

Modular Sandbox Design: Tools and Workflows for Hitman

Marinus Rørbech

Senior Programmer, Io-Interactive



GAME DEVELOPERS CONFERENCE™ EUROPE
CONGRESS-CENTRUM OST KOELNMESSE · COLOGNE, GERMANY
AUGUST 3-4, 2015



Who Am I?

Weird letter



- Marinus Rørbech
- Senior Programmer, Io-Interactive
- 8 years in the industry

The Bald Guy
(not me)





Agenda

- Motivation
- Introducing "Bricks"
- Impact
- Wrap Up



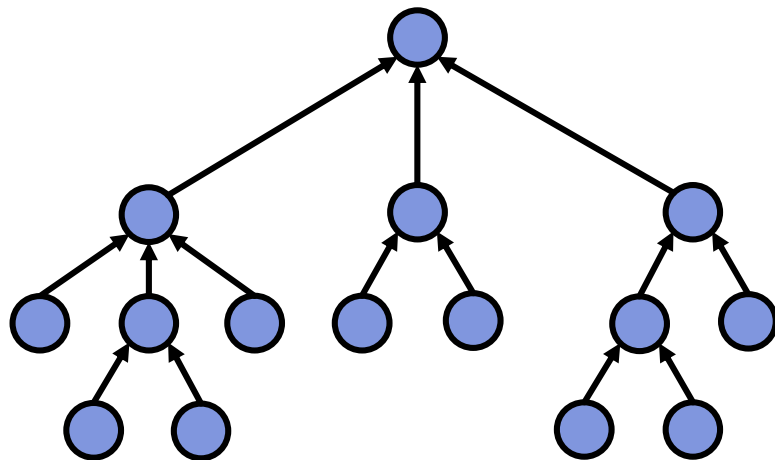
Agenda

- **Motivation**
- Introducing "Bricks"
- Impact
- Wrap Up



Traditional level structure at IOI

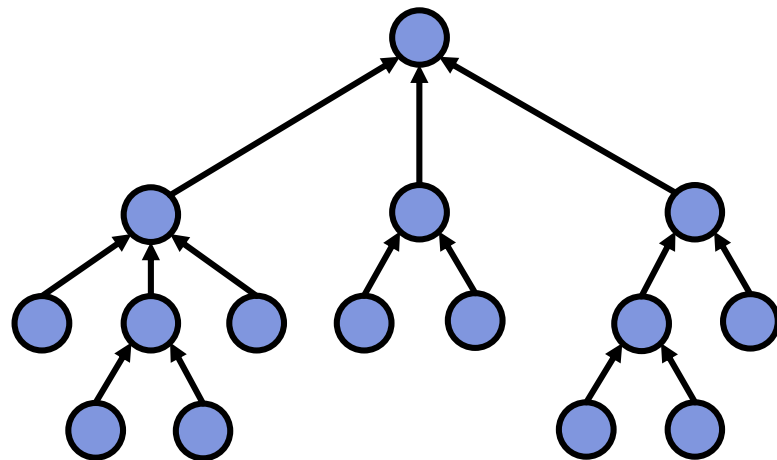
- Scene
 - Tree hierarchy of "entities"
 - Geometry
 - Logic from C++
 - Logic from visual programming
- Hitman: Absolution
 - 1 checkpoint = 1 scene \approx 30.000 entities





Traditional level design at IOI

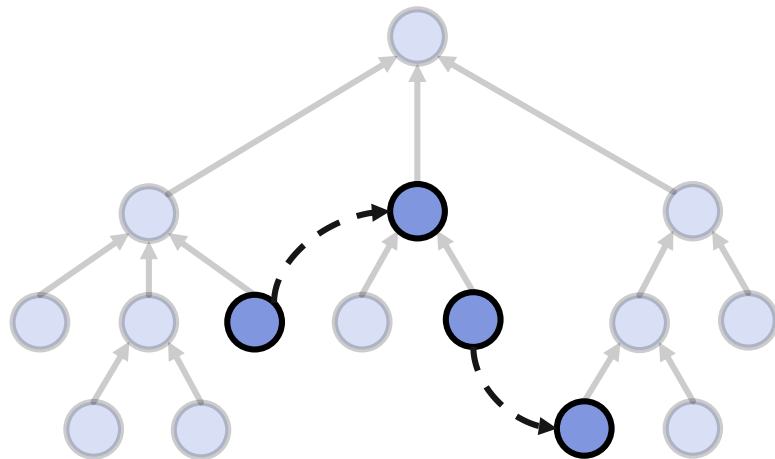
- Defining entity logic





Traditional level design at IOI

- Defining entity logic
 - Entity references wire entities together
 - Set property values
 - Register event handlers





Vision For Hitman

- Freedom of approach
 - High-definition sandbox
 - Many ways of completing a mission
 - More load-out options
 - Replayability



Vision For Hitman

- Ever expanding
 - New missions appear over time
 - More than one mission in the same location
 - Time-limited targets
 - React to community



Vision For Hitman

- Contracts mode
 - Asynchronous multiplayer
 - Players define their own missions
 - Mark target(s)
 - Define intended murder weapon
 - Planned escape route
 - Etc.
 - Increase player options



Challenges

- Static level structure
 - When a level is shipped, that's how it is
- (Almost) no sharing of set-ups across missions
 - Duplicated set-ups leading to repetitive work
 - Other systems get abused for workarounds
 - Download size



