



The open source game engine Bevy

Niklas Eicker

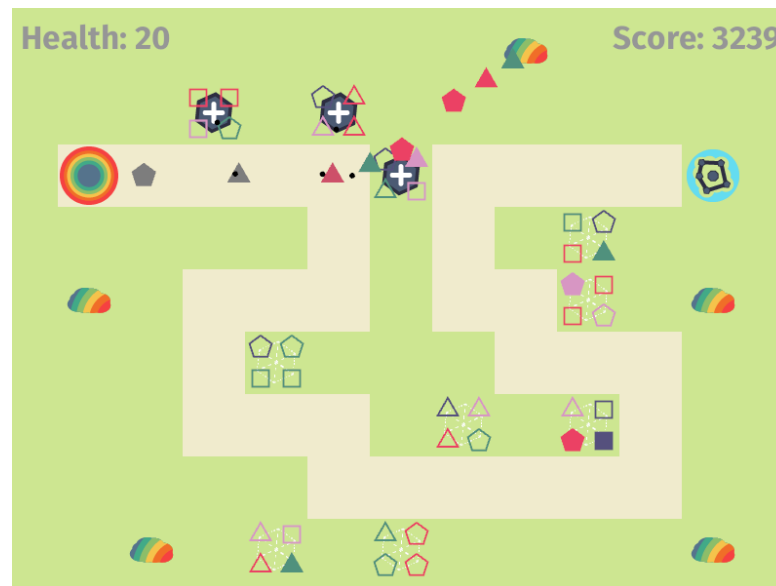
05.07.2024



About me

- senior consultant at TNG
- using Bevy since 2020
- maintaining several open source plugins
- occasional Bevy contributor

[@NiklasEi on GitHub](#)



My first game jam entry using Bevy. Oicana is a mixture of puzzle game and tower defence.

“A refreshingly simple data-driven
game engine built in Rust Free and
Open Source Forever!”

— bevyengine.org



Bevy Intro

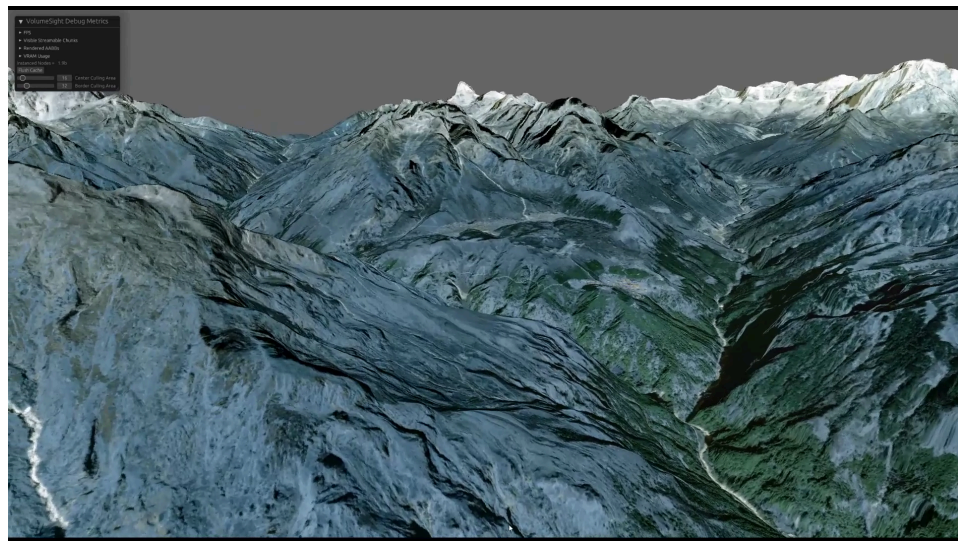
- dual-licensed under Apache 2.0 / MIT
- cross-platform: Windows, macOS, Linux, Web, Android, iOS
- rather new; first public version in August 2020

bevyengine.org



Projects using Bevy

- hundreds of small games on itch.io
- some usage outside of game dev like CAD and other simulations
- a couple of projects in mobile stores and on steam



32km² of Mount Everest in 1m resolution and automatic LOD.
Shared by @borgerking from Foresight on the Bevy Discord
(thread).



