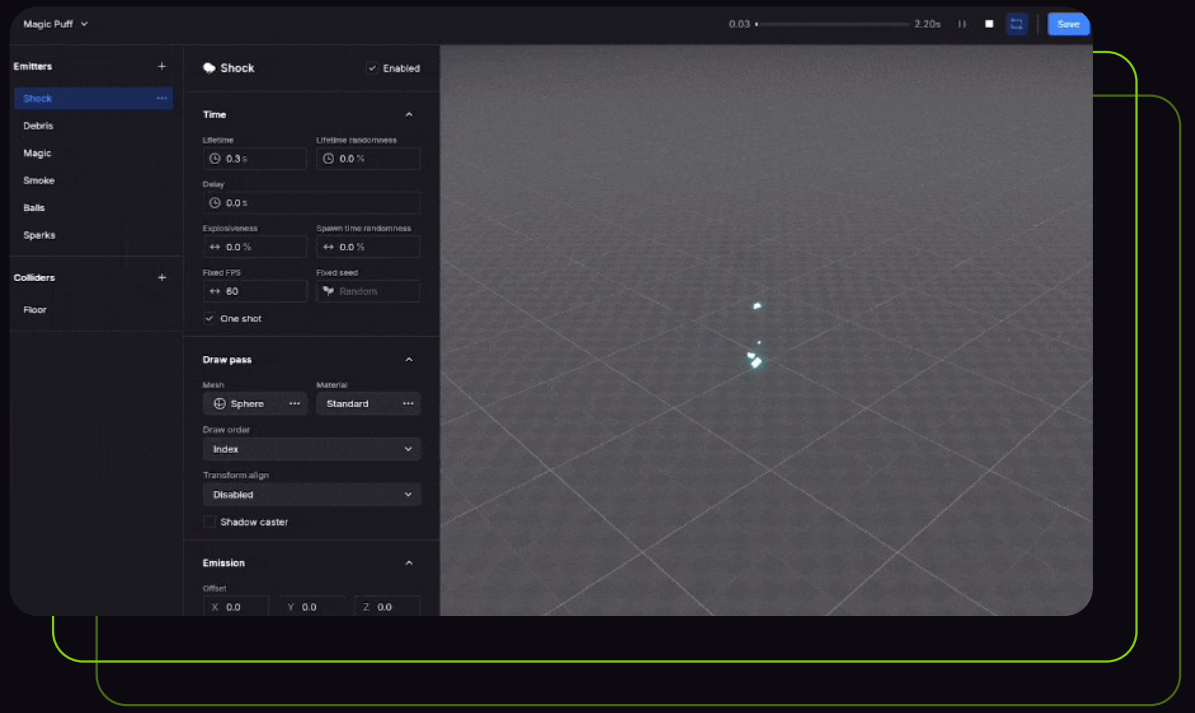
Making a VFX editor with bevy_ui



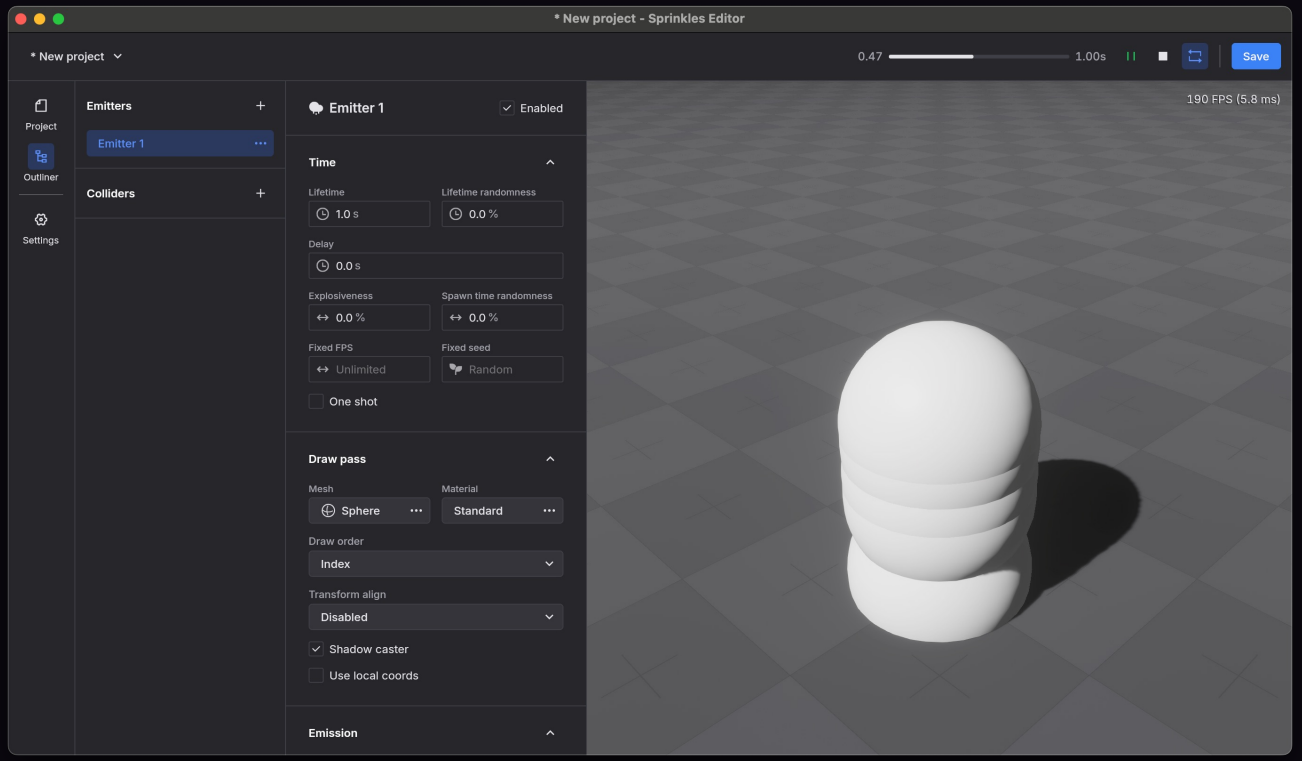
I’m Doce Fernandes

(he/they)

