automating raylib for the future

Ramon Santamaria (@raysan5)

Why automation?

BCN GAME FEST

Processes automation has been key for raylib sustainability

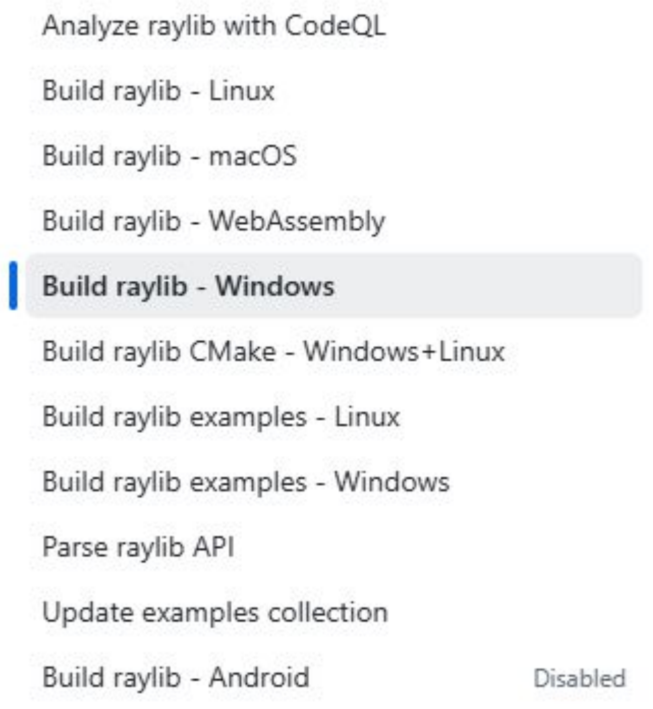
BCN GAME FEST

raylib automation processes

- Library building and deployment (CI/CD)
- Semantic code analysis (CodeQL)
- Bindings creation automation
- **Examples collection management**
    - Custom pipeline development
- **Future raylib pipelines**

raylib automation: library building (CI/CD)

- **11 workflows**
- Run on every commit
- Verify compilation
- +20 library versions generated



Analyze raylib with CodeQL
Build raylib - Linux
Build raylib - macOS
Build raylib - WebAssembly
Build raylib - Windows
Build raylib CMake - Windows+Linux
Build raylib examples - Linux
Build raylib examples - Windows
Parse raylib API
Update examples collection
Build raylib - Android Disabled

BCN GAME FEST

raylib automation: library building (CI/CD)

Windows	**Linux**	**macOS**	**Web**
x86_mingw-w64	x86	x86_64	wasm32
x64_mingw-w64	x86_64	arm64	
x64_msvc16	arm64		
x64_msvc16			
arm64_msvc16			

Static+Dynamic (x2)

raylib automation: code analysis (CodeQL)

- **Find potential security vulnerabilities**
- Detect code quality issues
- Supports configurable rules set
- Free for research and open-source

