



farseerTM
physics



Universe Rampage

- Prototype -



Content

- Physics
- Entities
- Particles
- Controls
- Prototype



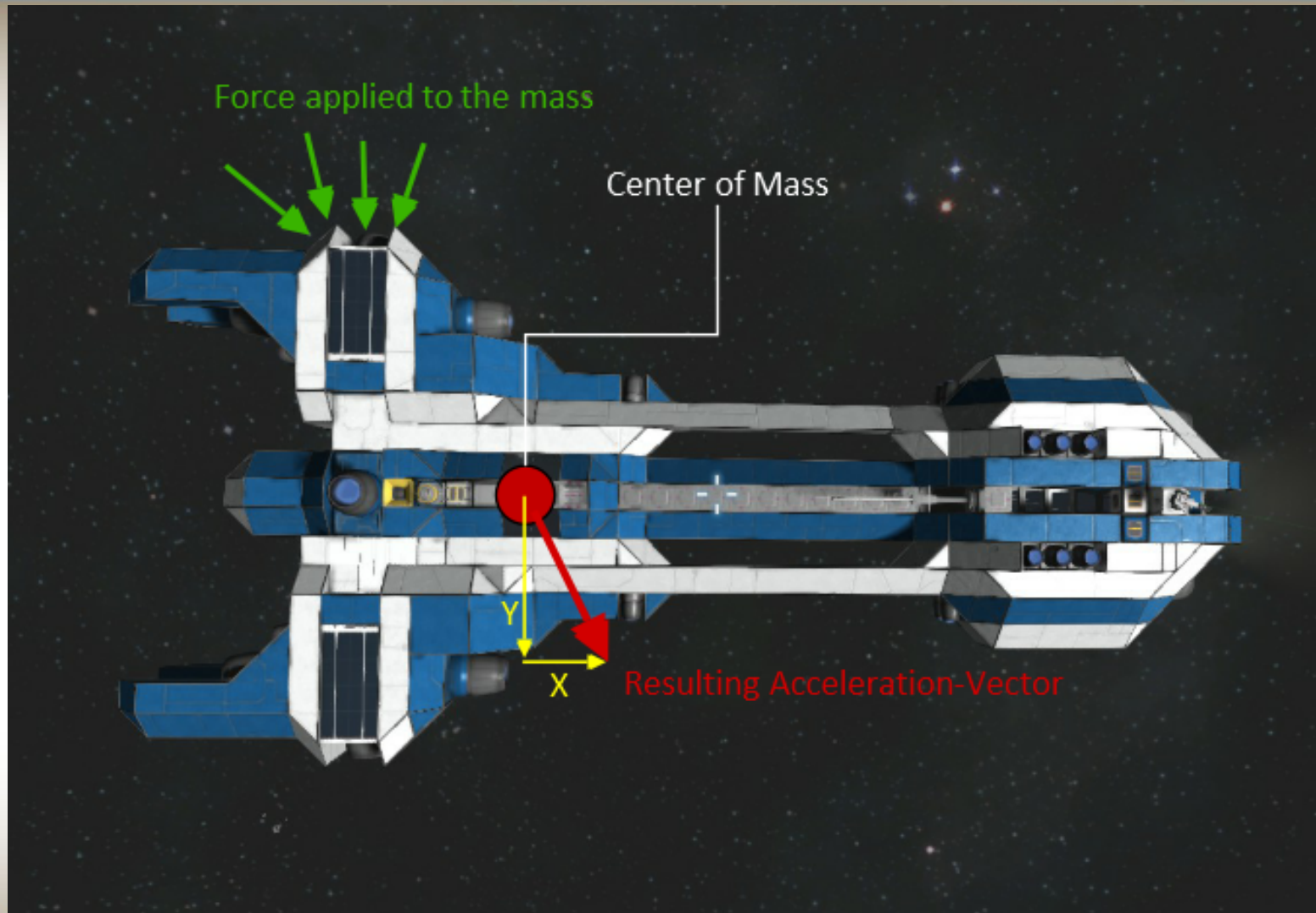
Physics

- Physics handled by Farseer Engine
- Body and World as central physics elements
- Physics divided in movement and collision