- Software Engineer @ Rail Europe
- Open-source maintainer
- I like making games
- Also a DJ sometimes

PART 0

Sprinkles


* New project - Sprinkles Editor
* New project
0.47
1.00s
Save
Project
Outliner
Settings
Emitters
Emitter 1
Colliders
Emitter 1
Enabled
Time
Lifetime
1.0 s
Lifetime randomness
0.0 %
Delay
0.0 s
Explosiveness
0.0 %
Spawn time randomness
0.0 %
Fixed FPS
Unlimited
Fixed seed
Random
One shot
Draw pass
Mesh
Sphere
Material
Standard
Draw order
Index
Transform align
Disabled
Shadow caster
Use local coords
Emission
190 FPS (5.8 ms)

```rust
fn setup(mut cmd: Commands, assets: Res<AssetServer>) {
    cmd.spawn(ParticleSystem3D {
        handle: assets.load("explosion.ron"),
    });
}
```

```rust
fn setup(mut cmd: Commands, assets: Res<AssetServer>) {
    let handle = assets.add(ParticleSystemAsset::new(
        "My Effect".into(),
        ParticleSystemDimension::D3,
        ..default(),
        vec![EmitterData { /* ... */ }],
        vec![],
        false,
        ..default(),
    ));

    cmd.spawn(ParticleSystem3D {
        handle,
    });
}
```

```rust
fn setup(mut cmd: Commands, assets: Res<AssetServer>) {
    let handle = assets.add(ParticleSystemAsset::new(
        "My Effect".into(),
        ParticleSystemDimension::D3,
        ..default(),
        vec![EmitterData { /* ... */ }],
        vec![],
        false,
        ..default(),
    ));

    cmd.spawn(ParticleSystem3D {
        handle,
    });
}
```

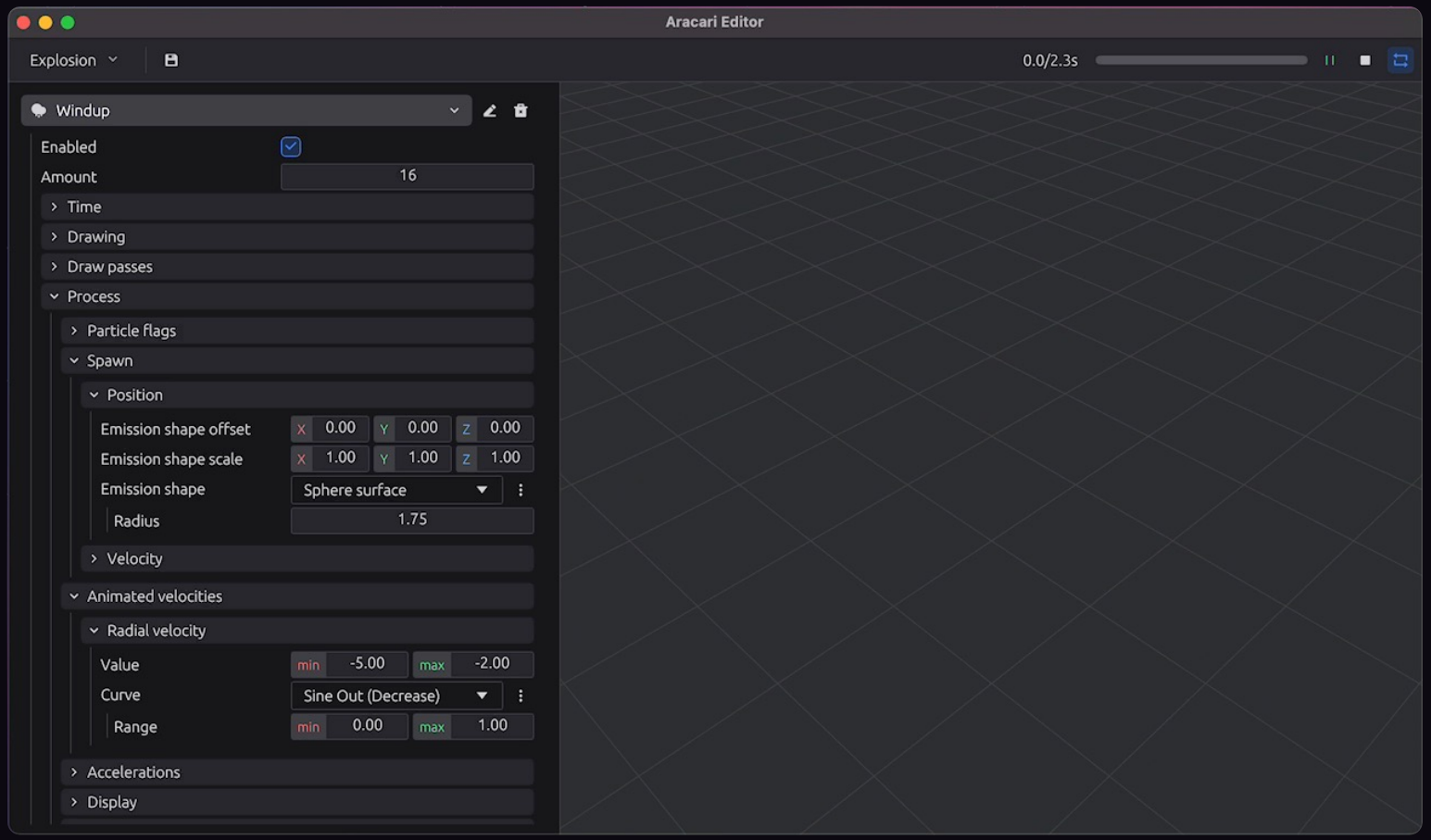
```
EmitterData {
    emission: EmitterEmission {
        particles_amount: 32,
        ..default()
    },
    velocities: EmitterVelocities {
        initial_velocity: ParticleRange::new(1.0, 5.0),
        spread: 90.0,
        ..default()
    },
    ..default()
}
```

What does the editor need?