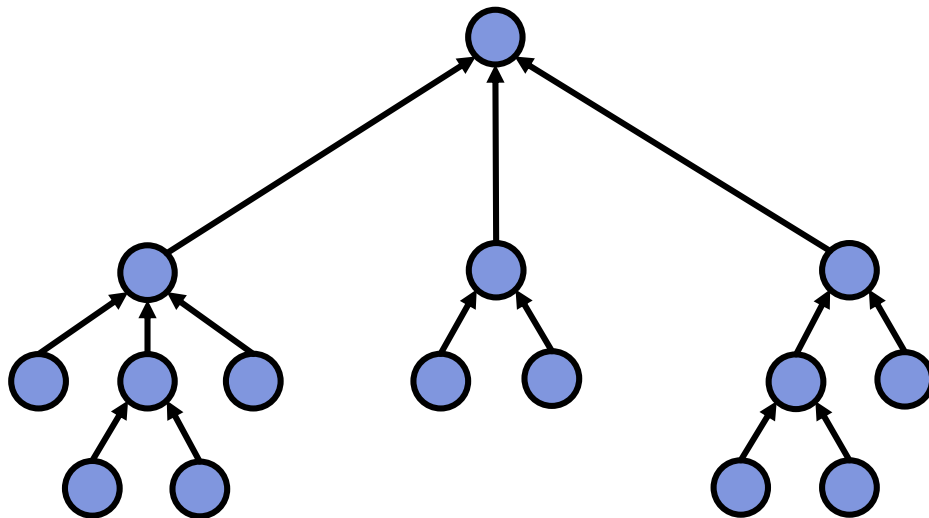
Agenda

- Motivation
- **Introducing "Bricks"**
- Impact
- Wrap Up



Introducing "Bricks"

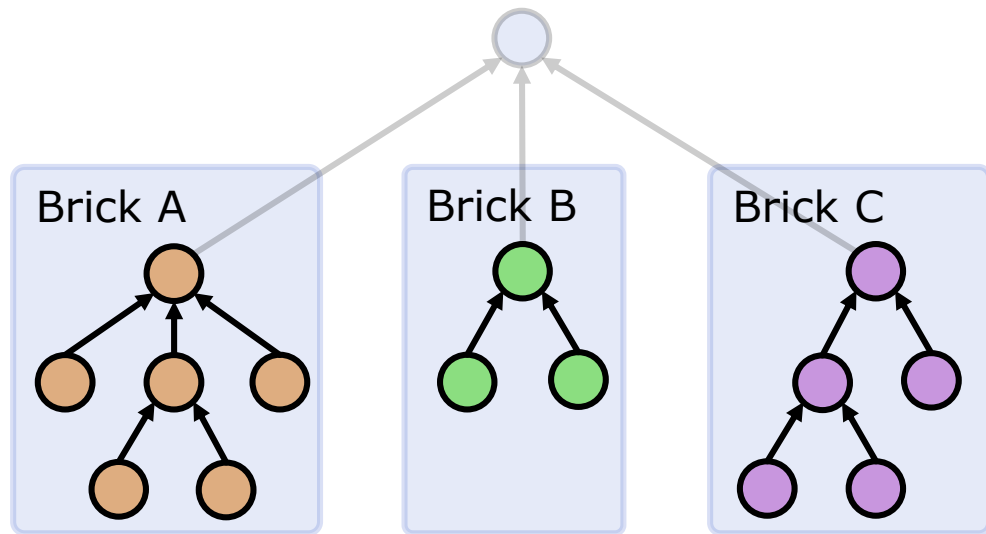
- Scenes are split into parts





Introducing "Bricks"

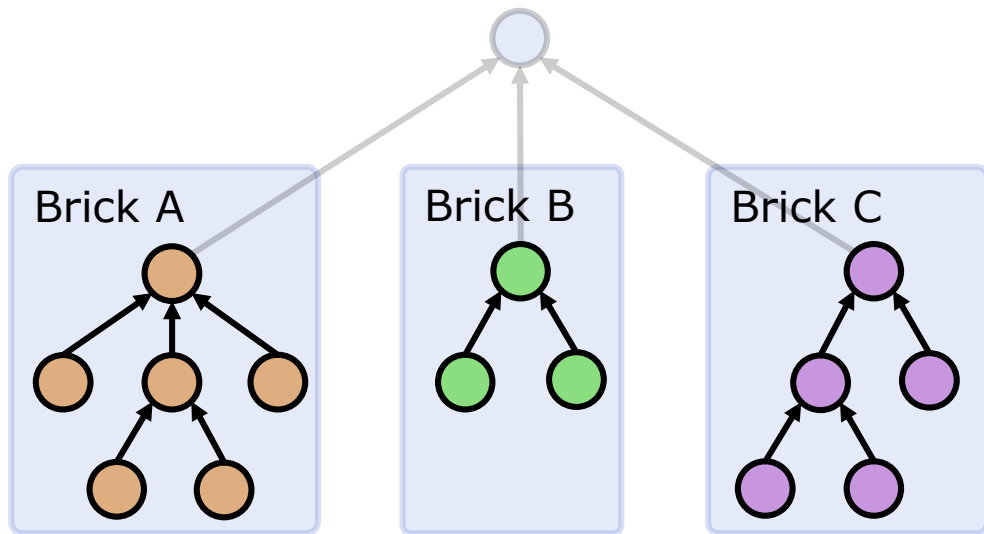
- Scenes are split into parts
 - 1 brick = 1 partial scene
 - 1 scene = N bricks





Introducing "Bricks"

