

Developing a Jump'n'Run puzzle game

Samiro Discher

Alexander Kugler

Christopher Kugler

Christof Mroz

RWTH Aachen University

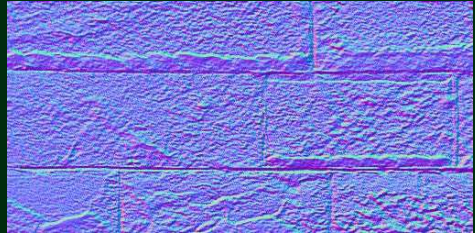
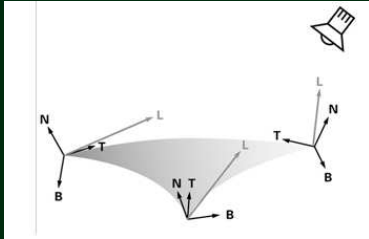
11. April 2012



Rendering



Normal Mapping



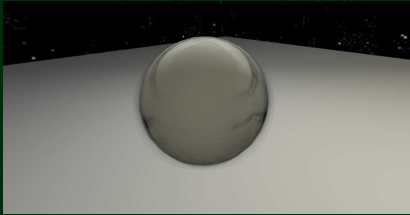
Difficulties

- Texture to Object space: matrix of binormal, tangent and normal
- tangents computed from neighbouring vertices

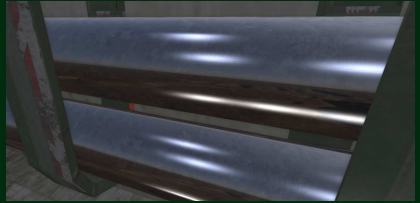


Reflection

Metallic



Environment Mapping



Fresnel Effect

- depending on viewing angle the amount of reflectance changes
- the reflected color may be mixed with either a base color or even a refracted color

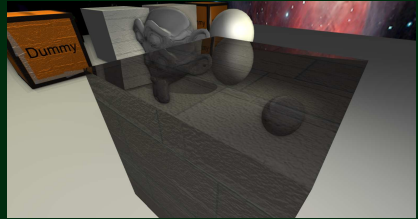


Refraction

Without Normalmapping



With Normalmapping



Dynamic vs. Static

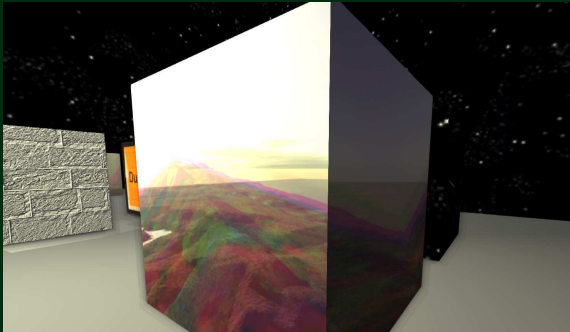
- static: uses a cube map
- dynamic: draw object as late as possible and use current screen buffer as lookup, solve lookup outside of screen buffer by clamping



Chromatic Dispersion

Characteristics

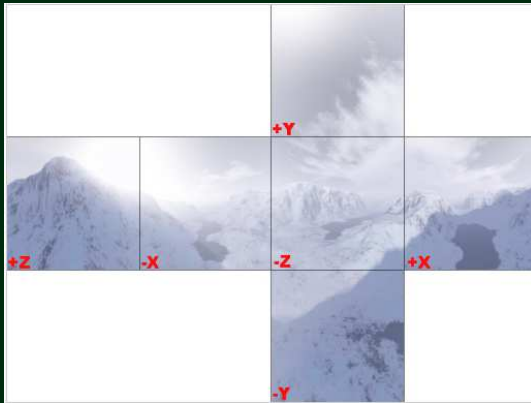
- different refraction for red, blue and green depending on refraction ratio (material dependent)
- can be combined with reflection (Fresnel Equation)