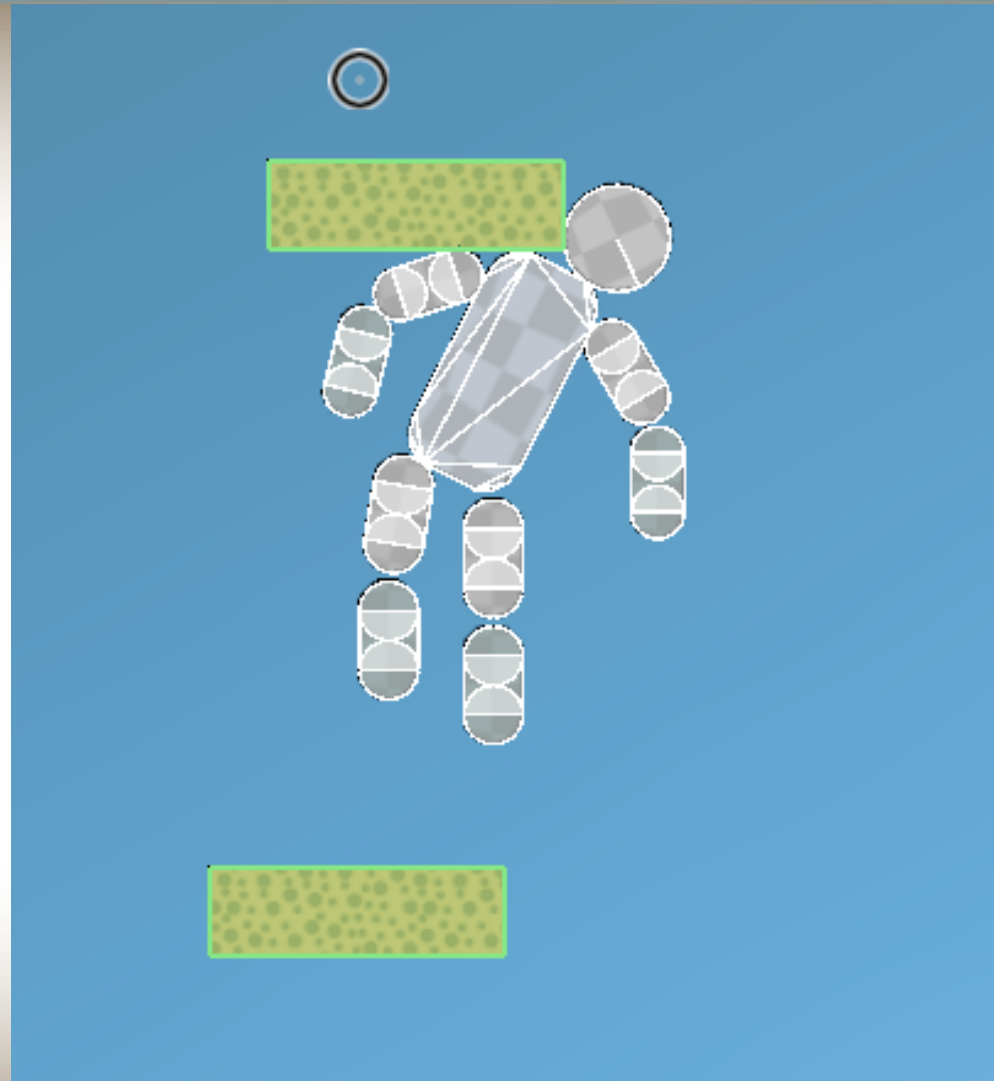


Physics - Movement





Physics - Collision





Entities



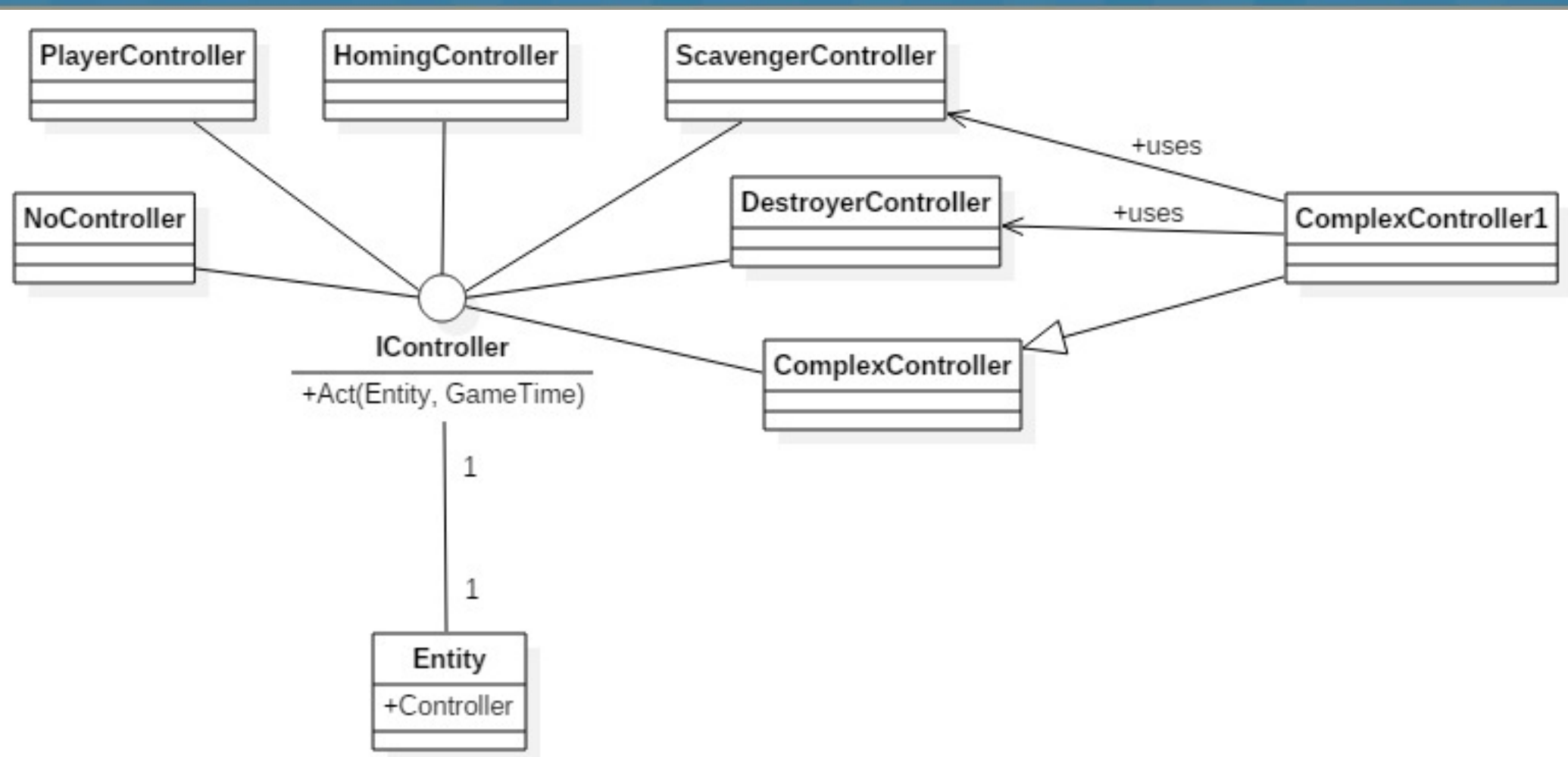


Entities - Controllers

- Entities are controlled by Controller Interface
- Controller defined for different behaviors
- Complex behaviors by combining controllers in a complex controller



