a webpage that cannot be found. If you'd like to customize your 404 page instead of using the default, you may add either a `_404.Rmd` or `_404.md` file to your project root and use code and/or Markdown syntax.

7.3 Metadata for sharing

Bookdown HTML books will provide HTML metadata for social sharing on platforms like Twitter, Facebook, and LinkedIn, using information you provide in the `index.Rmd` YAML. To setup, set the `url` for your book and the path to your `cover-image` file. Your book's `title` and `description` are also used.

This `gitbook` uses the same social sharing data across all chapters in your book—all links shared will look the same.

Specify your book's source repository on GitHub using the `edit` key under the configuration options in the `_output.yml` file, which allows users to suggest an edit by linking to a chapter's source file.

Read more about the features of this output format here:

<https://pkgs.rstudio.com/bookdown/reference/gitbook.html>

Or use:

```
?bookdown:::gitbook
```


Bibliography

Yihui Xie. *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition, 2015. URL <http://yihui.org/knitr/>. ISBN 978-1498716963.

Yihui Xie. *bookdown: Authoring Books and Technical Documents with R Markdown*, 2024. URL <https://github.com/rstudio/bookdown>. R package version 0.39.