Skybox

Characteristics

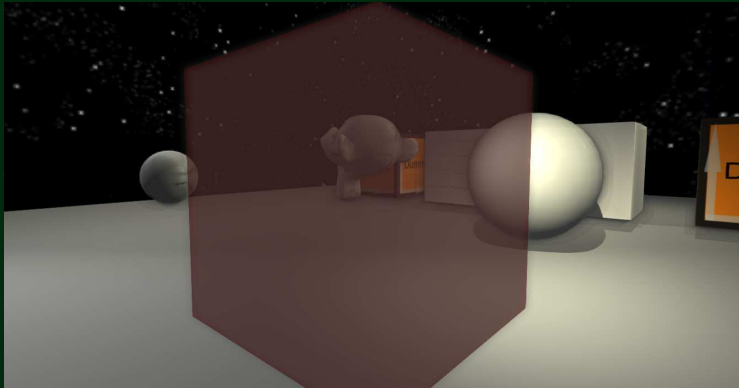
- Cube Map with camera as center



Transparency

Characteristics

- draw transparent objects in a second pass, use alpha blending



Post Processing

Effects

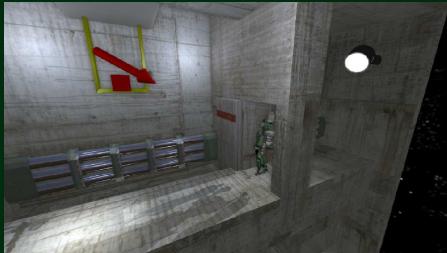
- Deferred Lighting
- Motion Blur
- FXAA
- SSAO
- Greyscale
- Glow



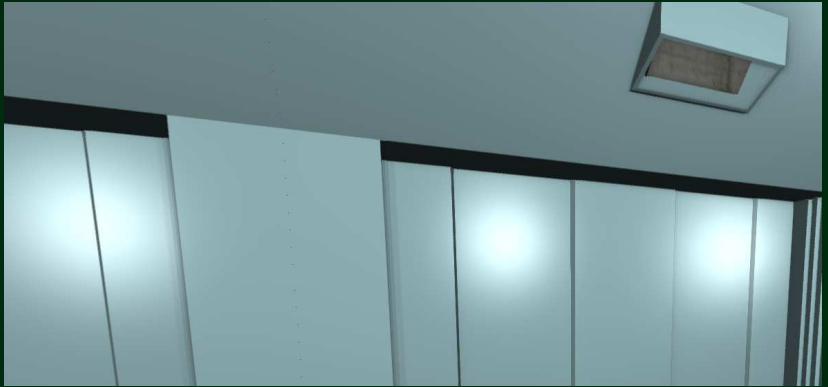
Deferred Lighting

Characteristics

- use depth buffer to calculate light direction in eye space
- use normal buffer and light direction to calculate the diffuse and specular amount
- combine diffuse, specular, ssao and the material's characteristics (fresnel exponent and intensity) adequately



