

# Spoopy Environment Planner



## Introduction

Spoopy Environment (Spoopy Envi. for short) is a surface-level, lightweight framework designed to simplify game development, similar to Kore from Kode. However, unlike Kore, it serves as a higher abstraction layer.

## Inspiration

*The structure and original inspiration, and still is the main inspiration, for this framework is HaxeFlixel. I've used HaxeFlixel for a couple of years now, publishing my first game, "Dinomite," and making a couple of mods for a certain rhythm game. I really like the feel and how simple and well-structured the framework is, and I wanted to create one that isn't limited to Haxe and also provides more low-level control.*



*When it comes to the style of code, format, and interior structure, I took heavy inspiration from Taisei, a Touhou fan game, because of how I found the project to use the best strategies when using the C language. It's shown in the source that they really know what they are doing and have a lot of experience when it comes to development. My goal is to mimic their style in a way I can comprehend it and expand and maybe even improve on it. That's why the framework's source is almost identical or, maybe by now, has hints.*

# Perk Badges

*Like a CS student getting accepted to Carnegie Mellon University, this badge signifies immense pride because it demonstrates that the task was unrealistic to complete in one week, which realistically should have taken about a month or so. If this badge is earned, just like the logo, you have every right to brag about it in a CMU essay; you deserve it! If they reject the application, they smoke a whole ounce of cocaine.*



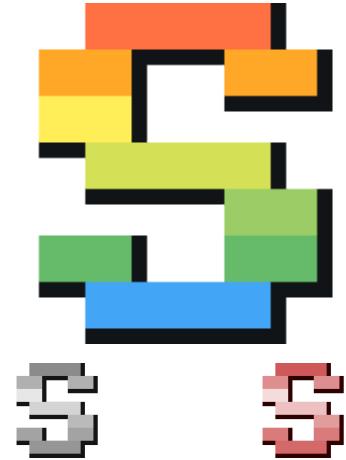
## Status of CMU Badge

- Left Version Signifies: **Pending**
- Right Version Signifies: **Failed**

*Rewriting a framework from the ground up (again) is an incredibly long and tedious process that 99% of the time takes more than a week. So! It's a huge achievement when you pull this off in a week! If this badge is earned, you have every right to "brag" about it in an essay, but probably don't straight-up brag about it, as they'll most likely reject the application.*

## Status of Sokol Badge

- Left Version Signifies: **Pending**
- Right Version Signifies: **Failed**



# Development Timeline (Beta 0.1)

**Due before HoloJam #8 (Basically one year)**

**Design alongside Video Editor + Musonite (aka. Dinomite 2.0)**

**Post every progress update starting from assignment #4**

**Each task (except Dreamcast) must be finished every week, unless a badge signifies otherwise**

**If people ACTUALLY enjoy your content, you must post a video for every task completed (shorts are qualifiable)**

#	Assignment	Type	Status	Start Date	End Date	Notes
	Add threading support	Design ▾	Completed ▾	Jul 11, ...	Jul 18,...	What you're literally doing right now with meson and C
	Debugging First Testing with Spoopy Threading	Debug ▾	Completed ▾	Jul 19, ...	Jul 19,...	Used Claude AI
	Receive Permission	N/A ▾	Completed ▾	Jul 16, ...	Jul 20,...	For Dreamcast support from Ross
	Configure Kore with Meson + Make Wrappers	Design ▾	Completed ▾	Jul 20, ...	Jul 22,...	Supports MacOS (basic spoopy_graphics.a with C)
	Set up Slang (for MacOS)	Design ▾	Completed ▾	Jul 31, ...	Aug 7,...	
	Configure Slang for a second attempt	Design ▾	Completed ▾	Aug 9, ...	Aug 1...	
	Attend the VCHC meeting on ZOOM	N/A ▾	Completed ▾	Aug 5, ...	Aug 5,...	At 11 am
	Triangle Prototype + It Runs!	All Above ▾	Completed ▾	Jul 23, ...	Sep 1...	This includes spoopy_graphics with Clang test on macOS