- ✦ Live preview
- ✦ Playback controls
- ✦ Expose parameters in an easy-enough-to-use inspector
- ✦ Handle complex fields like ranges, gradients, curves, enums...
- ✦ Save & load particle systems as files

PART 1

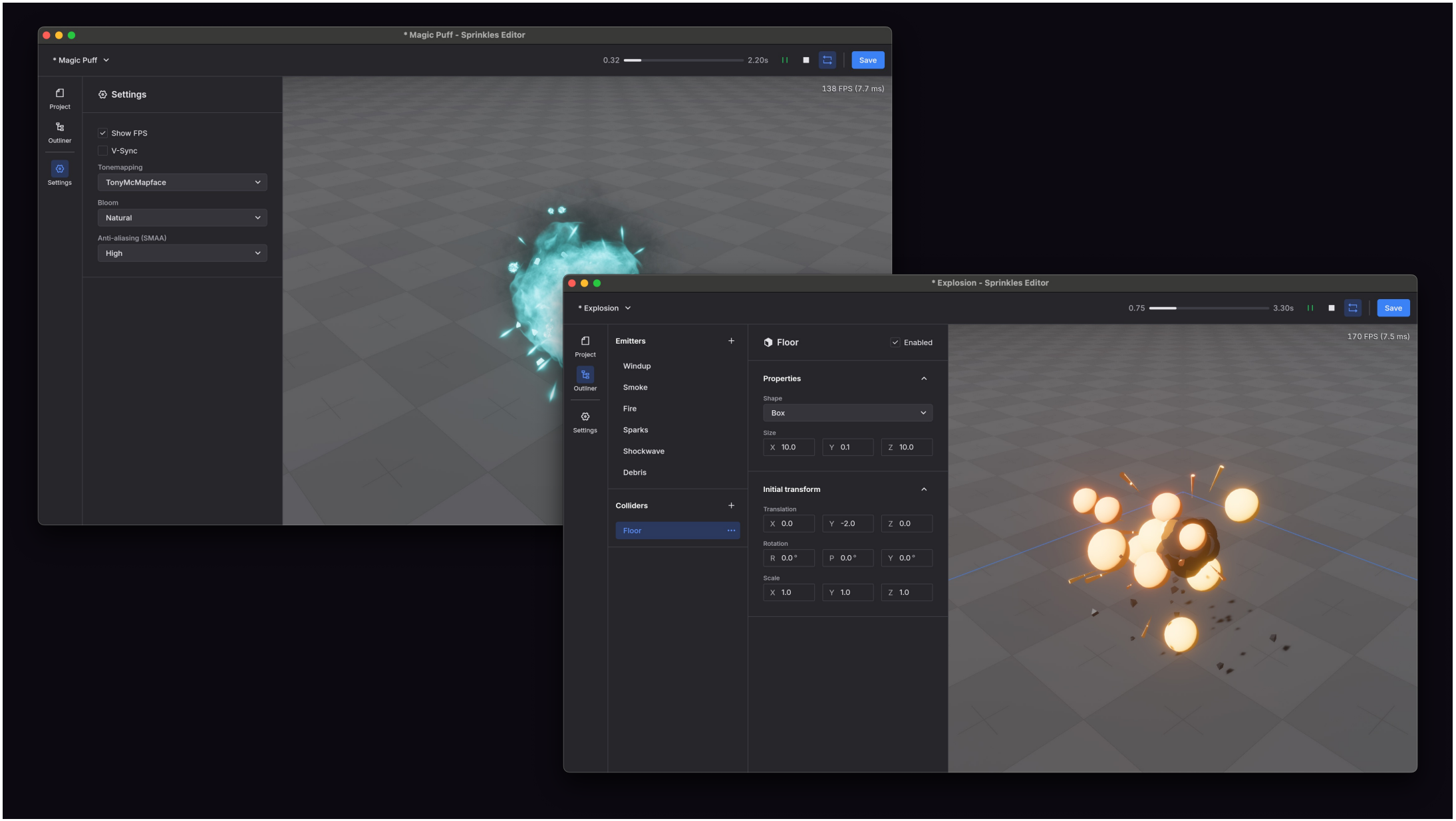
egui


Aracari Editor
Explosion
0.0/2.3s
Windup
Enabled
Amount
16
Time
Drawing
Draw passes
Process
Particle flags
Spawn
Position
Emission shape offset
X 0.00
Y 0.00
Z 0.00
Emission shape scale
X 1.00
Y 1.00
Z 1.00
Emission shape
Sphere surface
Radius
1.75
Velocity
Animated velocities
Radial velocity
Value
min -5.00
max -2.00
Curve
Sine Out (Decrease)
Range
min 0.00
max 1.00
Accelerations
Display