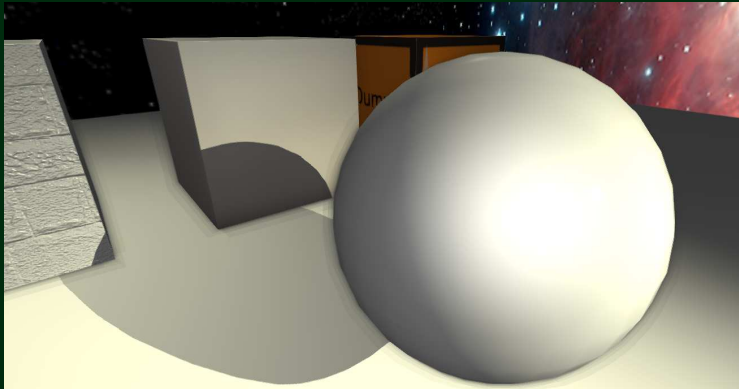
Pointlights



Spotlights

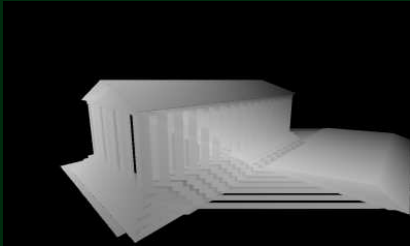
Characteristics

- distance and angle attenuation

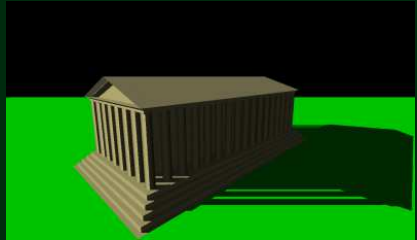


Spotlights With Shadows

Projected Geometry Pass



Lighting Pass



Pipeline characteristics

- Geometry Pass: draw geometry only from light position and save into buffer
- Lighting Pass: use generated buffer to determine whether pixel is in shadow or not



Motion Blur

Velocity Buffer



Result



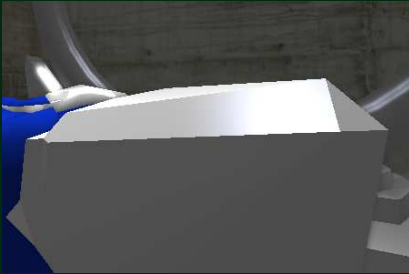
Pipeline characteristics

- compute velocity with current MVP and MVP of the last frame



Anti-Aliasing

Without Anti-Aliasing



With FXAA

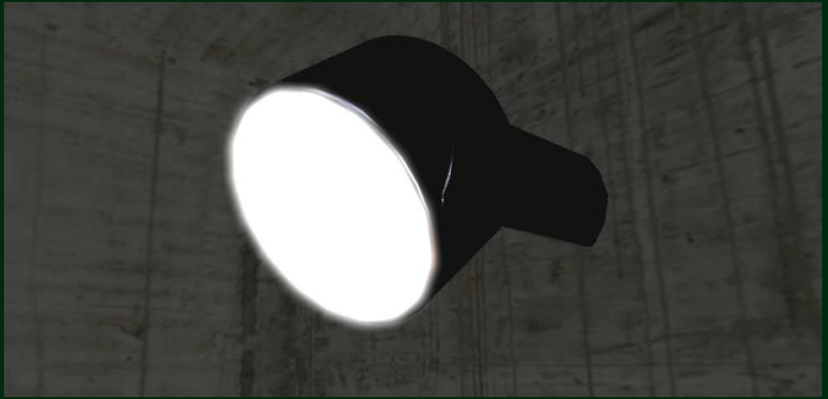


Pipeline characteristics

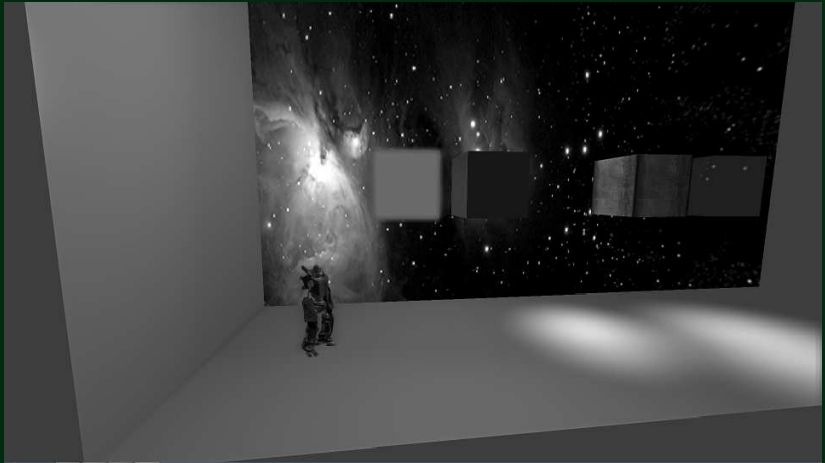
- high quality anti-aliasing with a single-pass post-process