	Make a README, then post a short with brainrot on TikTok, Shorts		Completed ▾	Oct 12, ...	Oct 12...	Ask for help and have others review the video, to then give the green light
	Dreamcast Support	Design ▾	Post Poned ▾	Aug 25, ...	ⓘ Date	Make sure Ross approves of this (Wait til Haxe 5.0 to make this call)
	Enhance Spooky's memory using <code>mimalloc</code>	Enhance ▾	Completed ▾	Dec 8, ...	Dec 1...	Construct a span allocator as well (use it with the texture case)
	Remove Kore v2 in favor of Sokol (Needs custom NSView)	Design ▾	In Progress ▾	Dec 16, ...	ⓘ Date	Discontinue `renderer` as a static library, but instead be part of the <code>DLL</code>
	Implement all Taisei Tests	All Above ▾	In Progress ▾	Jan 16, ...	ⓘ Date	'TODO (All Tests)'
	Optimize elements of the implementation	Enhance ▾	Not Started ▾	ⓘ Date	ⓘ Date	'TODO (Base Optimize)'
	Replace the JS scripts with Rhai and make a custom Rust Project CLI (includes custom make)	Enhance ▾	Not Started ▾	ⓘ Date	ⓘ Date	The Rust Project CLI must let Meson run Rhai scripts flawlessly and name it `environment.` (Remove Makefile)
	Rewrite the main <code>meson build</code> to have it be managed by the custom Rust project	Enhance ▾	Not Started ▾	ⓘ Date	ⓘ Date	
	Design the framework layout in Rust (use iocrafts)	Design ▾	Not Started ▾	ⓘ Date	ⓘ Date	Use Lime OpenFL as the reference. + Implement from 'TODO (Framework)'
	Make a fast-paced, short, and full-length video documentary about Rust and Rhai	N/A ▾	Not Started ▾	ⓘ Date	ⓘ Date	Must be about how you decided to use Rust with Rhai when first starting Spooky Envi.

	Implement Kore v3 like Samples, not including multi-window	<span>Enhance</span> ▾	Not Started ▾	<span>Date</span>	<span>Date</span>	Alongside the “This ‘TODO (Samples)’
	Create a Tech Demo of “Itstdemo I love you” - Kasane Teto, using Raymarching	<span>Design</span> ▾	Not Started ▾	<span>Date</span>	<span>Date</span>	Must post <u>multiple</u> shorts, a full Inigo Quilez-like video on the math to complete this task
	Lower Drawcall Usage if possible	<span>Debug</span> ▾	Not Started ▾	<span>Date</span>	<span>Date</span>	
	Implement an option to disable the runtime slang shader compilation	<span>Enhance</span> ▾ <span>Debug</span> ▾	Not Started ▾	<span>Date</span>	<span>Date</span>	Cut the source files out of compilation as well. <u>However</u> , this option can be overwritten when in DEBUG
	Heavily rewrite Slang to better support profiles and remove BX to use the custom Spoopy components	<span>Enhance</span> ▾	Not Started ▾	<span>Date</span>	<span>Date</span>	‘TODO (Slang)’
	Start Developing Musonite (macOS only for now)  <b>Goal:</b> Main gameplay mirrors the original	<span>All Above</span> ▾	Not Started ▾	<span>Date</span>	<span>Date</span>	<u>Minimum 14 MB Heap Capacity</u> and 76k VRam  Allowed to continue once the mirror is complete
	Have WebGPU + Add Emscripten Support	<span>All Above</span> ▾	Not Started ▾	<span>Date</span>	<span>Date</span>	WebGPU renderer is separate from Kore2
	Make both Emscripten and WebGPU compilable on every OS	<span>Debug</span> ▾ <span>Enhance</span> ▾	Not Started ▾	<span>Date</span>	<span>Date</span>	
	Implement Windows support for Spoopy	<span>Enhance</span> ▾	Not Started ▾	<span>Date</span>	<span>Date</span>	
	<u>Test Cases</u> for tests + samples universally	<span>Enhance</span> ▾	Not Started ▾	<span>Date</span>	<span>Date</span>	except WebGPU
	Implement Test Cases for the WebGPU	<span>Enhance</span> ▾	Not Started ▾	<span>Date</span>	<span>Date</span>	