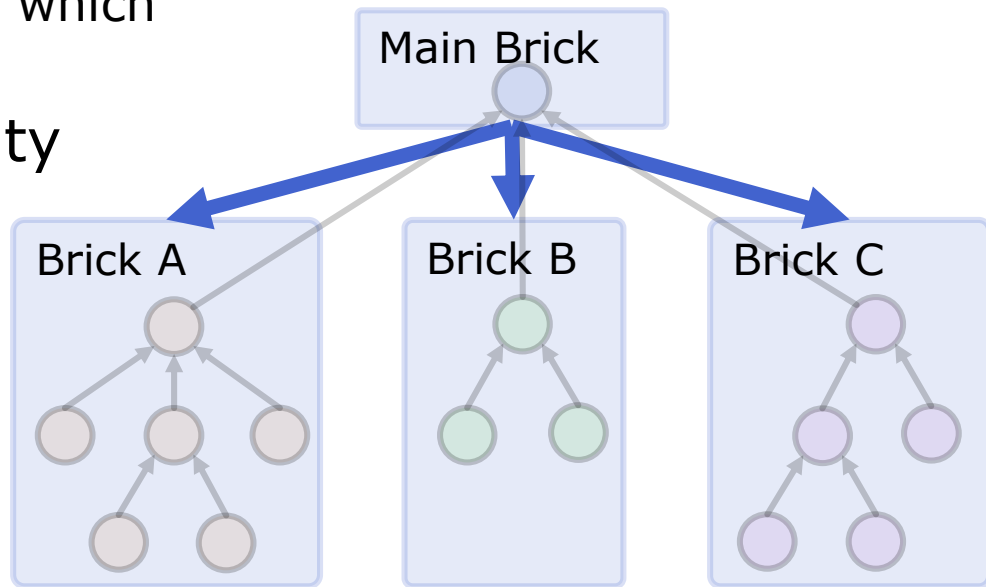
- Scene is defined by a brick





Brick Dependencies

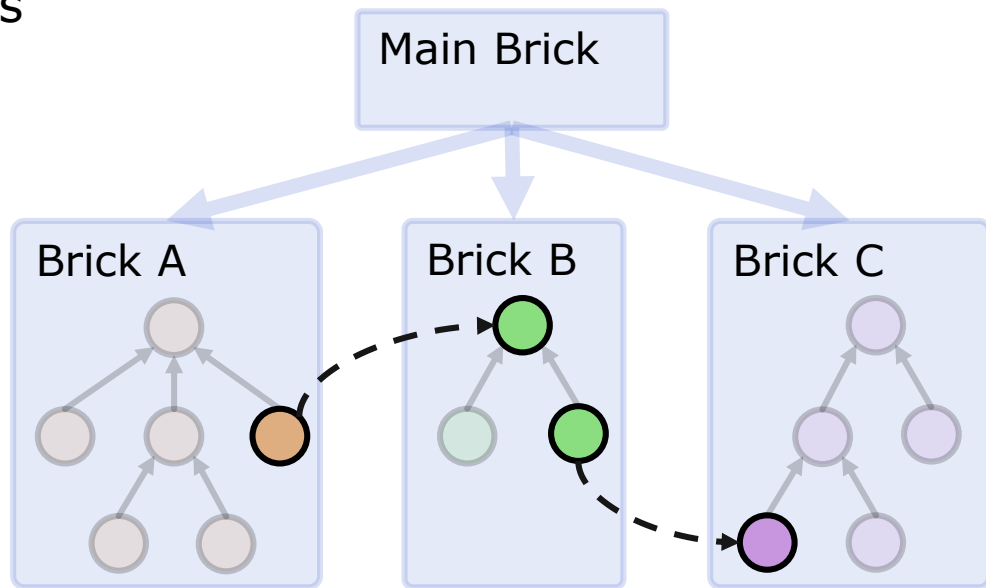
- Scene is defined by a brick
 - Brick references define which bricks are loaded
- Backwards compatibility
 - Old scenes are bricks with no other references





Brick Dependencies

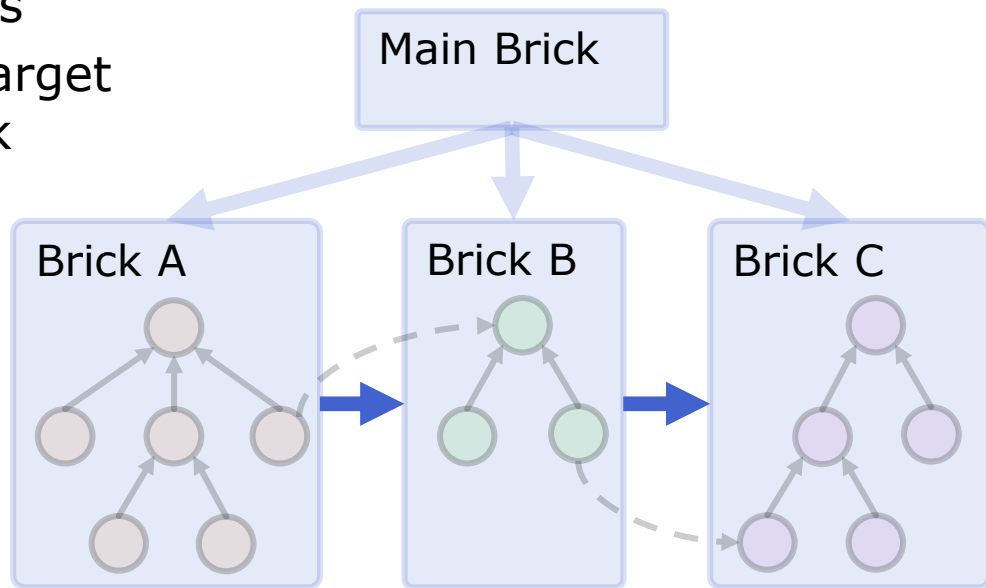
- Entity references
 - Entities have unique ids





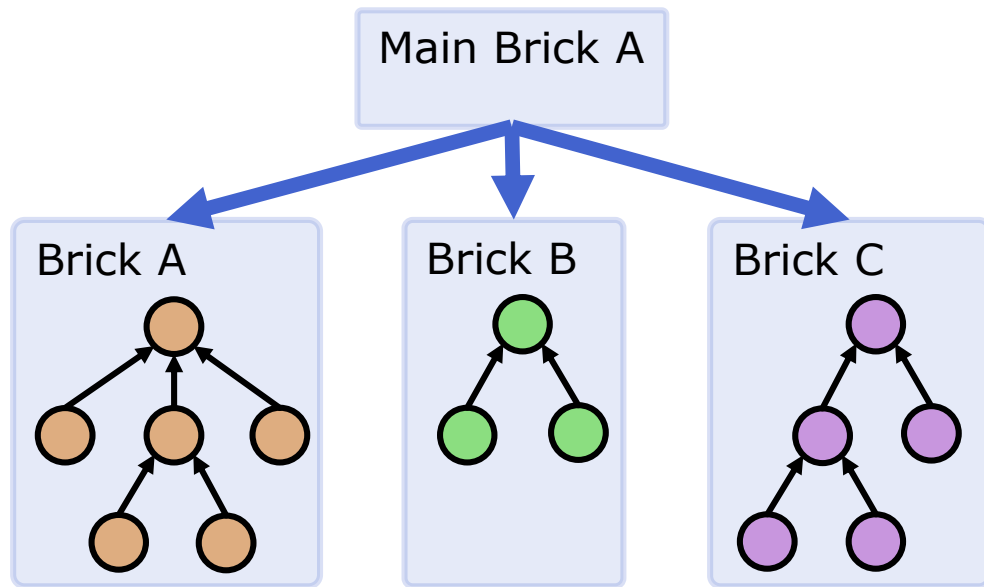
Brick Dependencies

- Entity references
 - Entities have unique ids
 - Ensure availability of target entities by adding brick references