Glow



Greyscale



Assets



Assets

Modeling

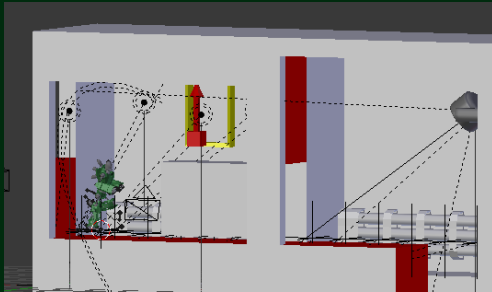
- 14 assets with 12 different functionalities
- planned puzzle logic system



Tools

- used Blender 2.59
- importer for 3D-objects, materials, textures (alpha, normal)

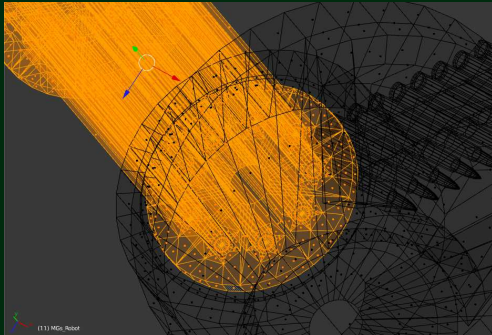
1st Level (in Blender)



Animation

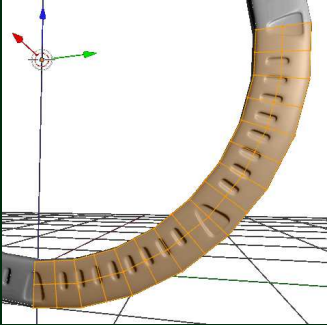
- animation applied with scene graph
- movable objects splitting into several sub-objects

Robots MG

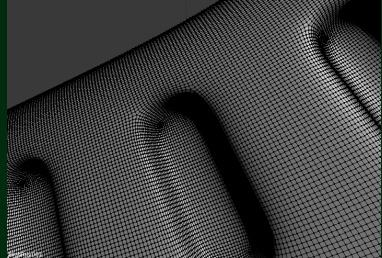


Normal Maps

Low Poly

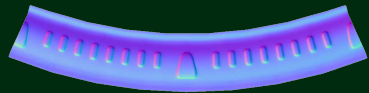


High Poly



- creation of normalmaps with baking high on low poly objects

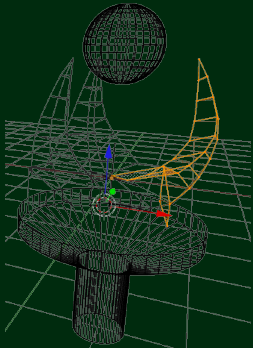
Normal Map



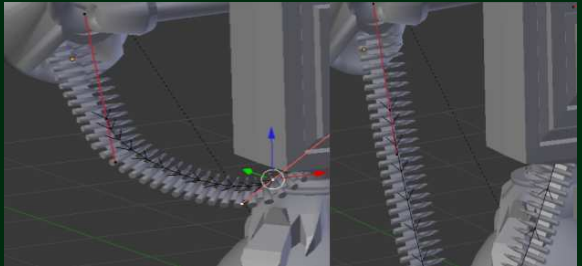
Modifiers

- clever use of modifiers

Eye Geometry



Robot Ammunition



Game Engine

Post Mortem

- The result was a disaster
- Refactoring costs → New feature development almost stalled
- How could this happen?
 - Presentation logic: unexpected complexity
 - Object-oriented entity hierarchy
- Engine was rewritten
 - but not in time, unfortunately