	platform					
	Add States	Design ▾	Not Started ▾	<input type="button"/> Date	<input type="button"/> Date	Like HaxeFlixel. 'TODO (States)'
	Implement a sample that uses States	Enhance ▾	Not Started ▾	<input type="button"/> Date	<input type="button"/> Date	Use HaxeFlixel tests as Ref
	Add LuaJIT scripting support (spoopy_scripts.a)	Design ▾	Not Started ▾	<input type="button"/> Date	<input type="button"/> Date	Have it be a wrapped sub-API
	Rewrite Musonite to have scripting support	All Above ▾	Not Started ▾	<input type="button"/> Date	<input type="button"/> Date	
	Have spoopy_ui.a program in Rust using Taffy	Design ▾	Not Started ▾	<input type="button"/> Date	<input type="button"/> Date	Required by the options and the death menu of Musonite
	Have spoopy_audio.a, which uses OpenAL	Design ▾	Not Started ▾	<input type="button"/> Date	<input type="button"/> Date	Must be finished with the overall renderer for Musonite and in need of audio
	Make Sound Engine for Musonite	All Above ▾	Not Started ▾	<input type="button"/> Date	<input type="button"/> Date	Minimum 14 MB Heap Capacity, 76 KB VRam + all supported platforms
	Separate the Updater to have a Render Thread and a Fixed Update Thread	Enhance ▾	Not Started ▾	<input type="button"/> Date	<input type="button"/> Date	
	Have that version script be compatible and able to warn the user if their Spoopy is outdated	Enhance ▾	Not Started ▾	<input type="button"/> Date	<input type="button"/> Date	
	Finish up Musonite!	After ▾	Not Started ▾	<input type="button"/> Date	<input type="button"/> Date	<b>0.1.0 (On Release)</b>
	Make a kick ass website for Spoopy (Ask Eddie for help)	Design ▾	Not Started ▾	<input type="button"/> Date	<input type="button"/> Date	<b>0.1.0 (On Release):</b> Host it with a custom domain
	Have the spoopy repo on GitHub be mirrored from the website	Enhance ▾	Not Started ▾	<input type="button"/> Date	<input type="button"/> Date	<b>0.1.0 (On Release)</b>

	Develop the first HoloJam Spoopy game!	<span>Design</span> ▾	<span>Not Started</span> ▾	<span>□ Date</span>	<span>□ Date</span>	<b>0.1.0 (On Release):</b> Most importantly, however, <i>have fun!</i>
	Publish framework on Homebrew, Scoop, and NPM	<span>After</span> ▾	<span>Not Started</span> ▾	<span>□ Date</span>	<span>□ Date</span>	<b>0.1.0 (On Release)</b>
	Implement universal multi-window/multi-viewport rendering support	<span>After</span> ▾	<span>Not Started</span> ▾	<span>□ Date</span>	<span>□ Date</span>	<b>0.2.0:</b> 'TODO (Multi-Window).'
	Have kmake Meson Exporter (Includes removing JSON)	<span>After</span> ▾	<span>Completed</span> ▾	Jul 30, ...	Sep 1...	<b>0.2.0:</b> Must have successfully pushed to Kore to pass
	Have Linux Support	<span>After</span> ▾	<span>Not Started</span> ▾	<span>□ Date</span>	<span>□ Date</span>	<b>0.2.0</b>
	Very Basic Editor UI template to showcase real-time shader loading (ImGui + Tweens)	<span>After</span> ▾	<span>Not Started</span> ▾	<span>□ Date</span>	<span>□ Date</span>	<b>0.2.0:</b> Basically just shaders and other basics that editors need that can be exported to .scene
	Crapton of language support	<span>After</span> ▾	<span>Not Started</span> ▾	<span>□ Date</span>	<span>□ Date</span>	<b>0.2.0:</b> Haxe, Rust, etc.
	Have a docs/contributions.md	<span>After</span> ▾	<span>Not Started</span> ▾	<span>□ Date</span>	<span>□ Date</span>	<b>0.2.0:</b> Rules, like: "log the last commit to a language binding in the BINDINGS.md."
	Design AI terrain renderer through 2D plane with 3D projection (for 4D)	<span>After</span> ▾	<span>Not Started</span> ▾	<span>□ Date</span>	<span>□ Date</span>	<b>0.2.0 Demo Project:</b> Best to do this after transferring and while in an ML class