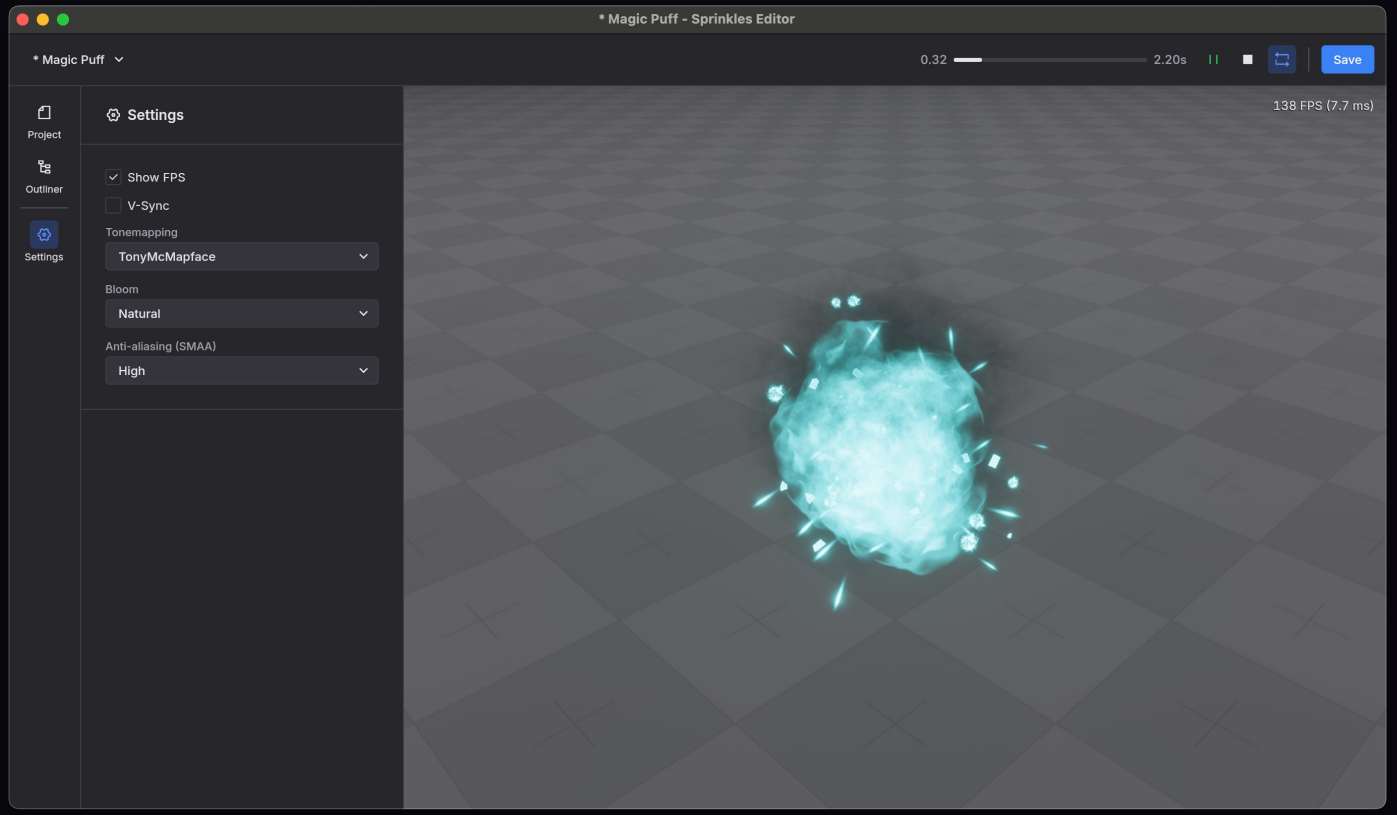
PART 2

Bulding nice UI with Bevy

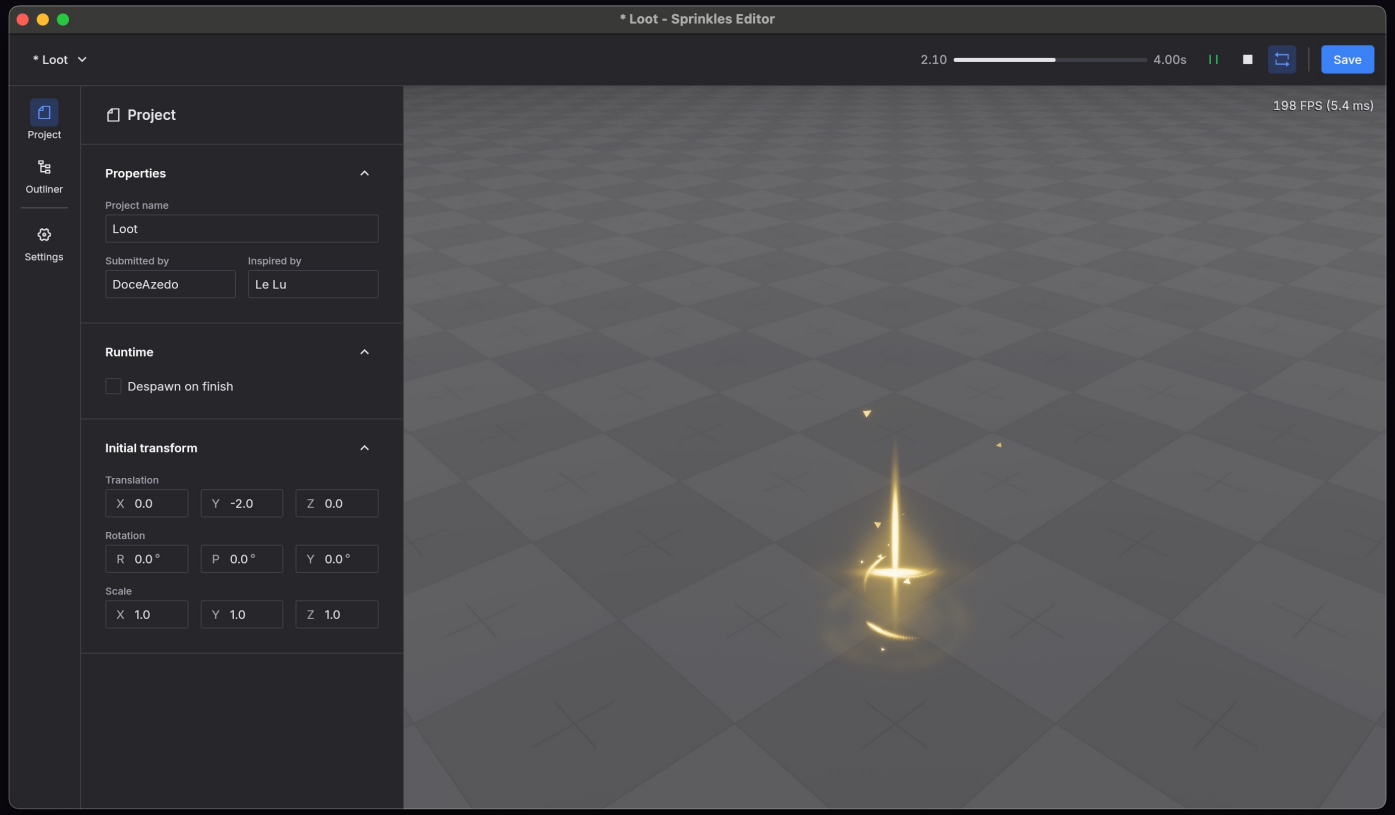
(from a frontend dev perspective)