raylib automation: bindings creation

raylib has bindings to +70 programming languages

raylib automation: bindings creation

rlp

rlparser

raylib header API parser

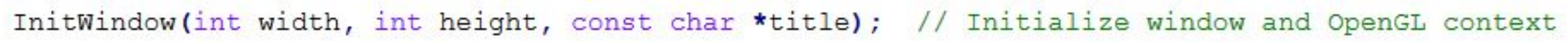
BCN GAME FEST

raylib automation: bindings creation: rlparser

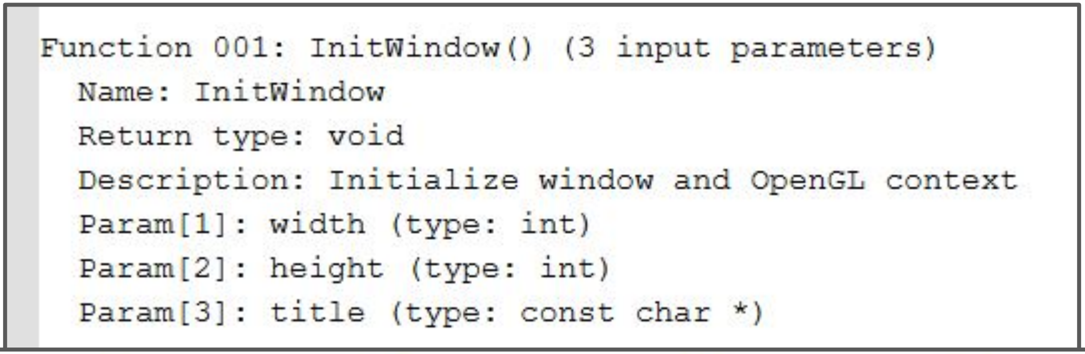
- Parses [raylib.h] to structured data
- Exports data as TXT, JSON, XML, CODE…
- Generates structured data useful to **simplify bindings creation**
- Supports other C libraries!

BCN GAME FEST

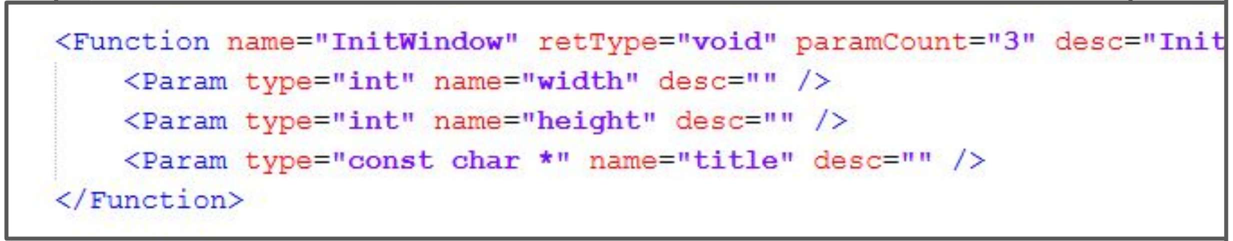
raylib automation: bindings creation: **rlparser**



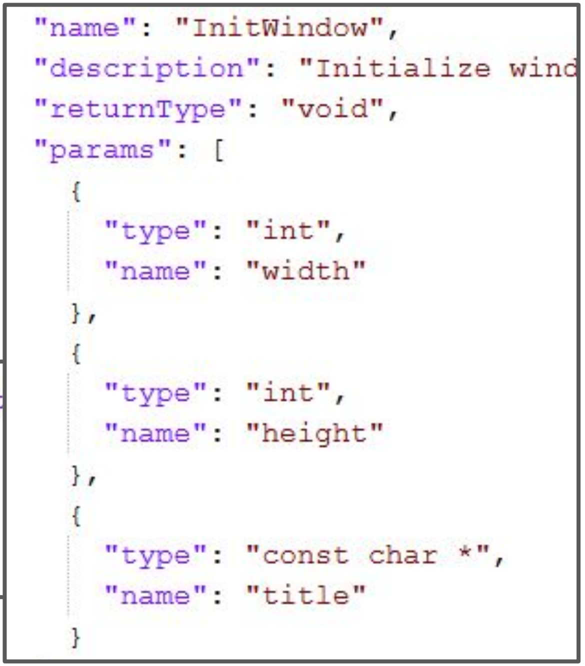
```
InitWindow(int width, int height, const char *title);  // Initialize window and OpenGL context
```



```
Function 001: InitWindow() (3 input parameters)
  Name: InitWindow
  Return type: void
  Description: Initialize window and OpenGL context
  Param[1]: width (type: int)
  Param[2]: height (type: int)
  Param[3]: title (type: const char *)
```



```
<Function name="InitWindow" retType="void" paramCount="3" desc="Init
    <Param type="int" name="width" desc="" />
    <Param type="int" name="height" desc="" />
    <Param type="const char *" name="title" desc="" />
</Function>
```