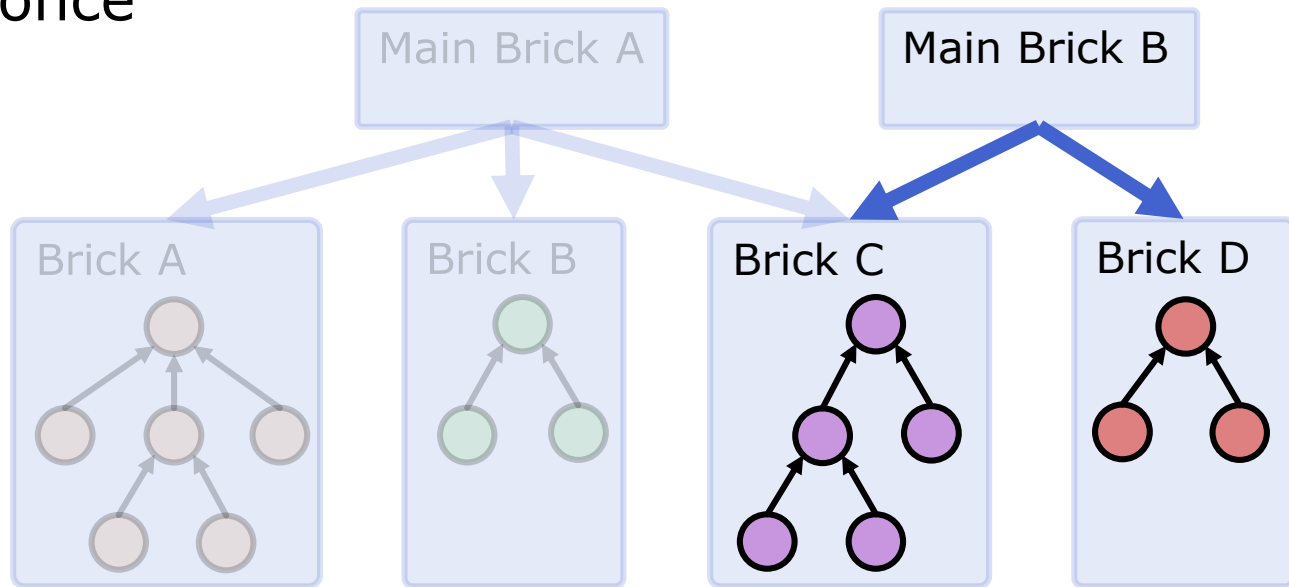
Sharing Bricks





Sharing Bricks

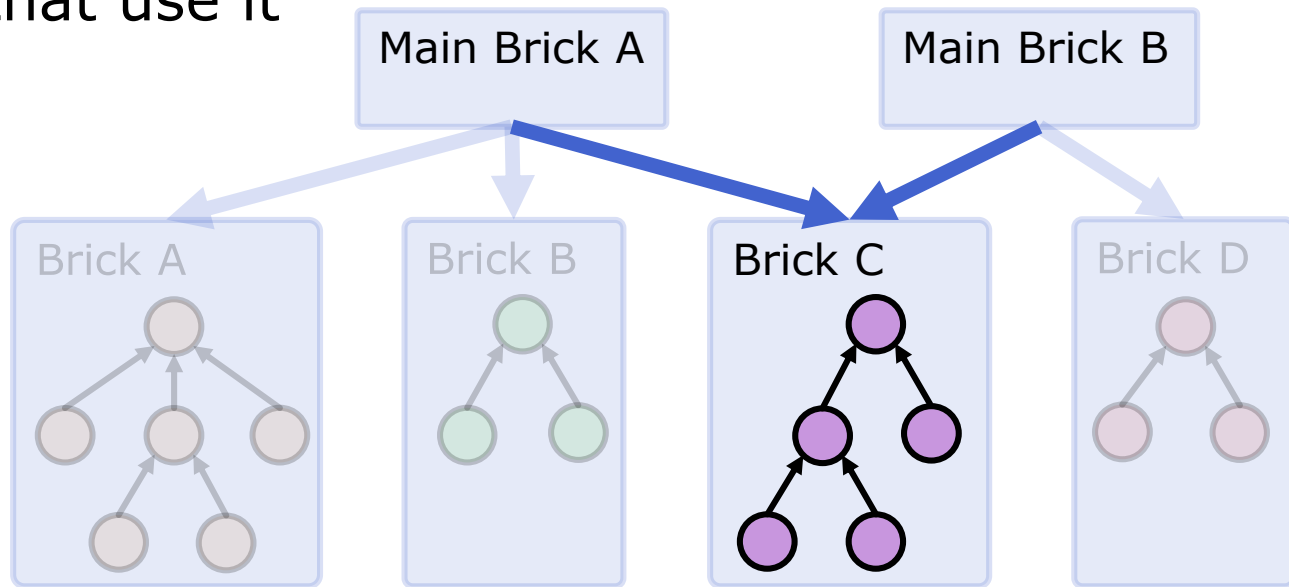
- Bricks can be referenced more than once





Sharing Bricks

- Changes to a brick affect all scenes that use it





Sharing Bricks

- Use bricks as building blocks
 - Reduces duplicated set-ups
 - Reduces repetitive bug-fixing
- Find the right granularity