* Magic Puff - Sprinkles Editor
* Magic Puff
0.32
2.20s
Save
Project
Outliner
Settings
Settings
Show FPS
V-Sync
Tonemapping
TonyMcMapface
Bloom
Natural
Anti-aliasing (SMAA)
High
138 FPS (7.7 ms)
* Explosion - Sprinkles Editor
* Explosion
0.75
3.30s
Save
Project
Outliner
Settings
Emitters
Windup
Smoke
Fire
Sparks
Shockwave
Debris
Colliders
Floor
Floor
Enabled
Properties
Shape
Box
Size
X 10.0
Y 0.1
Z 10.0
Initial transform
Translation
X 0.0
Y -2.0
Z 0.0
Rotation
R 0.0 °
P 0.0 °
Y 0.0 °
Scale
X 1.0
Y 1.0
Z 1.0
170 FPS (7.5 ms)

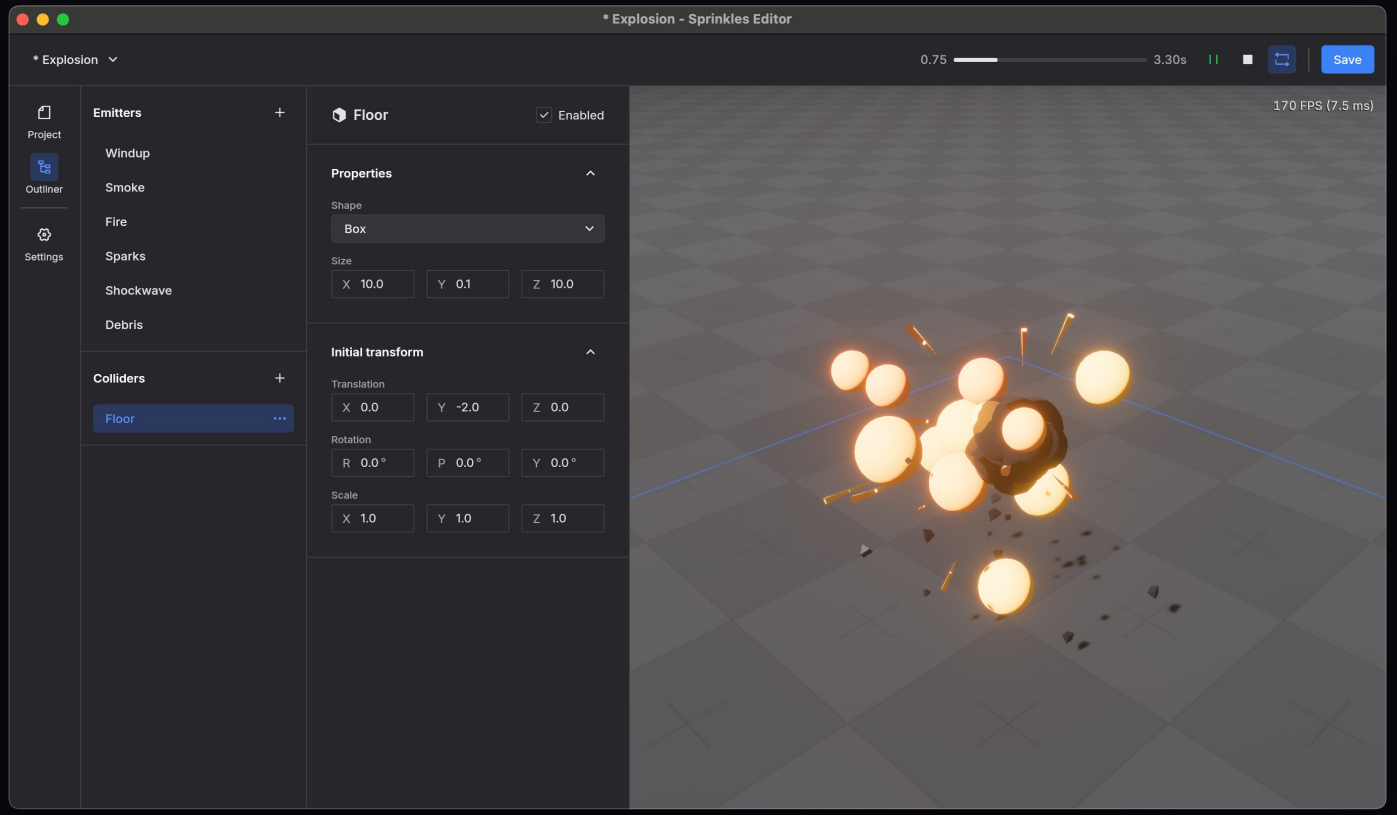
Three panel layout



Visual hierarchy



Colors & icons



```rust
// tokens.rs

pub const PRIMARY_COLOR: Srgba = tailwind::BLUE_500;
pub const BACKGROUND_COLOR: Srgba = tailwind::ZINC_800;
pub const BORDER_COLOR: Srgba = tailwind::ZINC_700;
pub const TEXT_BODY_COLOR: Srgba = tailwind::ZINC_200;
pub const TEXT_MUTED_COLOR: Srgba = tailwind::ZINC_400;
pub const TEXT_SIZE: f32 = 12.0;
pub const CORNER_RADIUS: Val = Val::Px(2.0);
pub const FONT_PATH: &str = "embedded://assets/InterVariable.ttf";
```

```
InspectorFieldProps::new("time.lifetime")
    .with_icon(ICON_TIME)