# Development Timeline (Full Release 0.2.5)

Design alongside “Spoopy’s Content Loader“ for a custom Pokémon game

#	Assignment	Type	Status	Start Date	End Date	Notes
	Add Hscript scripting support.	Design ▾	Not Started ▾	<input type="button" value="Date"/>	<input type="button" value="Date"/>	
	Add 3DS support	All Above ▾	Not Started ▾	<input type="button" value="Date"/>	<input type="button" value="Date"/>	Needs money \$200
	Implement a High-level Marching Cubes API	Design ▾	Not Started ▾	<input type="button" value="Date"/>	<input type="button" value="Date"/>	
	Optimize the Marching Cubes API for performance and speed	Enhance ▾	Not Started ▾	<input type="button" value="Date"/>	<input type="button" value="Date"/>	Add to AngelScript's Engine Layer
	Embed Haxe into the main engine via HashLink and injectively optimize HashLink	All Above ▾	Not Started ▾	<input type="button" value="Date"/>	<input type="button" value="Date"/>	This is for making in-engine menus
	Move necessary components from the Rust CLI app of Spoopy to a dynamic library called: spoopy_launcher	Enhance ▾	Not Started ▾	<input type="button" value="Date"/>	<input type="button" value="Date"/>	Most likely will include the Rhai framework and not include the iocrafts directly, but just the design logic
	Have Haxe with HL use the spoopy_launcher dynamic library	Enhance ▾	Not Started ▾	<input type="button" value="Date"/>	<input type="button" value="Date"/>	
	Integrate a “project launcher menu,” like other big game engines like Unity, Godot, etc.	Design ▾	Not Started ▾	<input type="button" value="Date"/>	<input type="button" value="Date"/>	Mainly a UI version of the CLI. However, this does mean that we have to separate Spoopy projects into two: <b>embedded</b> and <b>dynamic</b>

