

The State of Game Development in Rust

About Me

- Hi, I'm Lucio Franco
- Junior Computer Science Student at the University of Denver
- I'm a member of the Amethyst Game Engine team
- I'm an enthusiast and have a passion for game engines and distributed systems

Why would you want
to use Rust for Game
Development?

Why would you want to use Rust?

- It's safe but with flexibility
- Decreased development time
- Easy to build abstractions for backends, think graphics
- No garbage collector
- Easy cross platform development
- Cargo

Why would you not want to use Rust?

- Ecosystem
- Lack of good IDE (RustyCode with VSCode?)
- Lack of REPL (rusti?)
- Big learning curve for other developers

What is available?

piston - <http://piston.rs>

- Group of game engine libraries
- Not one solution and Modular
- Designed for 2D, 3D and interactive applications

Components:

- Piston
- Conrod
- Dyon
- And more....

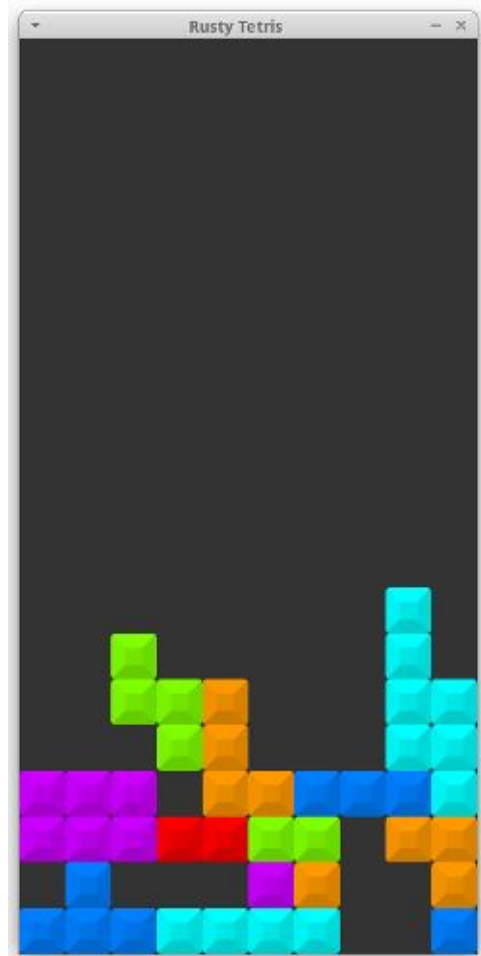
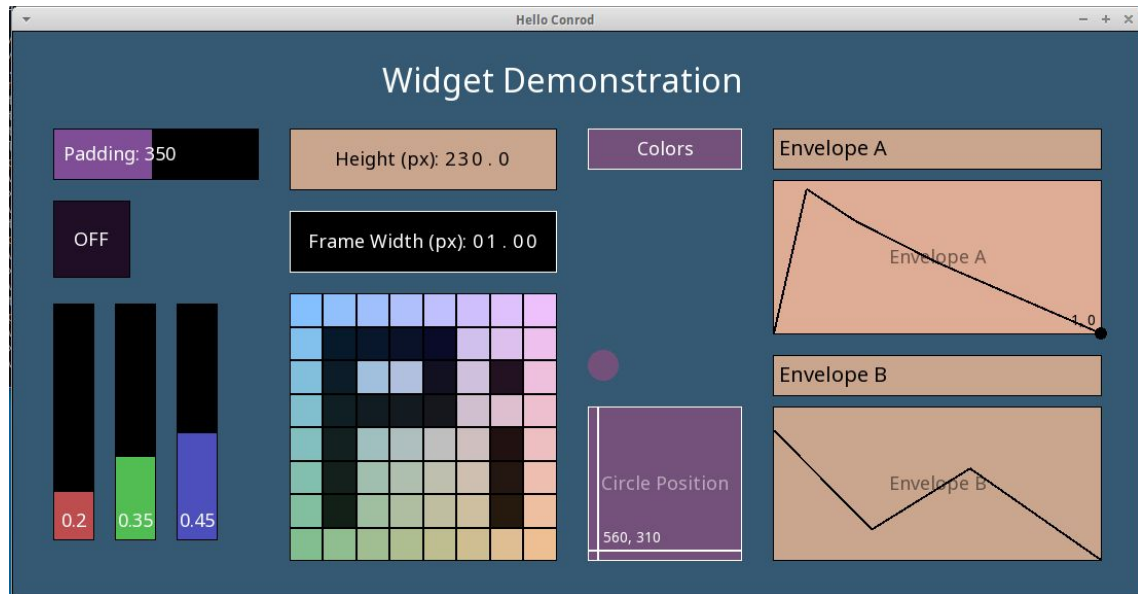
piston- <http://piston.rs>

```
use piston_window::*;

fn main() {
    let mut window: PistonWindow =
        WindowSettings::new("Hello Piston!", [640, 480]).exit_on_esc(true).build().unwrap();

    while let Some(e) = window.next() {
        window.draw_2d(&e, |c, g| {
            clear([1.0; 4], g);
            rectangle([1.0, 0.0, 0.0, 1.0], // red
                [0.0, 0.0, 100.0, 100.0],
                c.transform, g);
        });
    }
}
```