Game Engine

Post Mortem

- The result was a disaster
- Refactoring costs → New feature development almost stalled
- How could this happen?
 - Presentation logic: unexpected complexity
 - Object-oriented entity hierarchy
- Engine was rewritten
 - but not in time, unfortunately



Game Engine

Post Mortem

- The result was a desaster
- Refactoring costs → New feature development almost stalled
- How could this happen?
 - Presentation logic: unexpected complexity
 - Object-oriented entity hierarchy
- Engine was rewritten
 - but not in time, unfortunately



Game Engine

Post Mortem

- The result was a desaster
- Refactoring costs → New feature development almost stalled
- How could this happen?
 - Presentation logic: unexpected complexity
 - Object-oriented entity hierarchy
- Engine was rewritten
 - but not in time, unfortunately



Game Engine

Post Mortem

- The result was a desaster
- Refactoring costs → New feature development almost stalled
- How could this happen?
 - Presentation logic: unexpected complexity
 - Object-oriented entity hierarchy
- Engine was rewritten
 - but not in time, unfortunately



Game Engine

Post Mortem

- The result was a desaster
- Refactoring costs → New feature development almost stalled
- How could this happen?
 - Presentation logic: unexpected complexity
 - Object-oriented entity hierarchy
- Engine was rewritten
 - but not in time, unfortunately



Game Engine

Post Mortem

- The result was a desaster
- Refactoring costs → New feature development almost stalled
- How could this happen?
 - Presentation logic: unexpected complexity
 - Object-oriented entity hierarchy
- Engine was rewritten
 - but not in time, unfortunately



Game Engine

Post Mortem

- The result was a desaster
- Refactoring costs → New feature development almost stalled
- How could this happen?
 - Presentation logic: unexpected complexity
 - Object-oriented entity hierarchy
- Engine was rewritten
 - but not in time, unfortunately



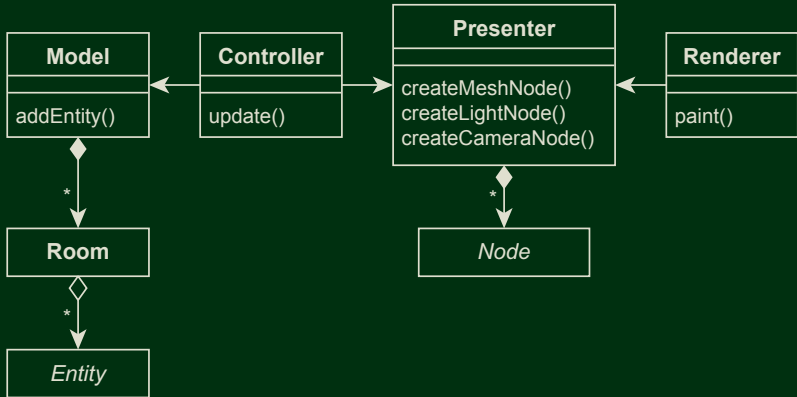
Game Engine

Post Mortem

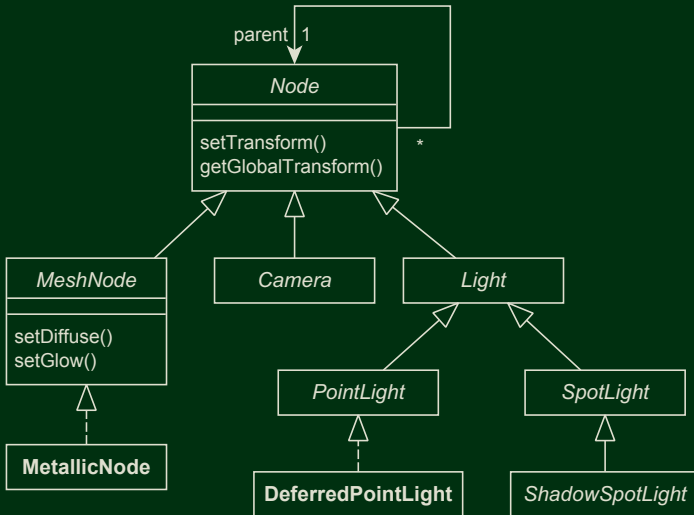
- The result was a desaster
- Refactoring costs → New feature development almost stalled
- How could this happen?
 - Presentation logic: unexpected complexity
 - Object-oriented entity hierarchy
- Engine was rewritten
 - but not in time, unfortunately