Library bricks

- Generic logic/content
- Can be used in all missions

Agent 47

In-game menu

Core elements

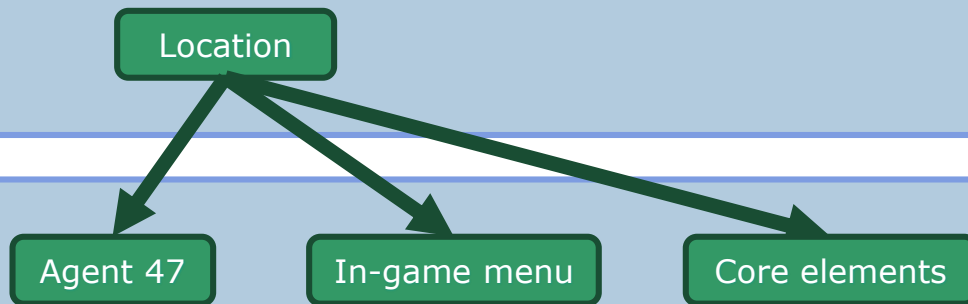


Location bricks

- Basic environment geometry
- Max. 1 per mission

Library bricks

- Generic logic/content
- Can be used in all missions





Ambient bricks

- Mission-agnostic set-ups
- Location-specific

NPC
set-up

NPC
set-up

Location bricks

- Basic environment geometry
- Max. 1 per mission

Location

Library bricks

- Generic logic/content
- Can be used in all missions

Agent 47

In-game menu

Core elements



Mission bricks

- Mission-specific set-ups
- Mission definition

Mission

Ambient bricks

- Mission-agnostic set-ups
- Location-specific

NPC
set-up

NPC
set-up

Location bricks

- Basic environment geometry
- Max. 1 per mission

Location

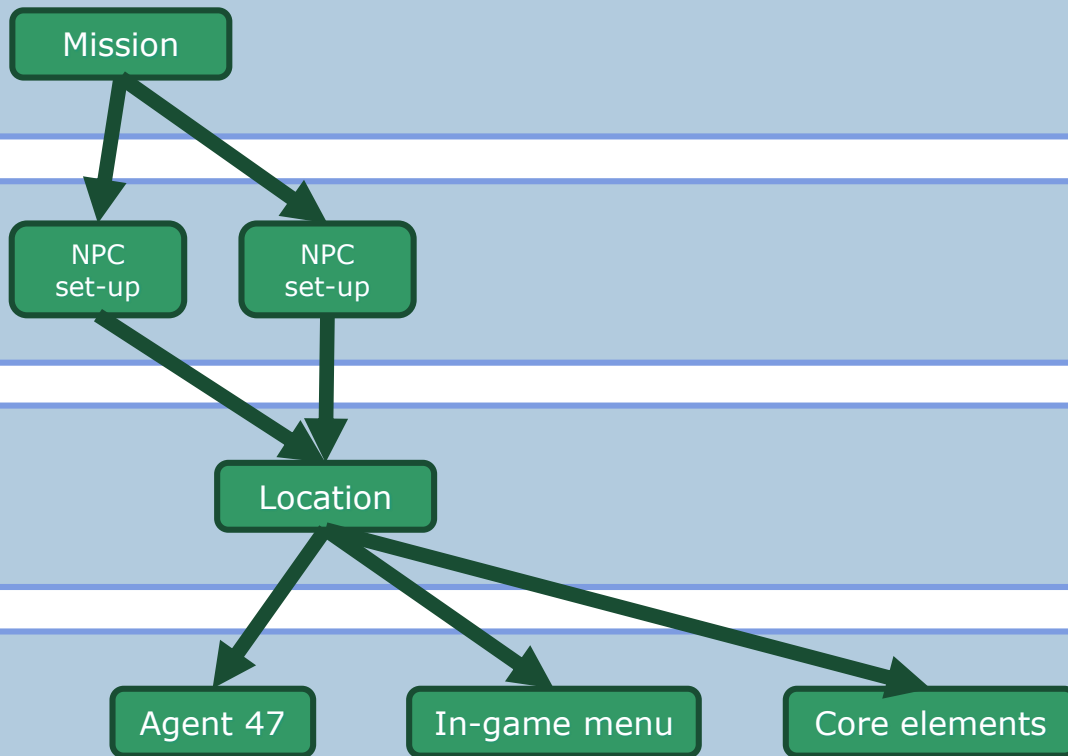
Library bricks

- Generic logic/content
- Can be used in all missions

Agent 47

In-game menu

Core elements





Mission bricks

- Mission-specific set-ups
- Mission definition

Mission

Mission

Ambient bricks

- Mission-agnostic set-ups
- Location-specific

NPC set-up

NPC set-up

NPC set-up

Location bricks

- Basic environment geometry
- Max. 1 per mission

Location

Location

Library bricks

- Generic logic/content
- Can be used in all missions

Agent 47

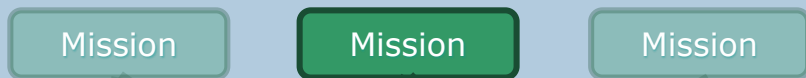
In-game menu

Core elements



Mission bricks

- Mission-specific set-ups
- Mission definition



Ambient bricks

- Mission-agnostic set-ups
- Location-specific



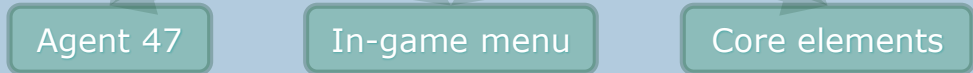
Location bricks

- Basic environment geometry
- Max. 1 per mission



Library bricks

- Generic logic/content
- Can be used in all missions





Mission bricks

- Mission-specific set-ups
- Mission definition



Ambient bricks

- Mission-agnostic set-ups
- Location-specific



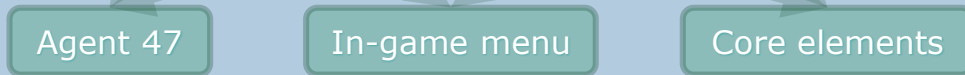
Location bricks

- Basic environment geometry
- Max. 1 per mission



Library bricks

- Generic logic/content
- Can be used in all missions



Location brick



Location brick + ambient brick(s)