Entities - Controllers



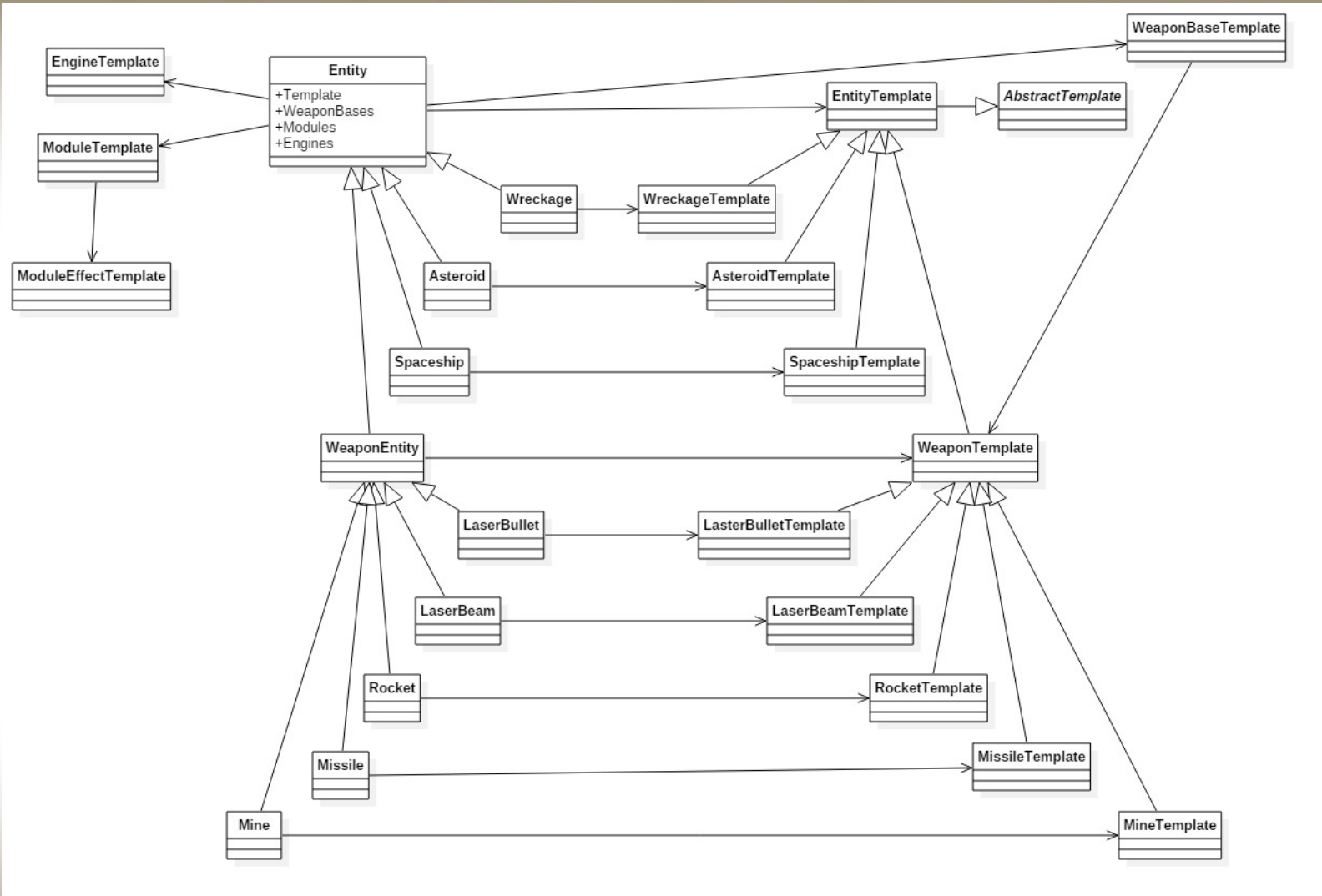


Entities - Templates

- Own Template for each object type
- Templates loaded from xml files
- Xml Files encrypted through content pipeline
- Entities initialized dynamically