	Design the high-level Application around the Ever-Changing game integrate in Spoopy	<span style="background-color: #c8f7e4; border-radius: 10px; padding: 2px 10px;">Design ▾</span>	<span style="background-color: #d9e1f2; border-radius: 10px; padding: 2px 10px;">Not Started ▾</span>	<span style="border: 1px solid #ccc; border-radius: 10px; padding: 2px 10px;">Date</span>	<span style="border: 1px solid #ccc; border-radius: 10px; padding: 2px 10px;">Date</span>	This begins the official code structure of Ever-Changing (Basically no 2D to keep the focus)
	Have Embedded Projects be like how Musonite was made	<span style="background-color: #e1c8ff; border-radius: 10px; padding: 2px 10px;">All Above ▾</span>	<span style="background-color: #d9e1f2; border-radius: 10px; padding: 2px 10px;">Not Started ▾</span>	<span style="border: 1px solid #ccc; border-radius: 10px; padding: 2px 10px;">Date</span>	<span style="border: 1px solid #ccc; border-radius: 10px; padding: 2px 10px;">Date</span>	Optional Haxe support via the language support of 0.2.0, but mainly C
	Dynamic Project will have at least one window that is used to display what the user wrote in their scripts	<span style="background-color: #e1c8ff; border-radius: 10px; padding: 2px 10px;">All Above ▾</span>	<span style="background-color: #d9e1f2; border-radius: 10px; padding: 2px 10px;">Not Started ▾</span>	<span style="border: 1px solid #ccc; border-radius: 10px; padding: 2px 10px;">Date</span>	<span style="border: 1px solid #ccc; border-radius: 10px; padding: 2px 10px;">Date</span>	Must not have any editor UI; this is just a blank window when nothing is written (Can be the same window as the launcher)
	By this point your confused about the Dynamic Project design: Recall how you want to design Ever Changing	<span style="background-color: #e1c8ff; border-radius: 10px; padding: 2px 10px;">All Above ▾</span>	<span style="background-color: #d9e1f2; border-radius: 10px; padding: 2px 10px;">Not Started ▾</span>	<span style="border: 1px solid #ccc; border-radius: 10px; padding: 2px 10px;">Date</span>	<span style="border: 1px solid #ccc; border-radius: 10px; padding: 2px 10px;">Date</span>	Don't forget that dynamic projects are basically how modern FNF handles mods and do need a <u>manifest.json</u>
	Have a separate tab, inspired by Roblox, that is just a catalog for imported dynamic projects	<span style="background-color: #ffccbc; border-radius: 10px; padding: 2px 10px;">Enhance ▾</span> <span style="background-color: #ffd700; border-radius: 10px; padding: 2px 10px;">Optional ▾</span>	<span style="background-color: #d9e1f2; border-radius: 10px; padding: 2px 10px;">Not Started ▾</span>	<span style="border: 1px solid #ccc; border-radius: 10px; padding: 2px 10px;">Date</span>	<span style="border: 1px solid #ccc; border-radius: 10px; padding: 2px 10px;">Date</span>	
	Full Ass Model and Animation Support	<span style="background-color: #c8f7e4; border-radius: 10px; padding: 2px 10px;">Design ▾</span>	<span style="background-color: #d9e1f2; border-radius: 10px; padding: 2px 10px;">Not Started ▾</span>	<span style="border: 1px solid #ccc; border-radius: 10px; padding: 2px 10px;">Date</span>	<span style="border: 1px solid #ccc; border-radius: 10px; padding: 2px 10px;">Date</span>	
	Optimize Full Ass Model and Animation Support	<span style="background-color: #ffccbc; border-radius: 10px; padding: 2px 10px;">Enhance ▾</span>	<span style="background-color: #d9e1f2; border-radius: 10px; padding: 2px 10px;">Not Started ▾</span>	<span style="border: 1px solid #ccc; border-radius: 10px; padding: 2px 10px;">Date</span>	<span style="border: 1px solid #ccc; border-radius: 10px; padding: 2px 10px;">Date</span>	
	Design built-in global Illumination as a "package."	<span style="background-color: #c8f7e4; border-radius: 10px; padding: 2px 10px;">Design ▾</span>	<span style="background-color: #d9e1f2; border-radius: 10px; padding: 2px 10px;">Not Started ▾</span>	<span style="border: 1px solid #ccc; border-radius: 10px; padding: 2px 10px;">Date</span>	<span style="border: 1px solid #ccc; border-radius: 10px; padding: 2px 10px;">Date</span>	Add to Hscript's Layer
	Optimize built-in global Illumination	<span style="background-color: #ffccbc; border-radius: 10px; padding: 2px 10px;">Enhance ▾</span>	<span style="background-color: #d9e1f2; border-radius: 10px; padding: 2px 10px;">Not Started ▾</span>	<span style="border: 1px solid #ccc; border-radius: 10px; padding: 2px 10px;">Date</span>	<span style="border: 1px solid #ccc; border-radius: 10px; padding: 2px 10px;">Date</span>	
	For Dynamic Projects, create an	<span style="background-color: #ffccbc; border-radius: 10px; padding: 2px 10px;">After ▾</span>	<span style="background-color: #d9e1f2; border-radius: 10px; padding: 2px 10px;">Not Started ▾</span>	<span style="border: 1px solid #ccc; border-radius: 10px; padding: 2px 10px;">Date</span>	<span style="border: 1px solid #ccc; border-radius: 10px; padding: 2px 10px;">Date</span>	<b>0.2.6:</b> Difference is that this is optional if

	export button for games					you want your dynamic project not be a mod of Spoopy, essentially, but exported as a standalone game
--	-------------------------	--	--	--	--	--

# Development Timeline (Full Release 1.0-nightly)

#	Assignment	Type	Status	Start Date	End Date	Notes
	Implement Raytraced Audio	Design ▾	Not Started ▾	<input type="button"/> Date	<input type="button"/> Date	
	Have Desmos 3D UI for Editor that loads bones and meshes (Programmed in Haxe)	Design ▾	Not Started ▾	<input type="button"/> Date	<input type="button"/> Date	Have “section models” that are folders of folders to clean up the UI (aka. Models with bones)
	If there is an audience, have plugin support for Desmos-like a UI for other engines because of Haxe	Design ▾ Optional ▾	Not Started ▾	<input type="button"/> Date	<input type="button"/> Date	This includes Godot, Unreal, Heaps, and Unity for your “fans” to use
	Design a 4D renderer	All Above ▾	Not Started ▾	<input type="button"/> Date	<input type="button"/> Date	
	Finish the First Full Chapter of Ever Changing and release it on Steam	All Above ▾	Not Started ▾	<input type="button"/> Date	<input type="button"/> Date	