piston- <http://piston.rs>



piston- <http://piston.rs>

```
fib(x) = if x <= 0 { 0 }  
        else if x == 1 { 1 }  
        else { fib(x-1) + fib(x-2) }
```

```
fn main() {  
    for i 20 { println(fib(i)) }  
}
```



Amethyst - <https://amethyst.rs>

- YAML based data oriented game engine
- Fast Parallel Architecture
- Combines specs-rs and gfx-rs
- Modular game engine with its own package manager
- Designed to also use Piston components
- Still in early stages of development



Amethyst - <https://amethyst.rs>

```
display_config:  
  dimensions: null  
  fullscreen: false  
  max_dimensions: null  
  min_dimensions: null  
  multisampling: 1  
  title: "Windowing example"  
  visibility: true  
  vsync: true  
  backend: OpenGL
```



Amethyst - <https://amethyst.rs>

```
#[derive(Debug)]
```

```
pub struct Position {
```

```
    pub x: f32,
```

```
    pub y: f32,
```

```
    pub z: f32,
```

```
}
```

```
impl Component for Position {
```

```
    type Storage = VecStorage<Position>;
```

```
}
```



Amethyst - <https://amethyst.rs>

```
pub struct Transform;
```

```
impl Processor<Arc<Mutex<Context>>> for Transform {  
    fn run(&mut self, arg: RunArg, _: Arc<Mutex<Context>>) {  
        let (mut p, s) =  
arg.fetch(|w| (w.write::(), w.read::()));  
  
        for (p, s) in (&mut p, &s).iter() {  
            p.x += s.0;  
            p.y += s.0;  
            p.z += s.0;  
        }  
    }  
}
```



Amethyst - <https://amethyst.rs>

Benchmark	ecs	specs	recs	trex
pos_vel build	2,052,168 ns/iter (+/- 70,073)	498,342 ns/iter (+/- 400,352)	19,816,170 ns/iter (+/- 3,132,394)	1,052,629 ns/iter (+/- 271,026)
pos_vel update	439,598 ns/iter (+/- 20,638)	81,890 ns/iter (+/- 28,103)	7,408,965 ns/iter (+/- 1,603,295)	238,207 ns/iter (+/- 5,423)
parallel build	2,005,786 ns/iter (+/- 497,210)	586,433 ns/iter (+/- 168,621)	23,465,808 ns/iter (+/- 2,086,760)	3,173,121 ns/iter (+/- 1,546,373)
parallel update	9,711,756 ns/iter (+/- 4,634,118)	104,090 ns/iter (+/- 40,966)	14,161,801 ns/iter (+/- 3,291,843)	500,353 ns/iter (+/- 16,157)



Amethyst - <https://amethyst.rs>

```
use amethyst::processors::Render;
```

```
let mut game = Application::build(MainScene, config)  
    .with(Transform, "Transform", 10)  
    .with(Render, "Render", 1000)  
    .done();
```




Anima - <http://anima-engine.org>

- Very early stage
- Larger monolithic engine
- Uses the power of mrusty
- Custom rendering backend with Vulkan support
- Possible editor



Anima - <http://anima-engine.org>

```
mrusty_class!(Cont, "Container", {  
  def!("initialize", |v: i32| {  
    Cont { value: v }  
  });  
  
  def!("value", |mruby, slf: (&Cont)| {  
    mruby.fixnum(slf.value)  
  });  
});
```

```
mruby.def_file::<Cont>("cont");
```

```
let result = mruby.run("  
  require 'cont'
```

```
  Container.new(3).value  
").unwrap();
```