    .with_suffix("s")
```

```
InspectorSection::new("Time", vec![
    vec![
        InspectorFieldProps::new("time.lifetime")
            .with_icon(ICON_TIME)
            .with_suffix("s"),
        InspectorFieldProps::new("time.lifetime_randomness")
            .percent(),
    ],
    vec![
        InspectorFieldProps::new("time.one_shot")
        .bool(),
    ],
])
```

PART 3

Data binding & reflection

```rust
pub struct EmitterTime {
    pub lifetime: f32,
    pub lifetime_randomness: f32,
    pub delay: f32,
    // ...
}
```

```rust
#[derive(Reflect)]
pub struct EmitterTime {
    pub lifetime: f32,
    pub lifetime_randomness: f32,
    pub delay: f32,
    // ...
}
```

```
data.reflect_path(".time.lifetime")
// &dyn PartialReflect
```

```
FieldBinding {
    accessor: "time.lifetime",
    kind: FieldKind::F32,
    target: BindingTarget::Inspected,
}

// or
FieldBinding::emitter("time.lifetime", FieldKind::F32)
```

Step 1

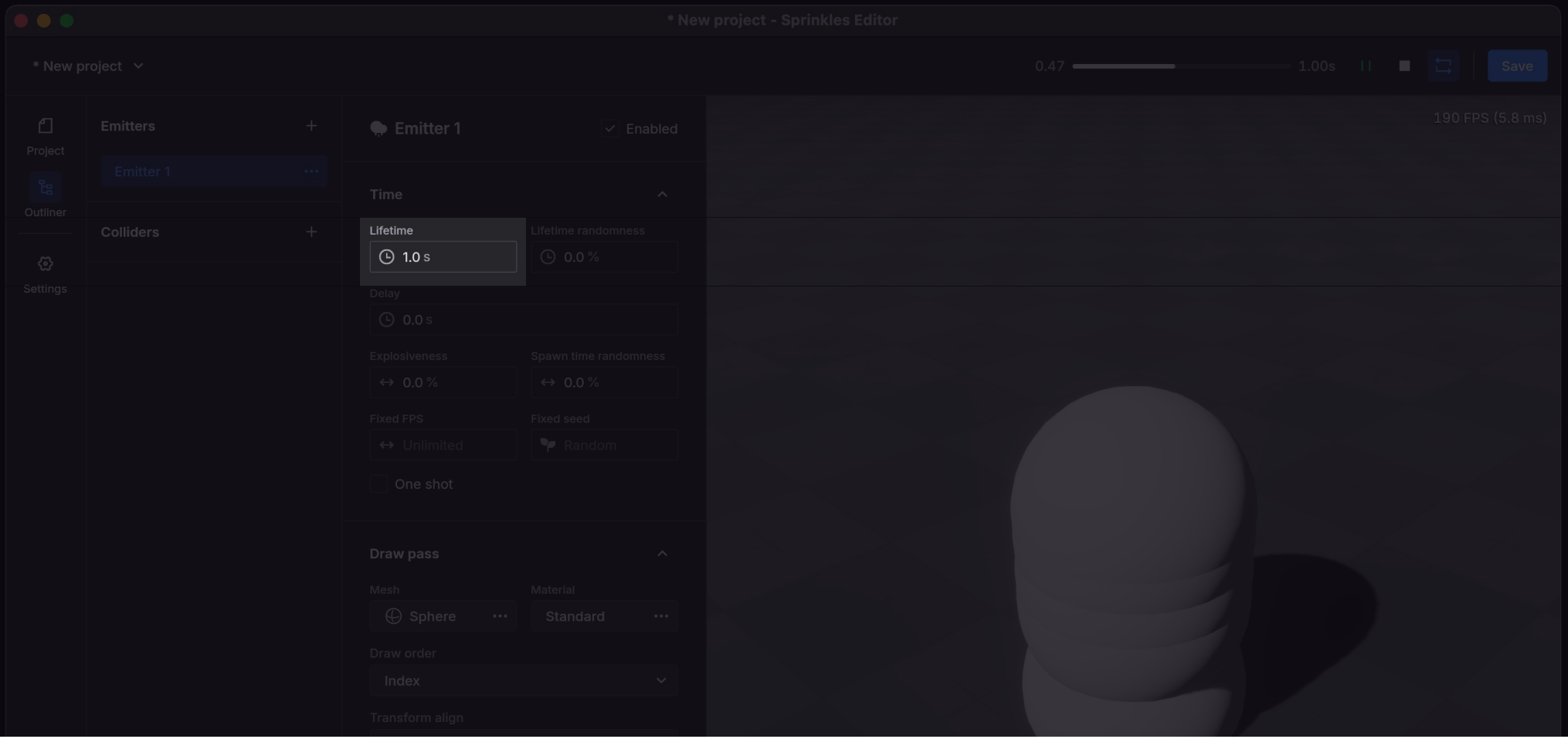
```
let path = ReflectPath::new(self.path()); // ".time.lifetime"
let value = data.reflect_path(path.as_str())?; // &dyn PartialReflect
```