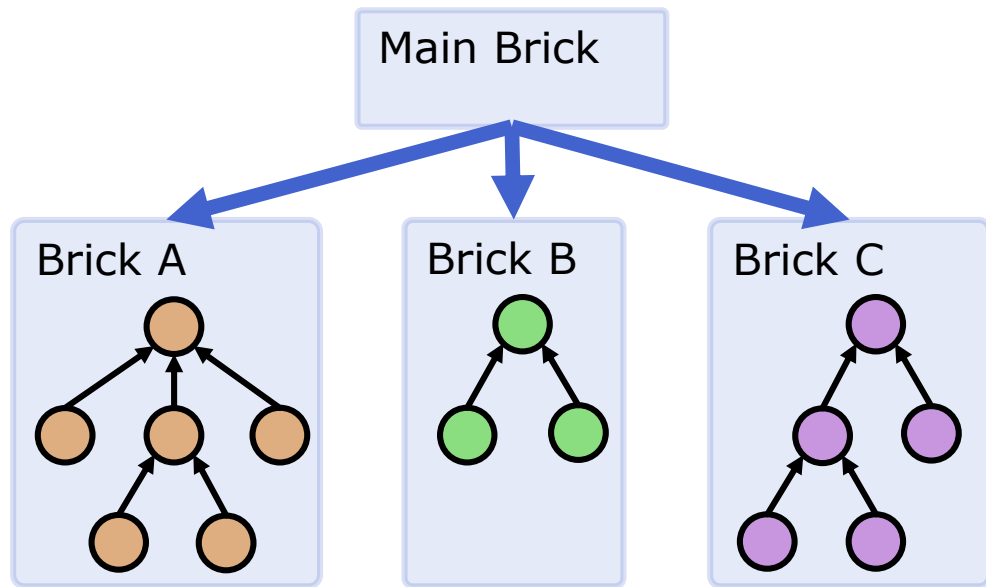
This is really great... but....

- Tree hierarchy is important
 - For humans
 - For gameplay logic
- New requirement:
 - Inject entities anywhere in the scene tree



Extending a Tree Hierarchy

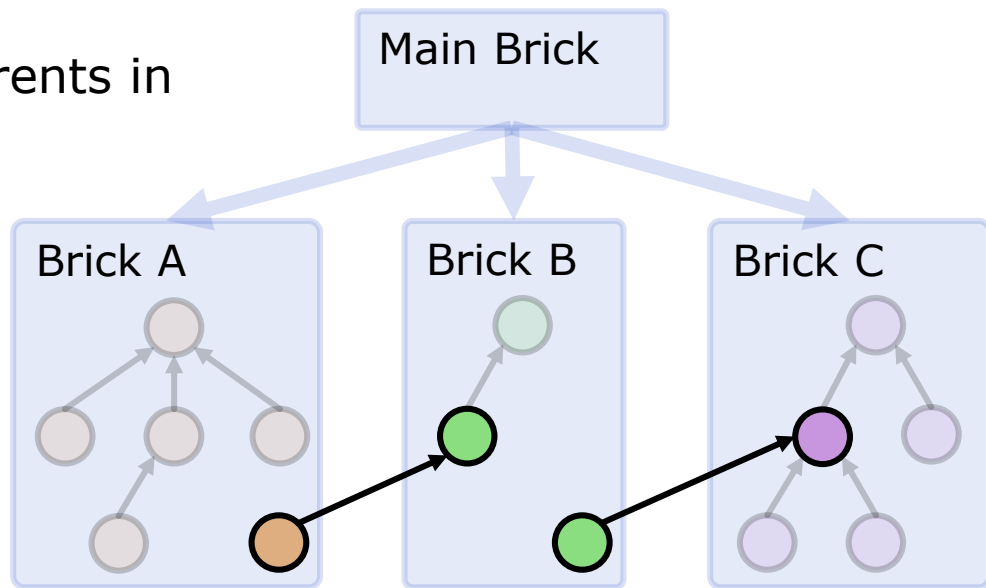
- Parent relations are similar to entity references





Extending a Tree Hierarchy

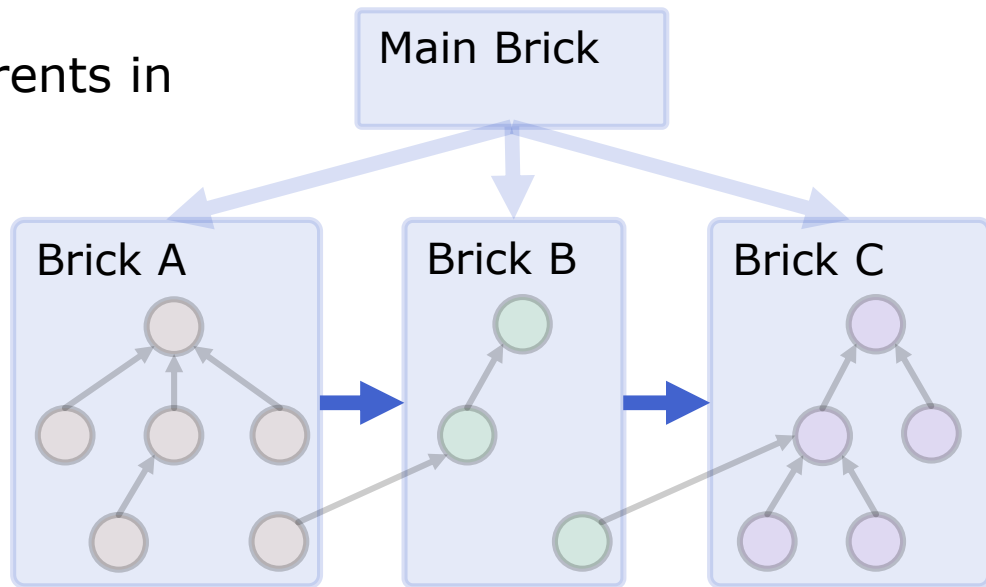
- Parent relations are similar to entity references
 - Entities can refer to parents in other bricks