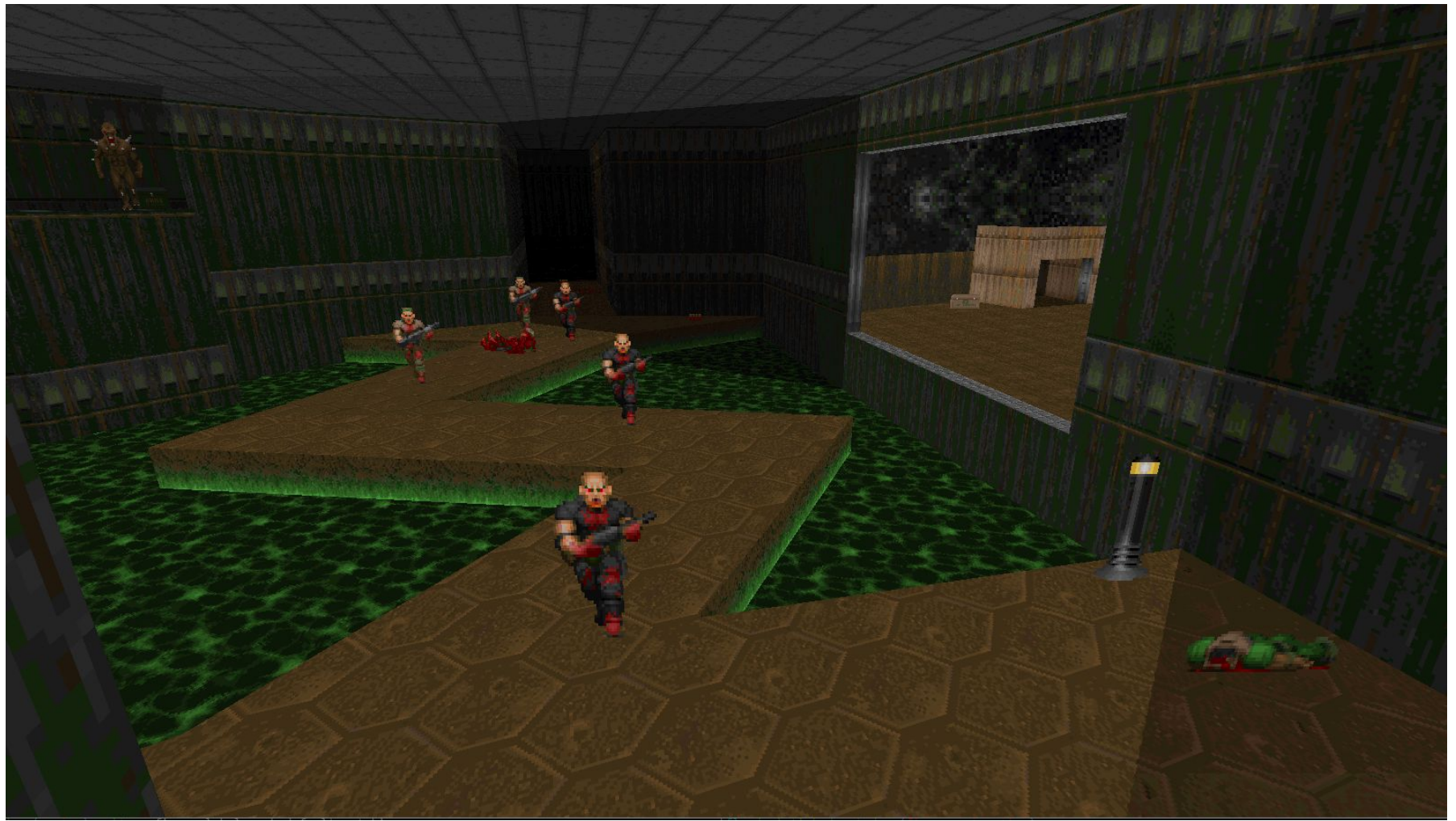
Don't want to use a
game engine?