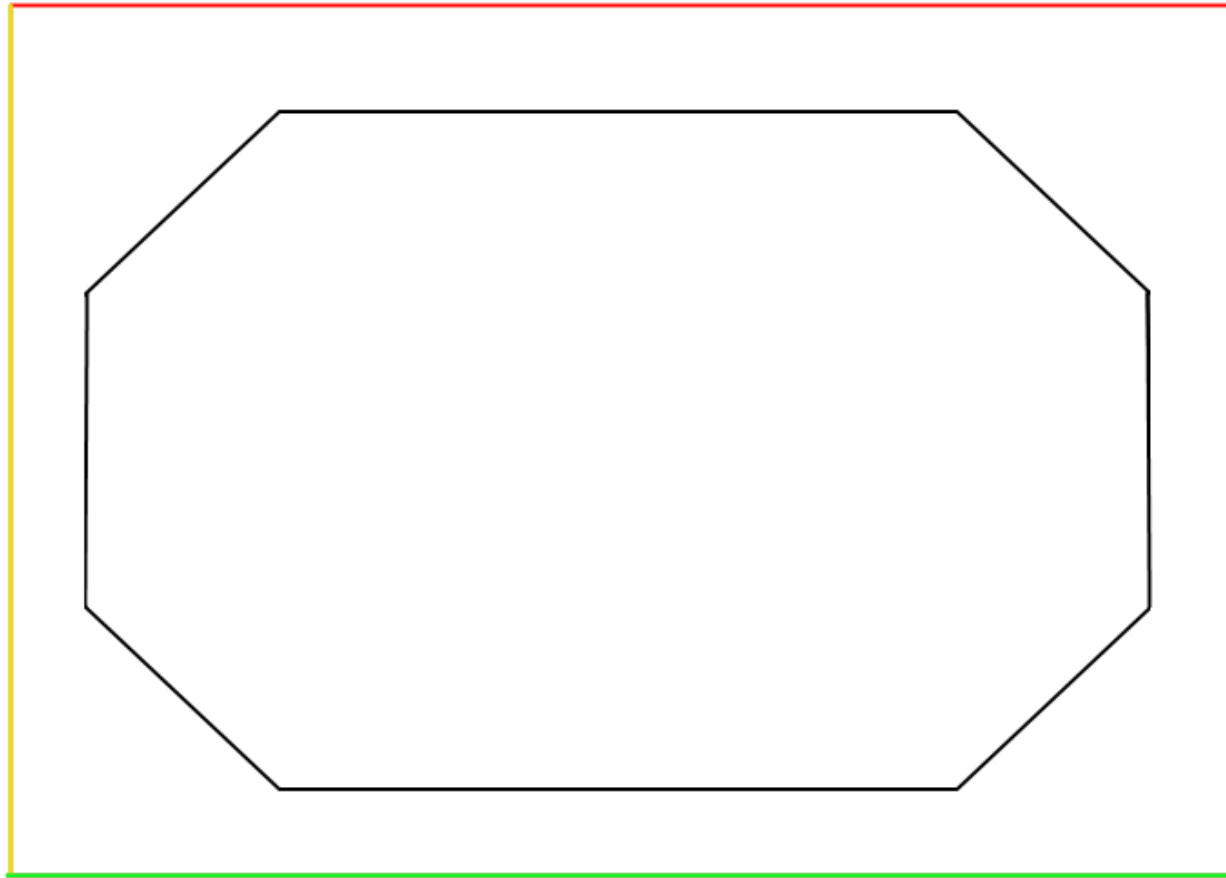


Entities - Templates



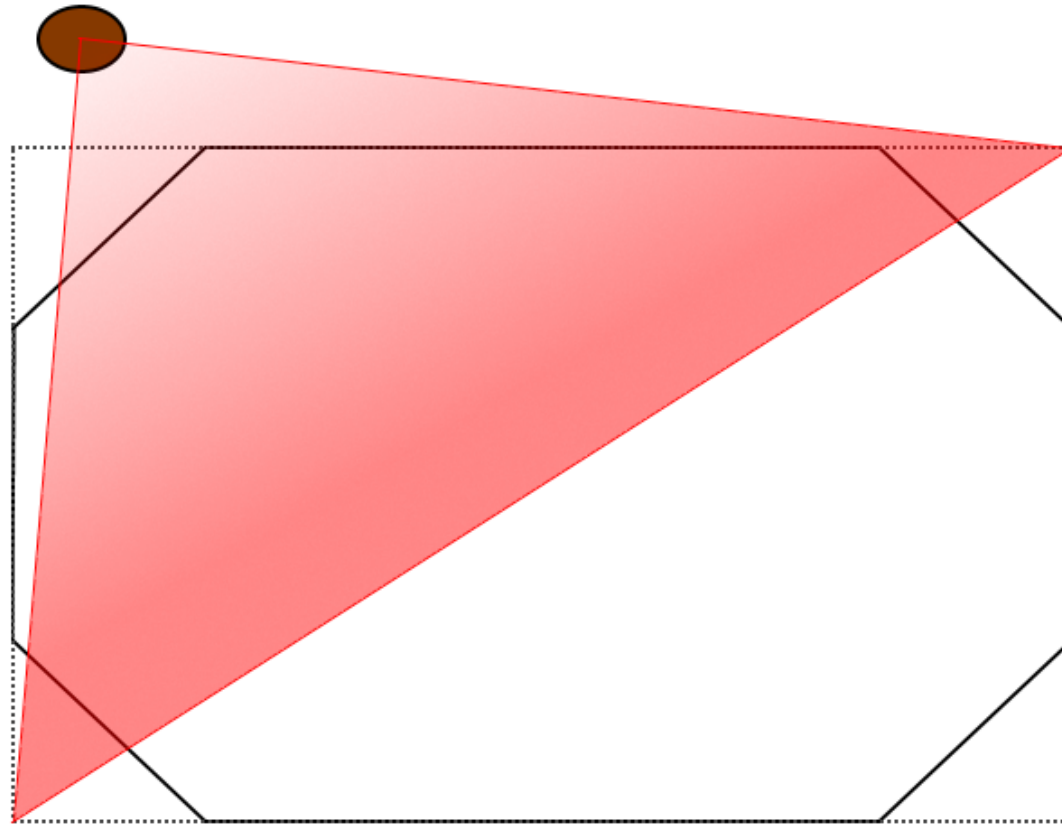


Entities - Asteroid Spawn



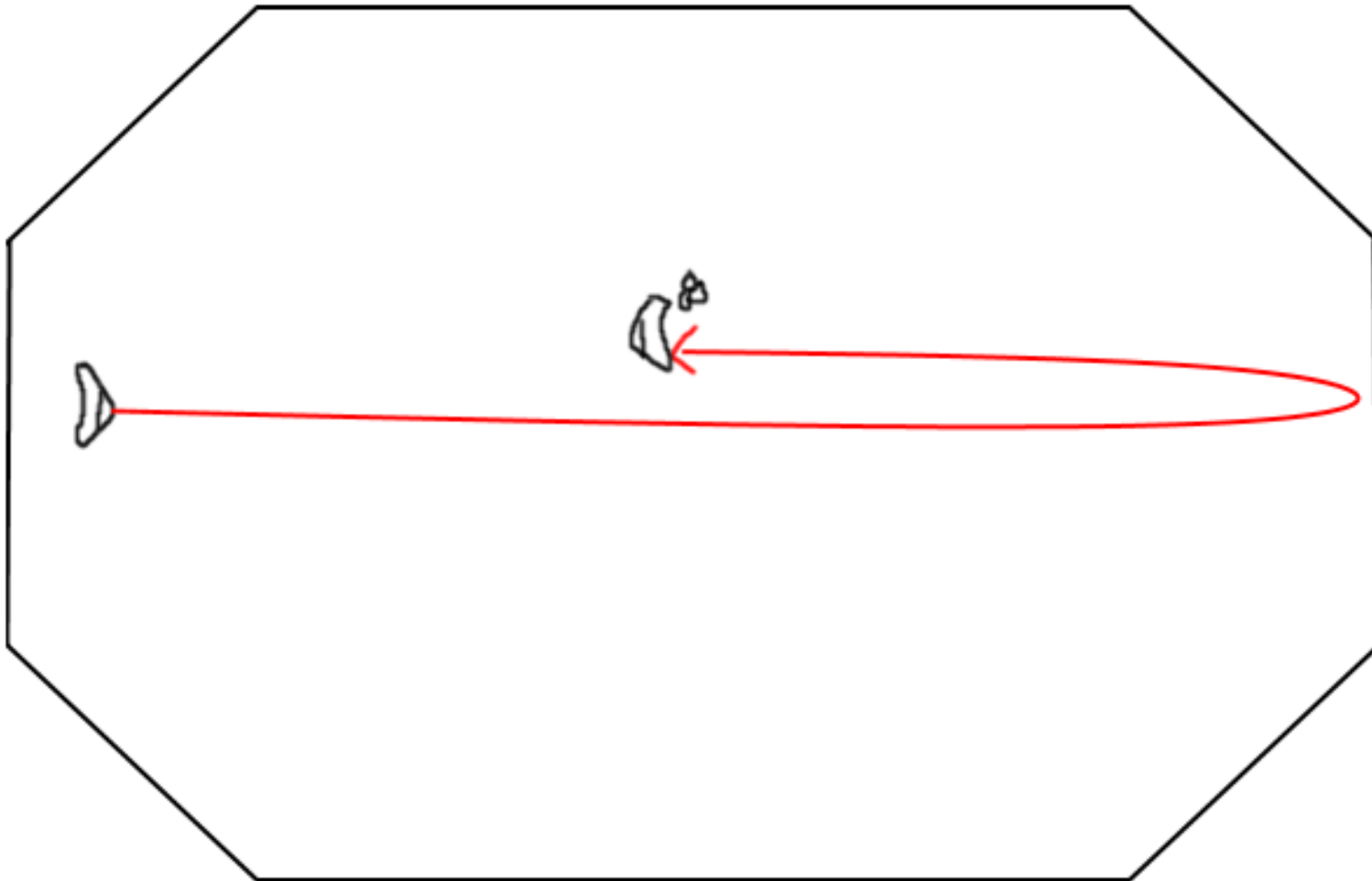


Entities - Asteroid Spawn



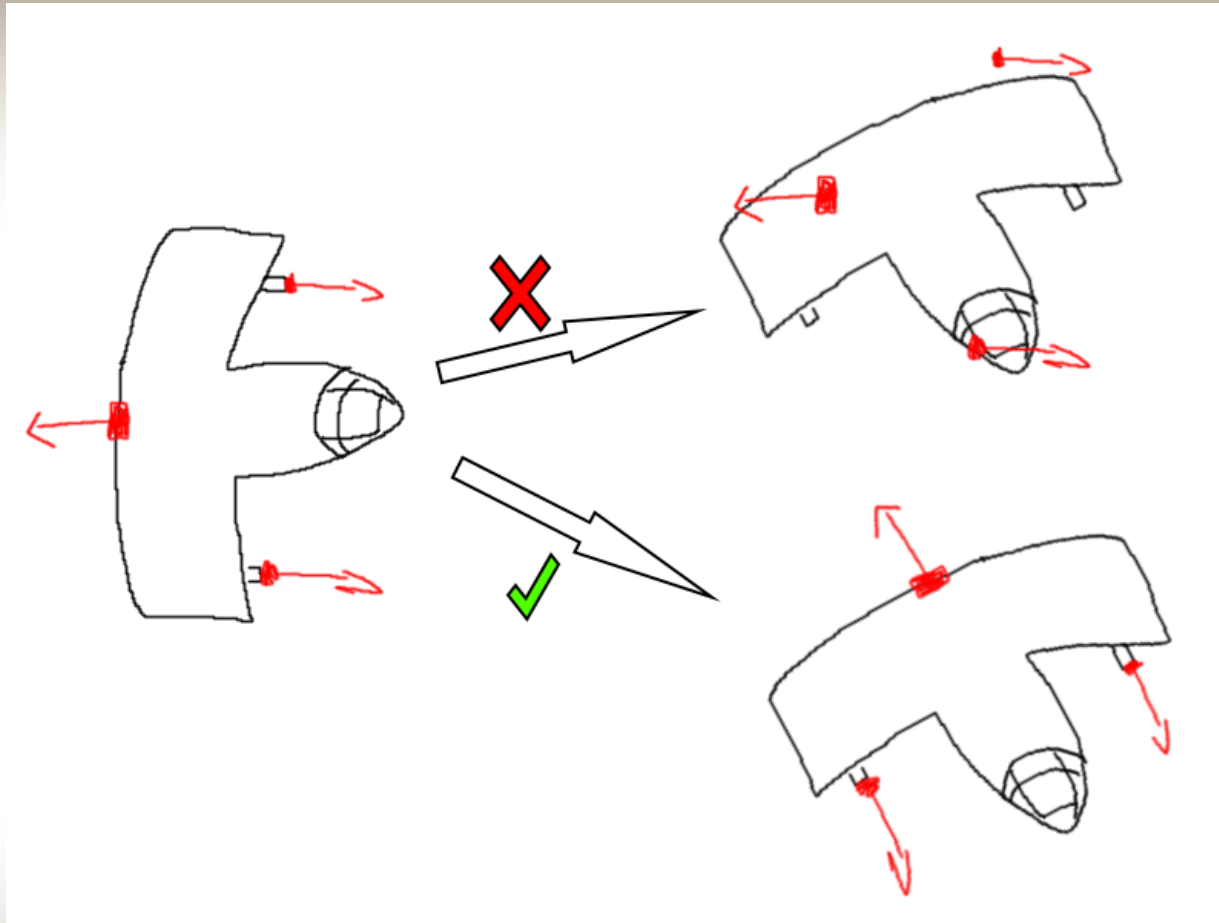


Entities - Fuel Capacity





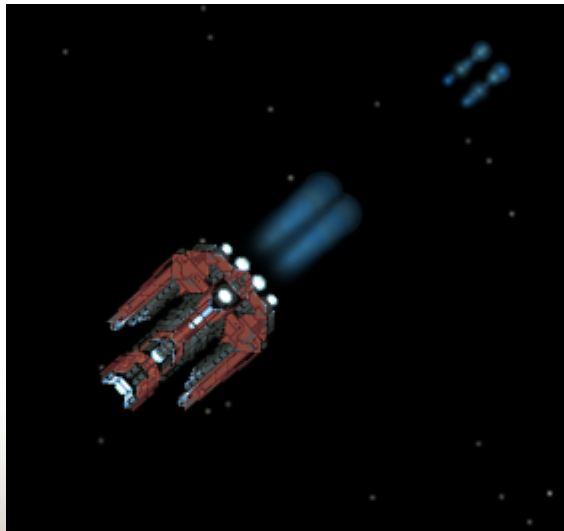
Entities - Local Transformation





Particles

- Particles completely handled by GPU
- Rendering and update with HLSL and DirectX
- Lifetime attributes calculated per particle type





