

Сравнение 2D-движков для соло-разработки игр

В этом документе представлено сравнение популярных 2D-игровых движков, а также собственной реализации движка на Python с использованием Pygame.

Сравнительная таблица

Движок	Платформа	Язык	Плюсы	Минусы
Godot	ПК, Web, моб., консоли	GScript, C#	Лёгкий, быстрый старт; настоящий 2D-рендеринг; встроенный редактор; open-source	Меньше ресурсов для 3D; меньше ассетов по сравнению с Unity
Unity (2D)	Все	C#	Профессиональный инструмент; поддержка рекламы и аналитики; Asset Store	Тяжелее для входа; коммерческая модель; большая сложность
Defold	ПК, Web, моб.	Lua	Очень лёгкий; HTML5 экспорт; хорошая модульность	Не очень гибкий; меньше туториалов и ассетов
Love2D	ПК, Web, моб.	Lua	Максимальная простота; полный контроль; идеален для джемов	Нет редактора; всё вручную
MonoGame	ПК, моб., консоли	C#	Близко к железу; гибкость; подходит для ретро	Большой объём кода; нет редактора

GDevelop	ПК, Web, моб.	Визуальный / JS	Без кода; мгновенный результат; отлично для новичков	Ограниченная гибкость; только для простых игр
Собственный движок (Python + Pygame)	ПК	Python	Полный контроль; модульность; учебная база; простой дебаг	Медленная производительность; нет редактора; сложно экспортировать; нет ассетов

Вывод

Собственный движок не конкурирует с профессиональными платформами по мощности и кроссплатформенности, но выигрывает в гибкости, простоте, контроле и ценности как обучающего ресурса. Он идеально подойдёт для экспериментов, прототипов и как база для создания собственного инструментария.