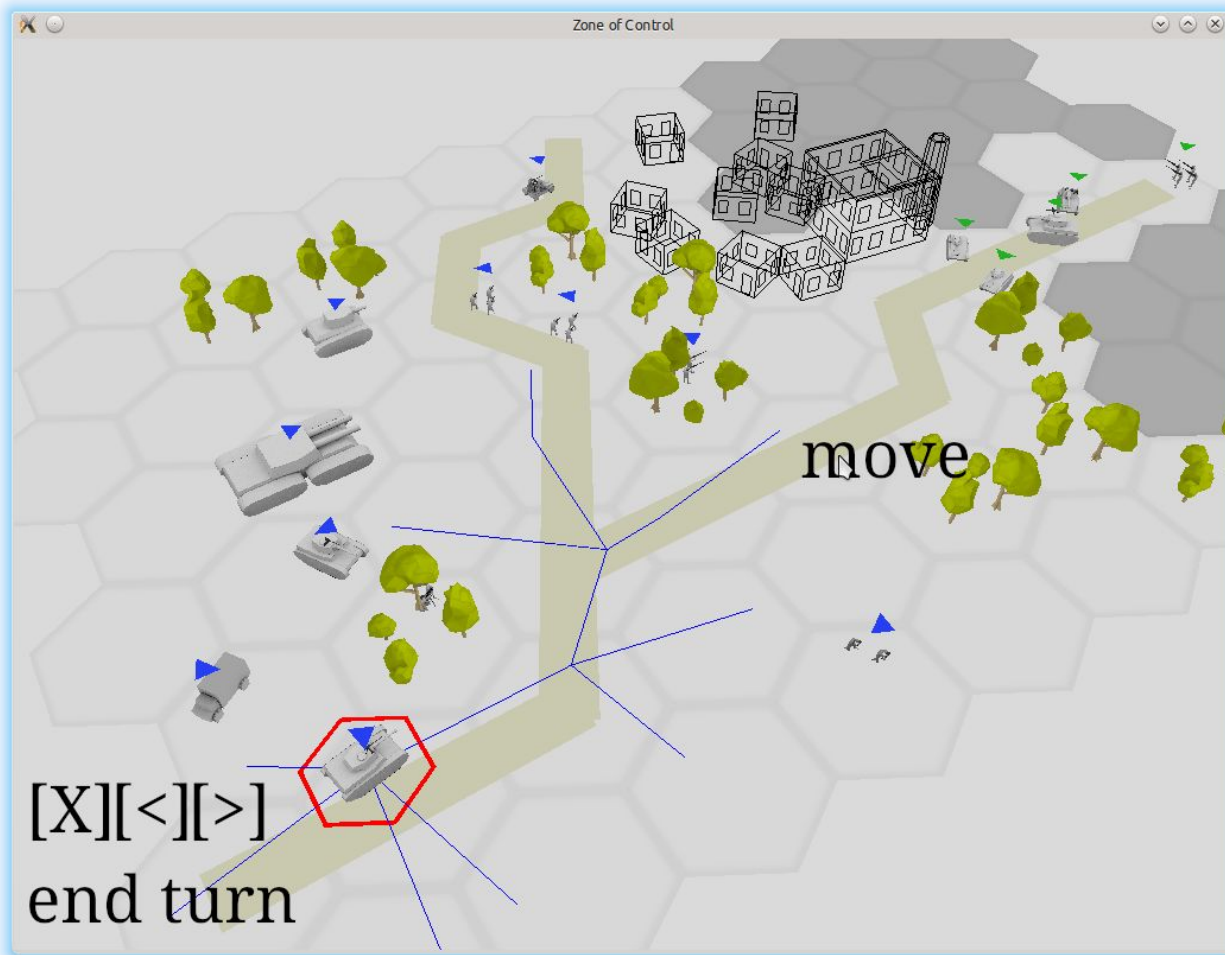
Don't want to use an engine?

- Glium - Safe OpenGL bindings
- Gfx-rs - Cross-platform multi-backend graphics library
- Vulkano - Safe Vulkan bindings
- Metal-rs - Safe Metal bindings
- Glutin - Alternative to GLFW in pure Rust
- Specs - Parallel Entity Component System
- OpenVR - Rust OpenVR bindings
- Gilrs - Cross-platform gamepad interfaces
- And more....

What are the benefits?

- Safe api's built around non-safe api's
- Easier to use
- Compatibility
- Safe but fast





Community

- [irc.mozilla.org #rust-gamedev](https://irc.mozilla.org/#rust-gamedev)
- [/r/rust_gamedev](https://www.reddit.com/r/rust_gamedev/)
- [/r/rust_gamedev/wiki/index](https://www.reddit.com/r/rust_gamedev/wiki/index)

github.com/LucioFranco/TSOGDIR