

A 2D Game Engine and Integrated Editor Proposal

“It’s like Gamemaker, but with Lua”

1 Introduction

This project aims to be simple, to welcome beginners, to let experienced developers prototype a game in minutes, and to be a free starting point for anyone. Complexity will range somewhere between RPGMaker and early versions of Gamemaker, with non-essential complexities allowing for the creation of more advanced games.

2 Features

Lua as a scripting language – One of the easiest languages to learn, familiar to game modders since it’s a common embedded scripting language for games.

A minimal editor with only a few concepts to keep track of – Clutter can overwhelm beginners to the point of quitting, but if they get past that, it’ll always be a persistent mental tax.

One-file no-installer desktop game distribution – Keeps things easy for everyone. Small games shouldn’t need an installer, but they also deserve more than to be a flash game.

A simple library – `create_instance`, `draw_sprite`, `play_sound`...

3 Not-Features

Drag-and-drop – Famous for being advised against in every circumstance.

Component-System – Great for clean code and decoupling in larger systems, but unnecessary for a system this minimal; different categories of material to process can be separated by defining functions. The simple API standalone will act as well as any bridge between components.

4 User System

When you start the game, you start in a room. This room has objects in it which exist and update until the room changes or the game closes. **This is the entire system.** Within every object’s code, you can draw sprites and play sounds. An object can be visual and have a place on the screen, or they can be omniscient; executing code forever without any spatial presence. Objects can have togglable persistence, meaning they can either exist only while a room exists or exist forever until it is destroyed.

Here’s a list of all assets

Objects	The functionality and state of <i>something</i> . (A light bulb is either on or off. This can change.)
Sprites	Appearance applied to – or directly drawn by – objects. (Light bulbs are kinda round?)
Sounds	Small one-shots (like the click of a switch!) or looping audio (background music).
Backgrounds	Applied to rooms to fill the blank spaces. (A room’s wallpaper)
Rooms	A finite 2D space where objects can exist. (Your front room. There’s a light in the ceiling.)
Tile sets	A set of image segments that are used to build level layouts within rooms.
*Timelines	A list of functions executed at given times in a given order. (Walk to switch... then press on.)
*Shaders	A program that changes the game’s appearance. (This light is brightening the whole room!)

**optional, advanced functions.*

At least one room **must** exist.

The application is configurable with a short form where you can define the game’s name, resolution, FPS, the window title, and so on.