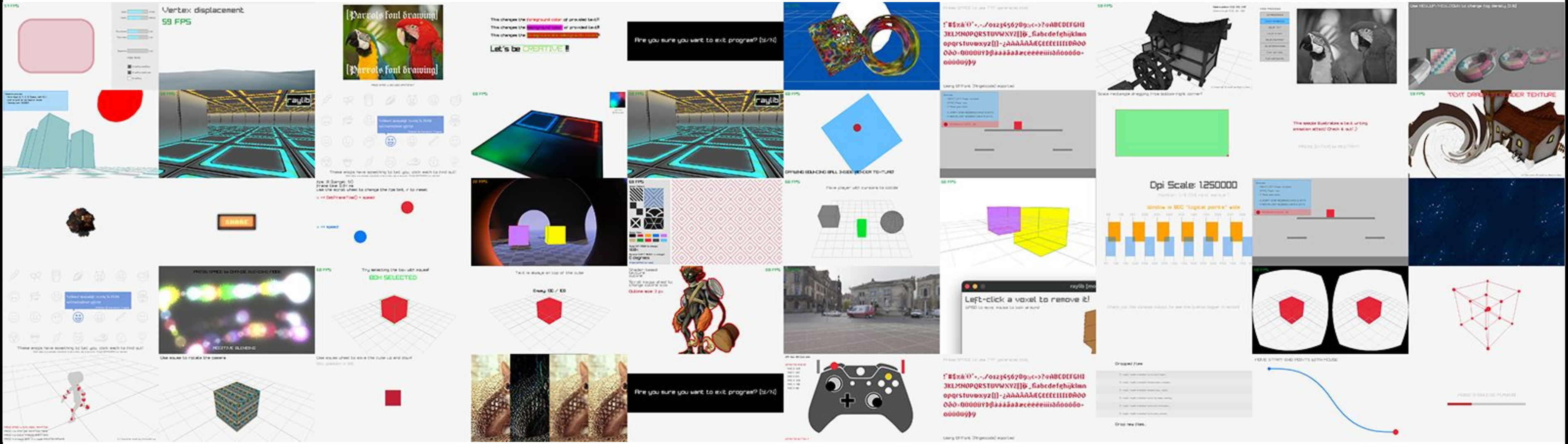


```
"name": "InitWindow",
"description": "Initialize wind
"returnType": "void",
"params": [
  {
    "type": "int",
    "name": "width"
  },
  {
    "type": "int",
    "name": "height"
  },
  {
    "type": "const char *",
    "name": "title"
  }
```

BCN GAME FEST

raylib automation: examples management

raylib has a collection of +170 examples!



BCN GAME FEST

raylib automation: examples management

- **Examples: main learning resource for raylib**
- Adding/changing examples is not trivial
- **Many elements** involved with every example
- Examples are a **key area for contributors**
    - **40%-50% contributed by community!**
- Managing full collection is **time-consuming**

BCN GAME FEST

raylib automation: examples management

What (many) contributors think adding a new example implies:

`my_new_cool_example.c`

BCN GAME FEST

raylib automation: examples management

What adding a new example REALLY implies:

- Follow code structure + metadata
- Review **code conventions**
- Provide screenshot (800x450)
- Update **build systems**
  - Makefile, VS2022, WEB
- Update required docs
- **Update webpage**, upload example

<category>_cool_example.c
<category>_cool_example.png
resources
Makefile
Makefile.Web
<category>_cool_example.vcxproj
raylib.sln
README.md
examples.js
<category>_cool_example.html
<category>_cool_example.data
<category>_cool_example.wasm
<category>_cool_example.js

BCN GAME FEST

raylib automation: examples management

SOLUTION 01: Provide <examples_template.c>

RESULT: FAIL (mostly)

- Many contributors **do not read** the requirements
- Many contributors **do not follow** the conventions
- Many contributors **feel overwhelmed** by requirements
- Most example contributions required **full review**

BCN GAME FEST

raylib automation: examples management

SOLUTION 02: automate examples management

but... HOW?

BCN GAME FEST

raylib automation: examples management

IDEA: Use C and raylib

But, is it possible to build complex pipelines in C with raylib?