Extending a Tree Hierarchy

- Parent relations are similar to entity references
 - Entities can refer to parents in other bricks
 - Ensure availability of parents by adding brick references



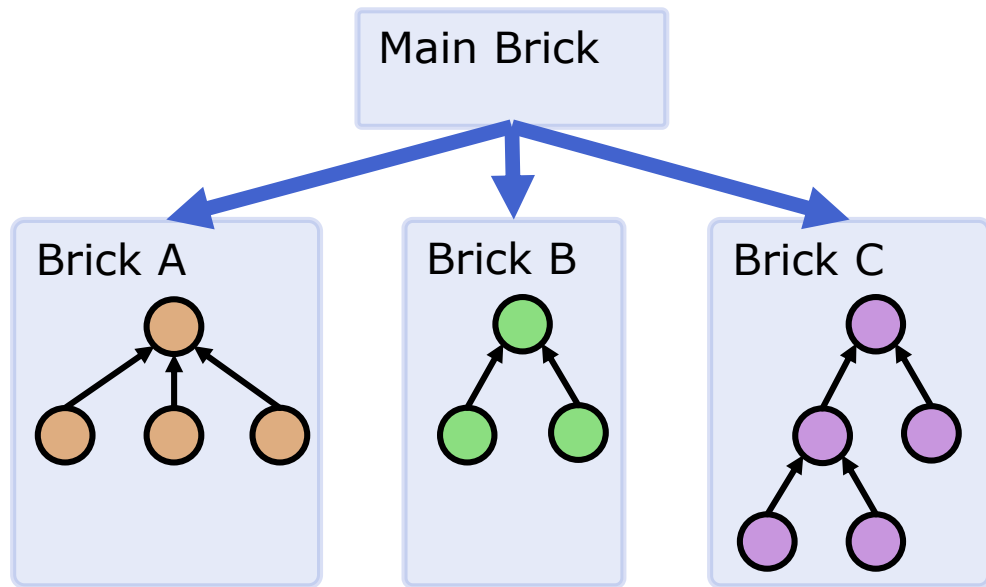


Now we can almost use the brick system... but...

- Properties are too static
- New requirement:
 - Override properties in other bricks



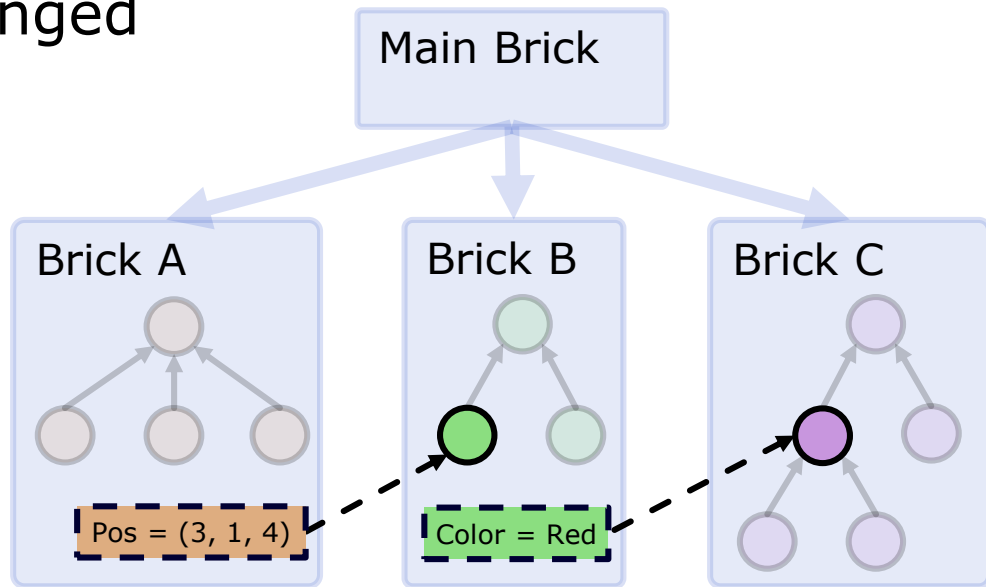
Property Overrides





Property Overrides

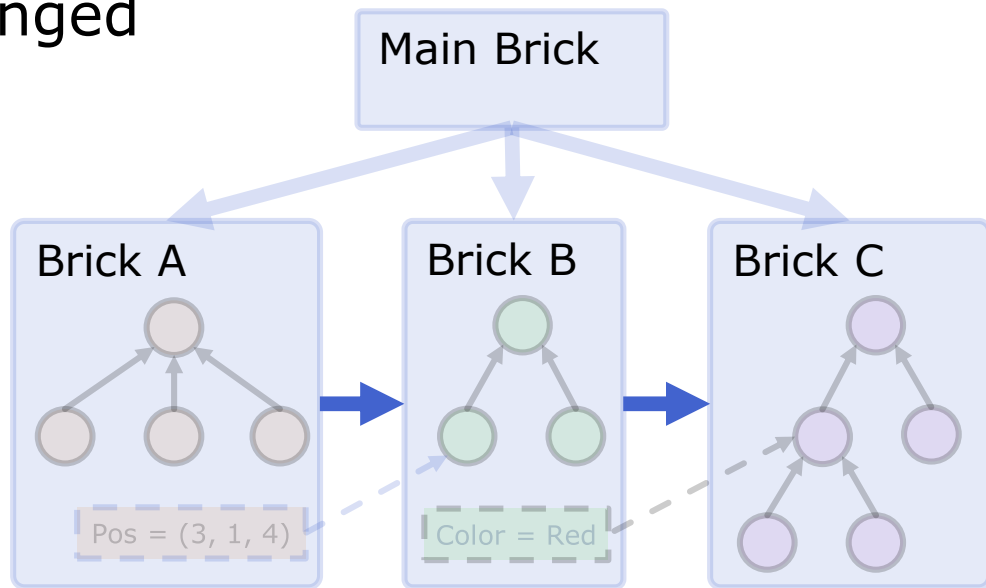
- Specify new value for target property
- Original brick is unchanged





Property Overrides

- Specify new value for target property
- Original brick is unchanged
- Ensure availability of target entity by adding brick references