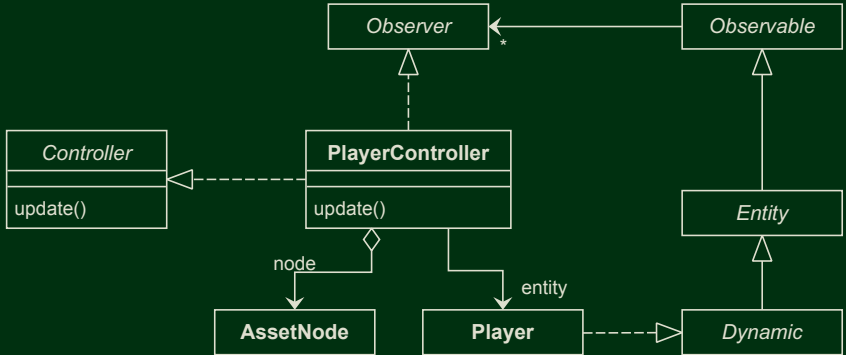
Architecture



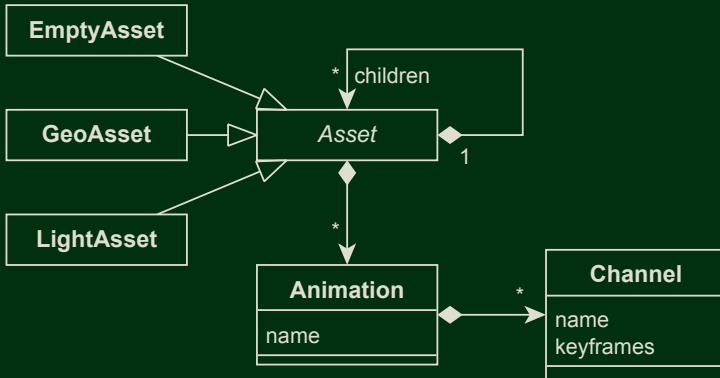
Scene Node Hierarchy



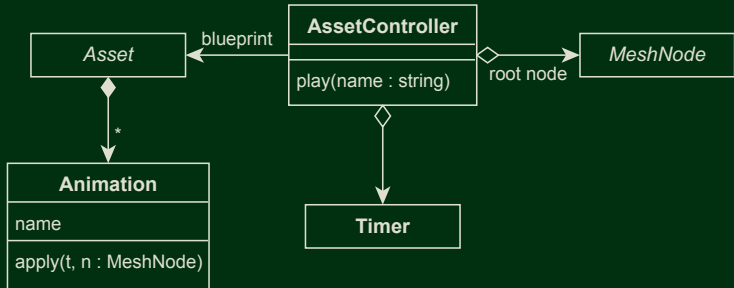
Entity Controllers



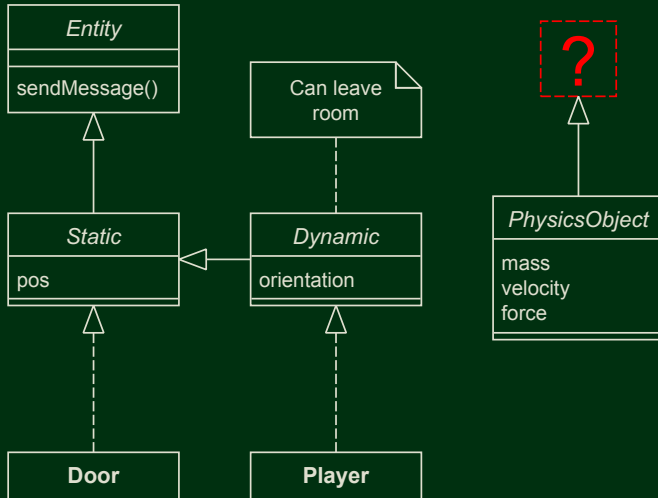
Asset Definition



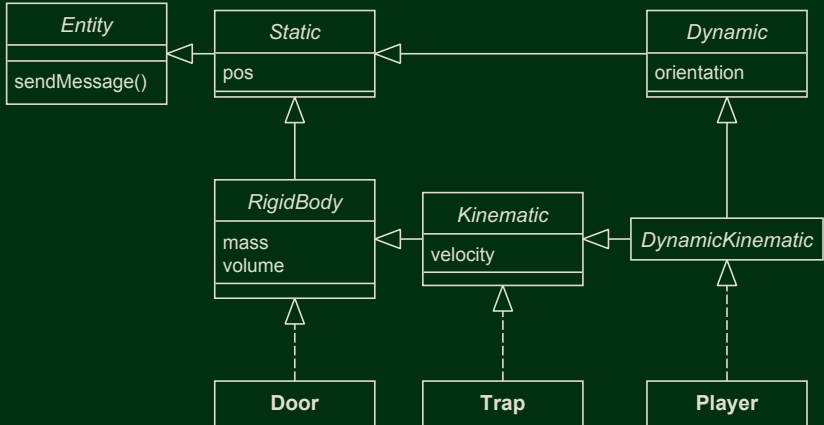
Asset Instantiation



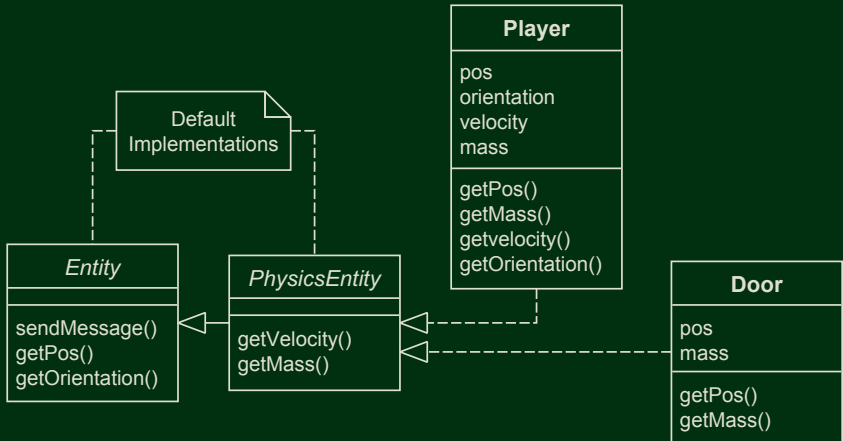
Approach: Entity Hierarchy



Idea: Refine Taxonomy



Our Solution: Blob Anti-Pattern



Discussion of the OO approach

Summary

- Placing new classes into the hierarchy: distracting
- Changes may propagate from root to subclass
- Changed requirements \implies (maybe) splitting of taxonomy
- Cost of refactoring inhibits creativity
- Maybe we just need to think harder?

Alternative Solution

Don't base entity management/processing on OOP to begin with



Discussion of the OO approach

Summary

- Placing new classes into the hierarchy: distracting
- Changes may propagate from root to subclass
- Changed requirements \implies (maybe) splitting of taxonomy
- Cost of refactoring inhibits creativity
- Maybe we just need to think harder?

Alternative Solution

Don't base entity management/processing on OOP to begin with



Discussion of the OO approach

Summary

- Placing new classes into the hierarchy: distracting
- Changes may propagate from root to subclass
- Changed requirements \implies (maybe) splitting of taxonomy
- Cost of refactoring inhibits creativity
- Maybe we just need to think harder?