BCN GAME FEST

raylib automation: examples management: **rexm**

rexm

raylib examples manager

raylib automation: examples management: **rexm**

rexm commands available:

- `create <new_example_name>  : Creates empty example, using template`
- **`add`** `<example_name>          : Add existing example`
- `rename <old_examples_name> <new_example_name> : Rename an existing example`
- `remove <example_name>      : Remove an existing example`
- `build <example_name>       : Build example for Desktop and Web platforms`
- **`validate`** `                   : Validate collection, generates report`
- **`update`** `                     : Validate and update examples collection`

raylib automation: examples management: **rexm**

rexm commands: **add**

- **Copy required files** (.c, .png)
- **Edit build systems (text files)**
  - Makefile + Makefile.Web
  - VS2022 project + solution
- **Edit** examples **README.md table**
- **Build web version + Copy files**
  - **Scan example resources!**

<category>_cool_example.c
<category>_cool_example.png
resources
Makefile
Makefile.Web
<category>_cool_example.vcxproj
raylib.sln
README.md
examples.js
<category>_cool_example.html
<category>_cool_example.data
<category>_cool_example.wasm
<category>_cool_example.js

raylib automation: examples management: **rexm**

rexm commands: **validate** --> **update**

```
 - [C]     : Missing .c source file
 - [CAT]   : Not a recognized category
 - [INFO]  : Inconsistent example header info (stars, author...)
 - [PNG]   : Missing screenshot .png
 - [WPNG]  : Invalid png screenshot (using default one)
 - [RES]   : Missing resources listed in the code
 - [MK]    : Not listed in Makefile
 - [MKWEB] : Not listed in Makefile.Web
 - [VCX]   : Missing Visual Studio project file
 - [SOL]   : Project not included in solution file
 - [RDME]  : Not listed in README.md
 - [JS]    : Not listed in Web (examples.js)
 - [WOUT]  : Missing Web build (.html/.data/.wasm/.js)
 - [WMETA] : Missing Web .html example metadata
```

BCN GAME FEST

EXAMPLE NAME	[C]	[CAT]	[INFO]	[PNG]	[WPNG]	[RES]	[MK]	[MKWEB]	[VCX]	[SOL]	[RDME]	[JS]	[WOUT]	[WMETA]
core_basic_window	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
core_delta_time	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
core_input_keys	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
core_input_mouse	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
core_input_mouse_wheel	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
core_input_gamepad	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
core_input_multitouch	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
core_input_gestures	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
core_input_gestures_testbed	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
core_input_virtual_controls	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
core_2d_camera	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
core_2d_camera_mouse_zoom	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
core_2d_camera_platformer	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
core_2d_camera_split_screen	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
core_3d_camera_mode	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
core_3d_camera_free	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
core_3d_camera_first_person	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
core_3d_camera_split_screen	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

raylib automation: examples management: **rexm**

rexm commands: **build**

- Setup required environment for target platform
  - Set compiler/libs paths, platform dependant
- **Build example, multiplatform support!**
  - **Desktop: Windows, Linux, macOS (Makefile)**
  - **Web: Emscripten (Makefile.Web)**
- Copy files to destination directory
- **Testbed for future projects!**

BCN GAME FEST

raylib automation: new **raylib** functionality

File System Management

```
- LoadFileData(), UnloadFileData()
- SaveFileData(), ExportDataAsCode()
- LoadFileText(), UnloadFileText()
- FileRename(), FileRemove()
- FileCopy(), FileMove()
- FileTextReplace()
- FileTextFindIndex()
- FileExists(), DirectoryExists()
- IsFileExtension(), GetFileExtension()
- GetFileName(), GetFileNameWithoutExt()
- GetWorkingDirectory()
- MakeDirectory(), ChangeDirectory()
- LoadDirectoryFiles()...
```

Text/String Management

```
- LoadTextLines(), UnloadTextLines()
- TextCopy(), TextFormat(),
- TextSubtext()
- TextLength(), TextIsEqual()
- TextRemoveSpaces()
- GetTextBetween()
- TextReplace(), TextReplaceBetween()
- TextInsert(), TextAppend()
- TextJoin(), TextSplit()
- TextFindIndex()
- TextToUpper(), TextToLower()
- TextToPascal(), TextToSnake()
- TextToInteger(), TextToFloat()
```

595 functions available!