Particles

```
150 // Custom vertex shader animates particles entirely on the GPU.
151 VertexShaderOutput ParticleVertexShader(VertexShaderInput input)
152 {
153     VertexShaderOutput output;
154
155     // Compute the age of the particle.
156     float age = CurrentTime - input.Time;
157
158     // Apply a random factor to make different particles age at different rates.
159     age *= 1 + input.Random.x * DurationRandomness;
160
161     // Normalize the age into the range zero to one.
162     float normalizedAge = saturate(age / Duration);
163
164     // Compute the particle position, size, color, and rotation.
165     output.Position = ComputeParticlePosition(input.Position, input.Velocity,
166         age, normalizedAge);
167
168     float size = ComputeParticleSize(input.Random.y, normalizedAge);
169     float2x2 rotation = ComputeParticleRotation(input.Random.w, age);
170
171     output.Position.xy += mul(input.Corners, rotation) * size * ViewportScale;
172
173     output.Color = ComputeParticleColor(output.Position, input.Random.z, normalizedAge);
174     output.TextureCoordinate = (input.Corners + 1) / 2;
175
176     return output;
177 }
178
179 // Pixel shader for drawing particles.
180 float4 ParticlePixelShader(VertexShaderOutput input) : COLOR0
181 {
182     return tex2D(Sampler, input.TextureCoordinate) * input.Color;
183 }
184
```



Controls





Prototype





The End

THANK YOU FOR YOUR ATTENTION

PLEASE CLAP AND DON'T ASK TOUGH QUESTIONS

memecrunch.com