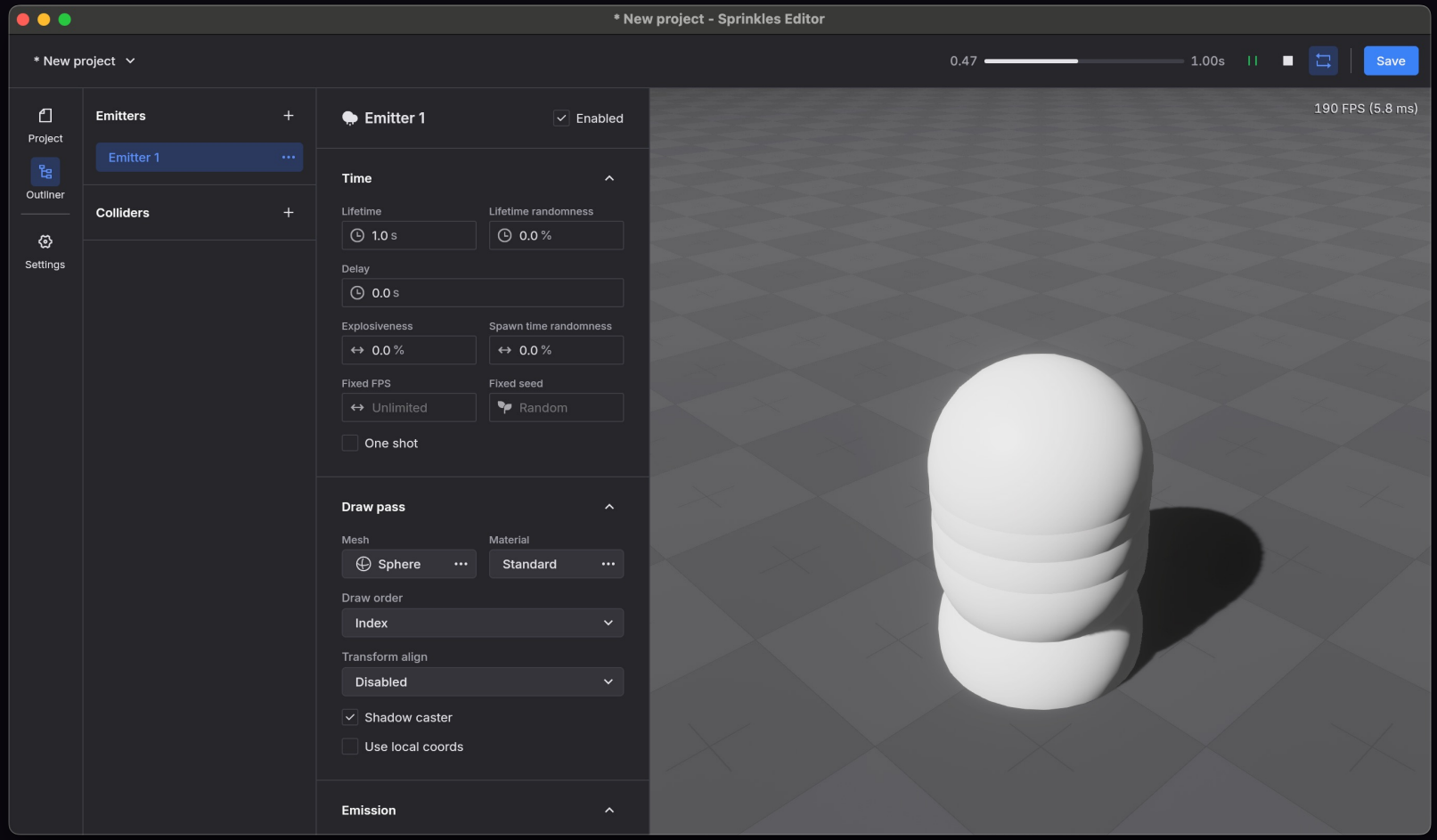
Step 2

```
if let Some(v) = value.try_downcast_ref::<f32>() {
    return FieldValue::F32(*v);
}

// ... and so on for each supported type
```

Step 3


* New project - Sprinkles Editor
* New project
0.47
1.00s
Save
Project
Outliner
Settings
Emitters
Emitter 1
Colliders
Emitter 1
Enabled
Time
Lifetime
1.0 s
Lifetime randomness
0.0 %
Delay
0.0 s
Explosiveness
0.0 %
Spawn time randomness
0.0 %
Fixed FPS
Unlimited
Fixed seed
Random
One shot
Draw pass
Mesh
Sphere
Material
Standard
Draw order
Index
Transform align
190 FPS (5.8 ms)


* New project - Sprinkles Editor
* New project
0.47
1.00s
Save
Project
Outliner
Settings
Emitters
Emitter 1
Colliders
Emitter 1
Enabled
Time
Lifetime
1.0 s
Lifetime randomness
0.0 %
Delay
0.0 s
Explosiveness
0.0 %
Spawn time randomness
0.0 %
Fixed FPS
Unlimited
Fixed seed
Random
One shot
Draw pass
Mesh
Sphere
Material
Standard
Draw order
Index
Transform align
Disabled
Shadow caster
Use local coords
Emission
190 FPS (5.8 ms)

```rust
let target = data.reflect_path_mut(path.as_str())?;
target.try_apply(&2.0_f32);
```

```
InspectorSection::new("Time", vec![
    vec![
        InspectorFieldProps::new("time.lifetime")
            .with_icon(ICON_TIME)
            .with_suffix("s"),
        InspectorFieldProps::new("time.lifetime_randomness")
            .percent(),
    ],
    vec![
        InspectorFieldProps::new("time.one_shot")
        .bool(),
    ],
])
```