BCN GAME FEST

So, is it possible to build complex pipelines in C with raylib?

BCN GAME FEST

Yes.

raylib automation: pipelines building

Benefits of pipeline development in C + raylib:

- Unified tech stack, **minimize dependencies**
- **High-performance**, low memory footprint
- **Portable and multi-platform** pipeline
- Lot of **media functionality** already provided
- Compilable pipeline as **custom tool (with UI)**

raylib automation: pipelines building

Limitations of pipeline development in C + raylib:

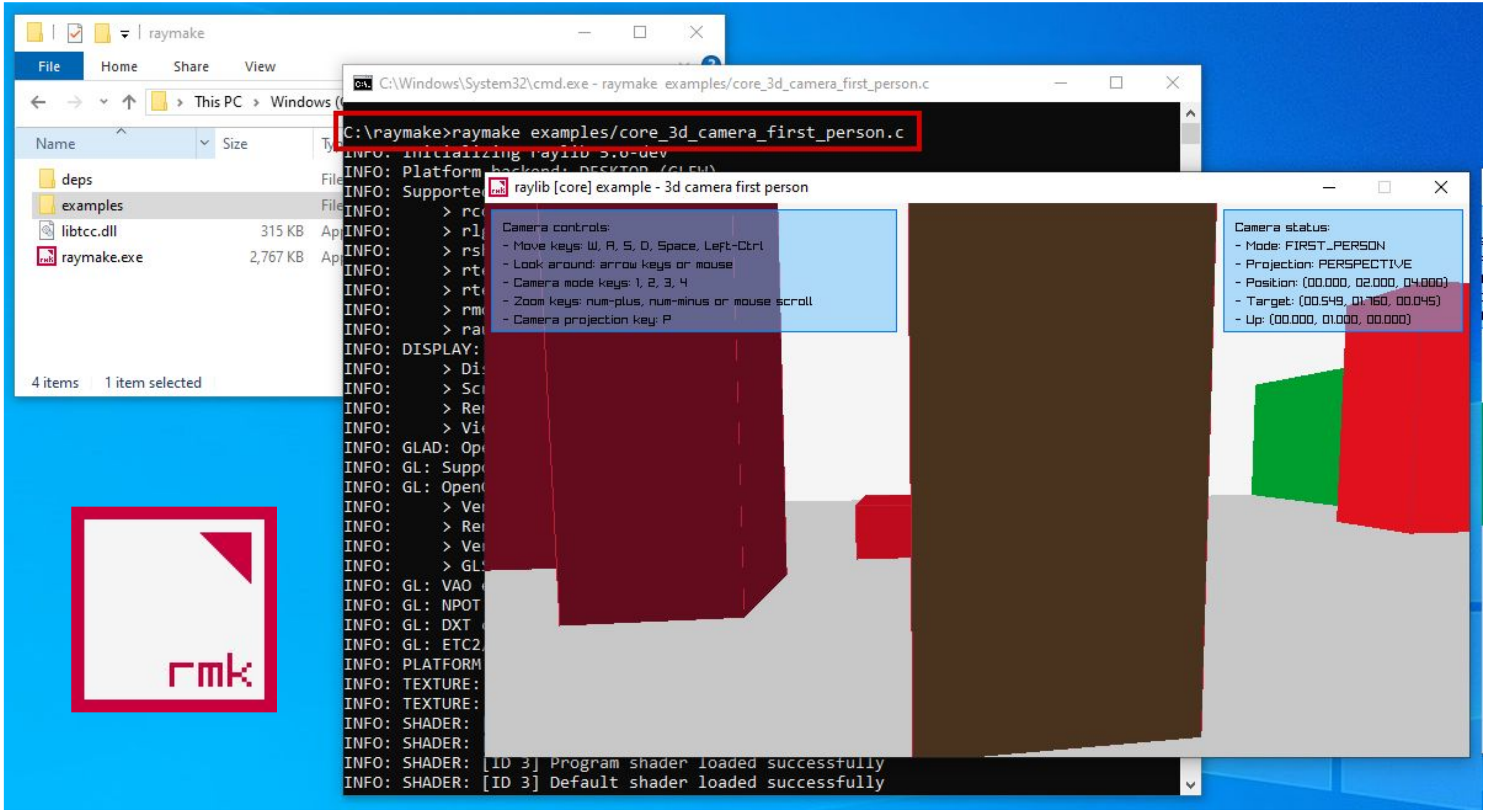
- Not as popular as scripting languages (**Python**)
- Limited packages**?** and advanced functionality
- Requires coding in C, slow, more complexity**?**
- Pipeline maintenance, more complexity**?**
- Pipeline must be compiled... **really?**