Alternative Solution

Don't base entity management/processing on OOP to begin with



Discussion of the OO approach

Summary

- Placing new classes into the hierarchy: distracting
- Changes may propagate from root to subclass
- Changed requirements \implies (maybe) splitting of taxonomy
- Cost of refactoring inhibits creativity
- Maybe we just need to think harder?

Alternative Solution

Don't base entity management/processing on OOP to begin with



Discussion of the OO approach

Summary

- Placing new classes into the hierarchy: distracting
- Changes may propagate from root to subclass
- Changed requirements \implies (maybe) splitting of taxonomy
- Cost of refactoring inhibits creativity
- Maybe we just need to think harder?

Alternative Solution

Don't base entity management/processing on OOP to begin with



Discussion of the OO approach

Summary

- Placing new classes into the hierarchy: distracting
- Changes may propagate from root to subclass
- Changed requirements \implies (maybe) splitting of taxonomy
- Cost of refactoring inhibits creativity
- Maybe we just need to think harder?

Alternative Solution

Don't base entity management/processing on OOP to begin with



Discussion of the OO approach

Summary

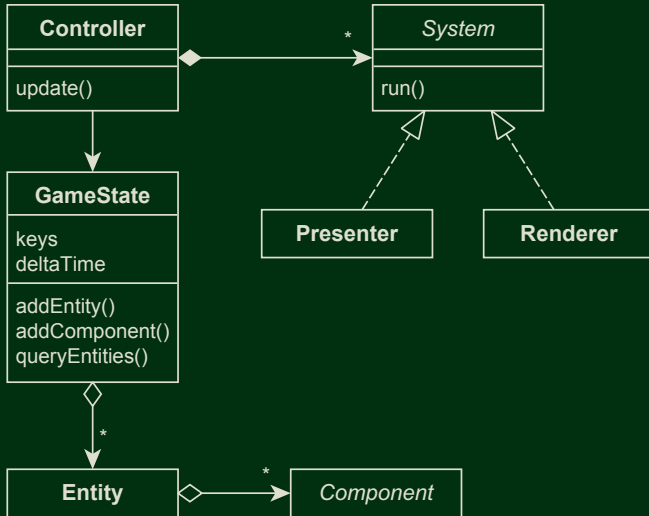
- Placing new classes into the hierarchy: distracting
- Changes may propagate from root to subclass
- Changed requirements \implies (maybe) splitting of taxonomy
- Cost of refactoring inhibits creativity
- Maybe we just need to think harder?

Alternative Solution

Don't base entity management/processing on OOP to begin with



New engine: Entity System



Entity System Example

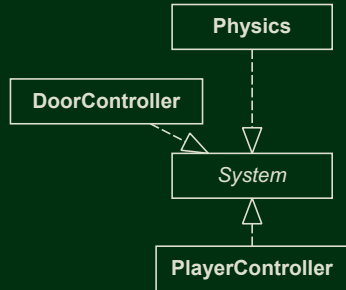
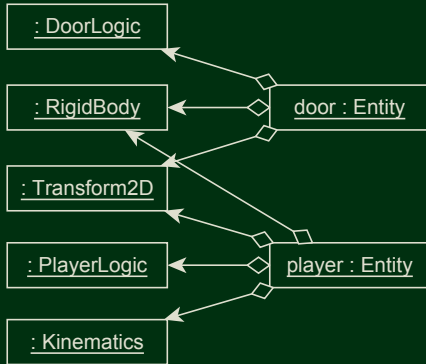


Image Sources

- <http://de.wikipedia.org/w/index.php?title=Datei:7fin.png>
- <http://en.wikipedia.org/wiki/File:4overmap.png>
- <http://wiki.delphigl.com/index.php/Datei:Skybox.jpg>