PART 4

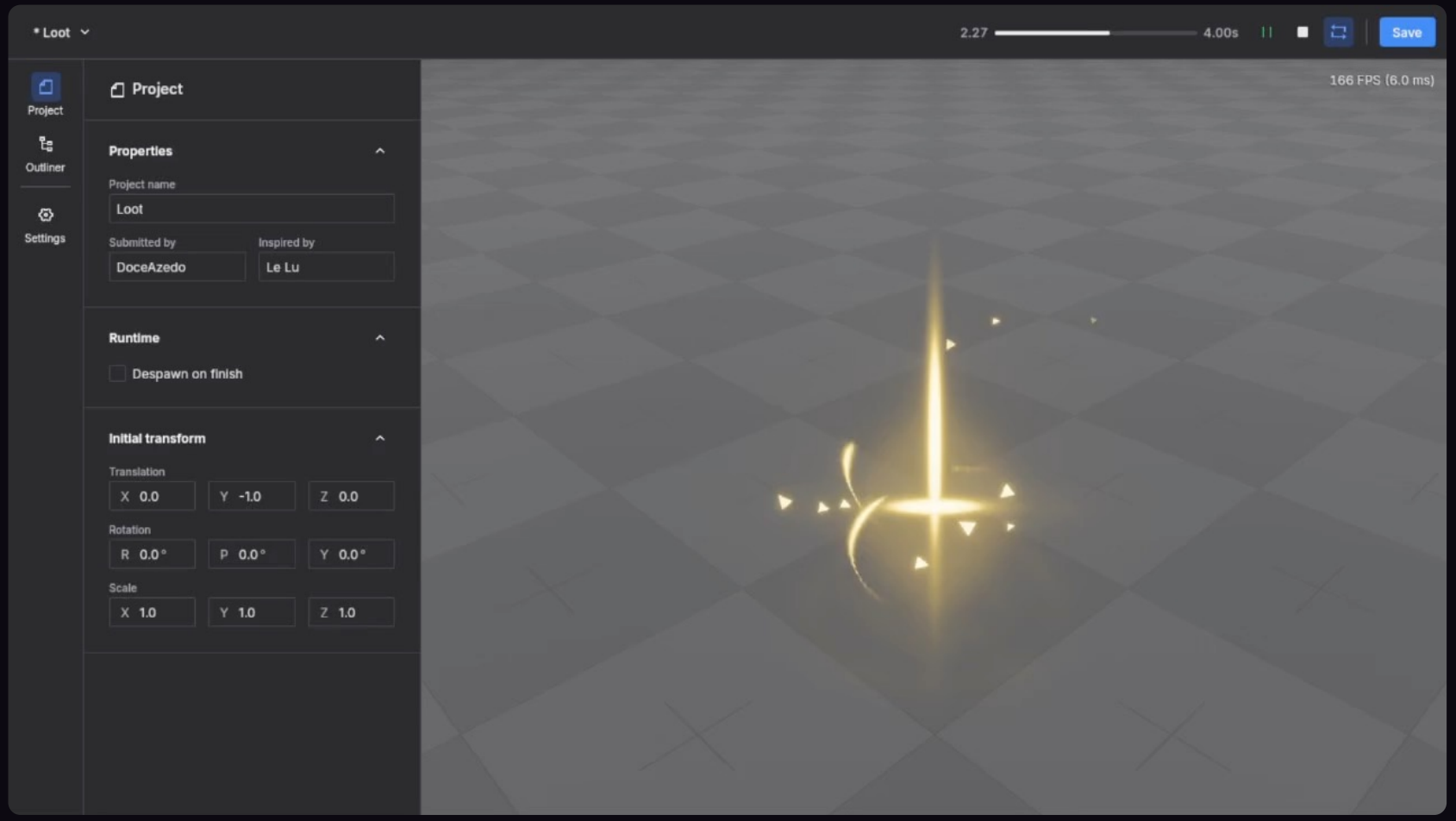
BSN in Bevy 0.19

```rust
cmd.spawn((
    Node {
        width: percent(100),
        height: percent(100),
        flex_direction: FlexDirection::Column,
        ..default()
    },
    children![
        (
            Text::new("Hello"),
            TextFont {
                font_size: 24.0,
                ..default()
            },
        ),
    ]
));
```

```
bsn! {
    Node {
        width: percent(100),
        height: percent(100),
        flex_direction: FlexDirection::Column,
    }
    Children [
        Text("Hello") TextFont { font_size: 24.0 }
    ]
}
```

```rust
fn button() -> impl Scene {
    bsn! {
        Button
        Children [ Text("Button") ]
    }
}

fn icon_button(icon: &str) -> impl Scene {
    bsn! {
        button()
        Children [ Image { path: {icon} } ]
    }
}
```


* Loot
2.27
4.00s
Save
Project
Outliner
Settings
Project
Properties
Project name
Loot
Submitted by
DoceAzedo
Inspired by
Le Lu
Runtime
Despawn on finish
Initial transform
Translation
X 0.0
Y -1.0
Z 0.0
Rotation
R 0.0°
P 0.0°
Y 0.0°
Scale
X 1.0
Y 1.0
Z 1.0
166 FPS (6.0 ms)

Thank you!

All my links:

- doce.sh
- github.com/doceazedo
- bsky.app/profile/doceazedo.com