raylib automation: C code runner: **raymake**

rmk

raymake

raylib JIT C code runner



```
C:\raymake>raymake examples/core_3d_camera_first_person.c
INFO: Initializing raylib 3.0-dev
INFO: Platform backend: DESKTOP (GLFW)
INFO: Supporte
INFO:     > rc
INFO:     > rl
INFO:     > rs
INFO:     > rt
INFO:     > rt
INFO:     > rm
INFO:     > ra
INFO: DISPLAY:
INFO:     > Di
INFO:     > Sc
INFO:     > Re
INFO:     > Vi
INFO: GLAD: Op
INFO: GL: Supp
INFO: GL: Open
INFO:     > Ve
INFO:     > Re
INFO:     > Ve
INFO:     > GL
INFO: GL: VAO
INFO: GL: NPOT
INFO: GL: DXT
INFO: GL: ETC2
INFO: PLATFORM
INFO: TEXTURE:
INFO: TEXTURE:
INFO: SHADER:
INFO: SHADER:
INFO: SHADER: [ID 3] Program shader loaded successfully
INFO: SHADER: [ID 3] Default shader loaded successfully
```

Camera controls:
- Move keys: W, A, S, D, Space, Left-Ctrl
- Look around: arrow keys or mouse
- Camera mode keys: 1, 2, 3, 4
- Zoom keys: num-plus, num-minus or mouse scroll
- Camera projection key: P

Camera status:
- Mode: FIRST_PERSON
- Projection: PERSPECTIVE
- Position: (00.000, 02.000, 04.000)
- Target: (00.549, 01.760, 00.045)
- Up: (00.000, 01.000, 00.000)

raymake
File
Home
Share
View
This PC
Windows
Name
Size
deps
examples
libtcc.dll 315 KB
raymake.exe 2,767 KB
4 items 1 item selected
C:\Windows\System32\cmd.exe - raymake examples/core_3d_camera_first_person.c
raylib [core] example - 3d camera first person
rmk

Future raylib pipelines?