Bevy Games on Steam - Tunnet



Bevy Games on Steam - Times of progress



Community overview

- official Discord server: discord.gg/bevy
- development on GitHub: github.com/bevyengine
- resources and assets: bevyengine.org/assets



A short organisational history

August 2020



First public version

Version 0.1 is published with a blog post and a community starts to build up.

January 2022



Release of Bevy 0.6

From now on: start release process every 3 months

January 2023



Subject Matter Experts (SMEs)

New role to improve bottleneck in reviews



March 2024



Bevy foundation

A non-profit organisation; the board are the maintainers.
Move Sponsorship from individual contributors to the foundation.

May 2024



Alice hired by the foundation

Alice, a long term maintainer, is hired by the foundation to work on Bevy full-time.

June 2024



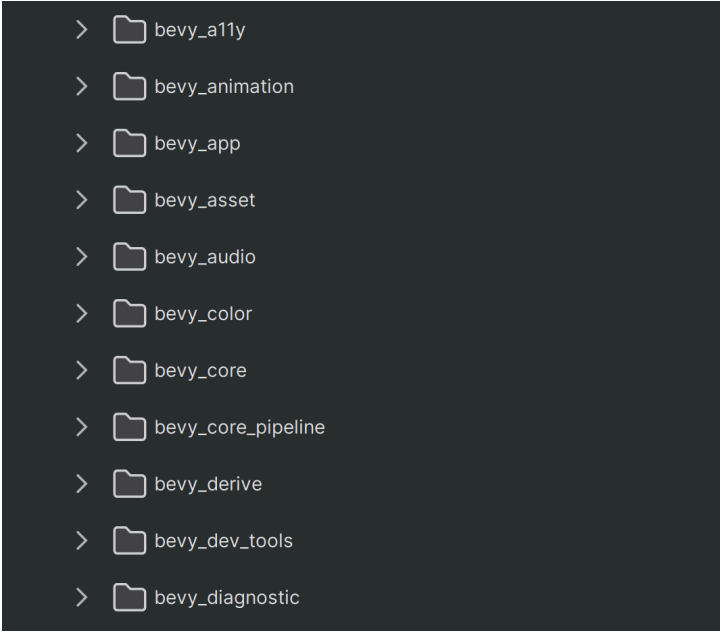
Version 0.14.0

... more to come 😊



What makes Bevy different?

- everything based on Entity Component System (ECS)
- very modular (40 “internal plugins”)
- written in Rust



```
> bevy_a11y
> bevy_animation
> bevy_app
> bevy_asset
> bevy_audio
> bevy_color
> bevy_core
> bevy_core_pipeline
> bevy_derive
> bevy_dev_tools
> bevy_diagnostic
```

Some of the internal Plugins that make up Bevy.



ECS

Separation of data and behaviour

- **Entity:** id (think 1, 2, 3...)
- **Component:** Data (like position, health points, ...)
- **Systems:** Behaviour (like “update hp based on position”)

Components get assigned to an entity. Systems can query and mutate components.



Bevy ECS

```
fn main() {  
    let mut world = World::new();  
    world.spawn(Counter { value: 0 });  
  
    let mut schedule = Schedule::default();  
    schedule.add_systems(increase_counter);  
    schedule.run(&mut world);  
}  
  
#[derive(Component)]  
struct Counter {  
    pub value: i32,  
}  
  
fn increase_counter(mut counter: Query<&mut Counter>) {  
    counter.single_mut().value += 1;  
}
```



Bevy Examples

Rust/Cargo supports examples in libraries

It's time for some code



Upcoming features

- Better UI
- Editor



Where to go from here

- Join the Discord
- Play around with Bevy examples
- Cheatbook
- Bevy game jam

Thank you!

Any questions?



Test