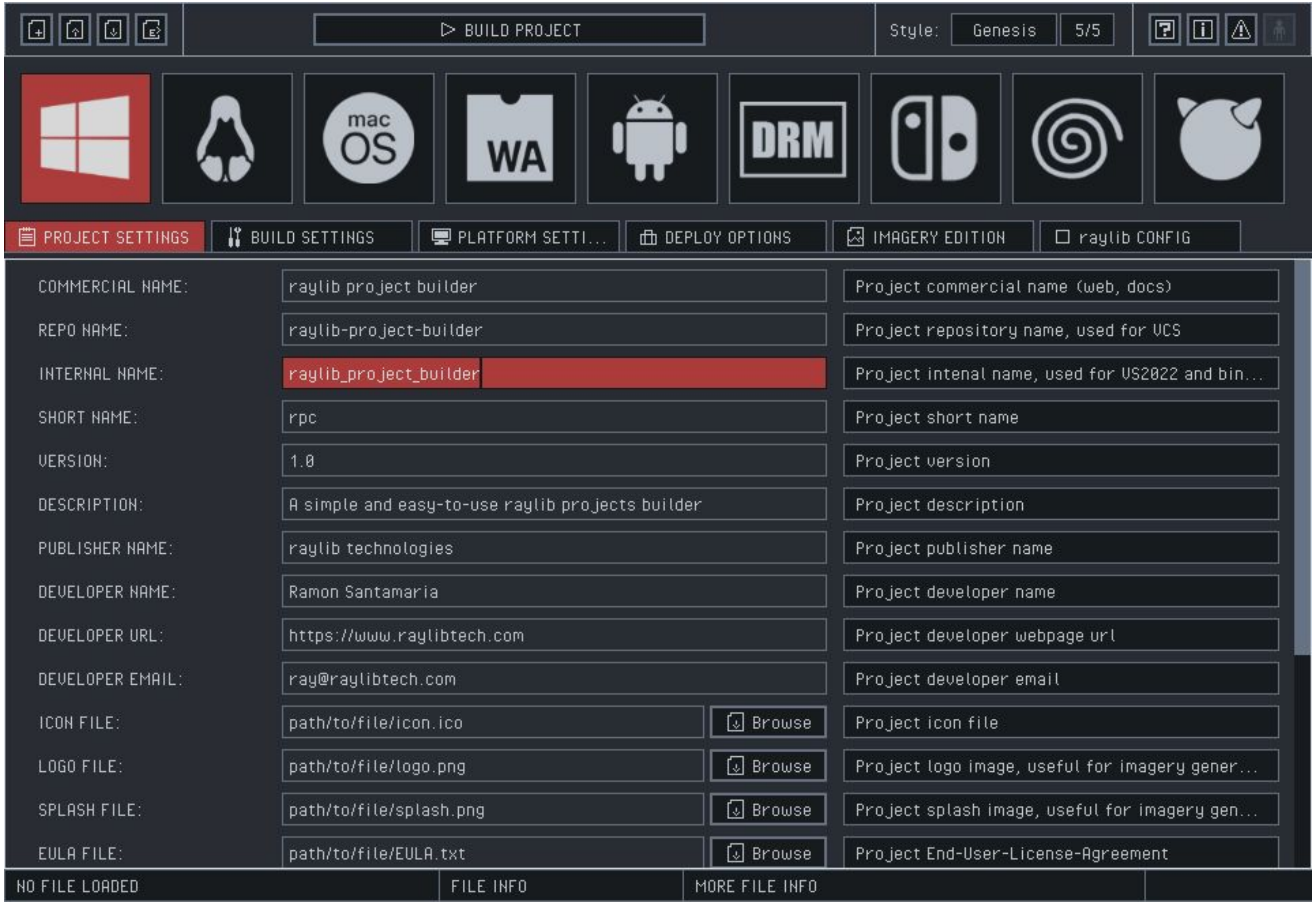
BCN
GAME FEST

Automated raylib projects **building** and **deployment**

BCN GAME FEST

raylib automation: future: **raylib-project-builder**

BCN GAME FEST



COMMERCIAL NAME:	raylib project builder	Project commercial name (web, docs)
REPO NAME:	raylib-project-builder	Project repository name, used for VCS
INTERNAL NAME:	raylib_project_builder	Project intenal name, used for VS2022 and bin...
SHORT NAME:	rpc	Project short name
VERSION:	1.0	Project version
DESCRIPTION:	A simple and easy-to-use raylib projects builder	Project description
PUBLISHER NAME:	raylib technologies	Project publisher name
DEVELOPER NAME:	Ramon Santamaria	Project developer name
DEVELOPER URL:	https://www.raylibtech.com	Project developer webpage url
DEVELOPER EMAIL:	ray@raylibtech.com	Project developer email
ICON FILE:	path/to/file/icon.ico Browse	Project icon file
LOGO FILE:	path/to/file/logo.png Browse	Project logo image, useful for imagery gener...
SPLASH FILE:	path/to/file/splash.png Browse	Project splash image, useful for imagery gen...
EULA FILE:	path/to/file/EULA.txt Browse	Project End-User-License-Agreement

NO FILE LOADED | FILE INFO | MORE FILE INFO

rpb

BCN GAME FEST

raylib

QUESTIONS?

ray@raylib.com

BCN GAME FEST