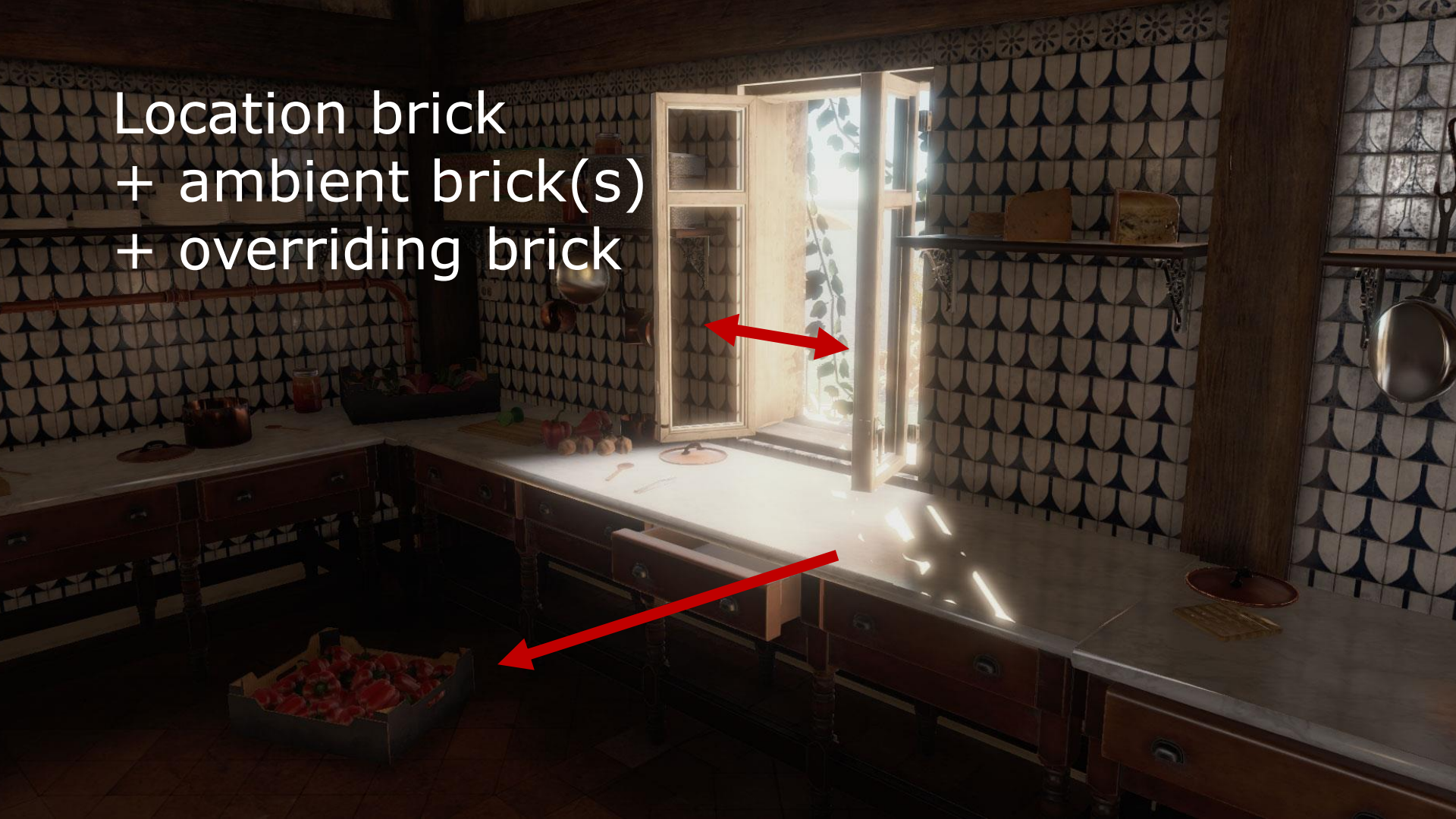
Location brick
+ ambient brick(s)



Location brick
+ ambient brick(s)



Location brick
+ ambient brick(s)
+ overriding brick



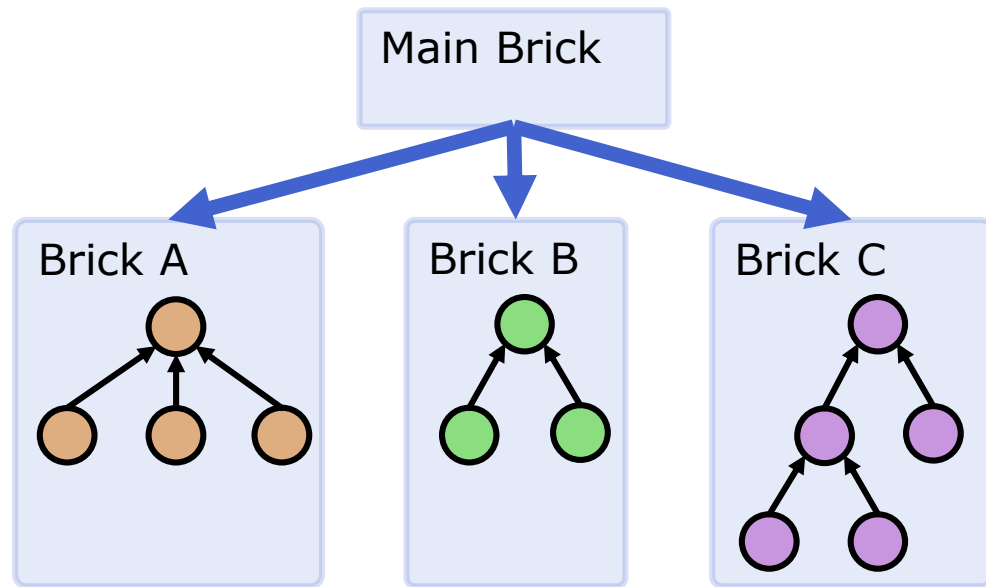


NOW we can almost use the brick system... but...

- Scene tree is too static
- New requirement:
 - Remove entities in other bricks



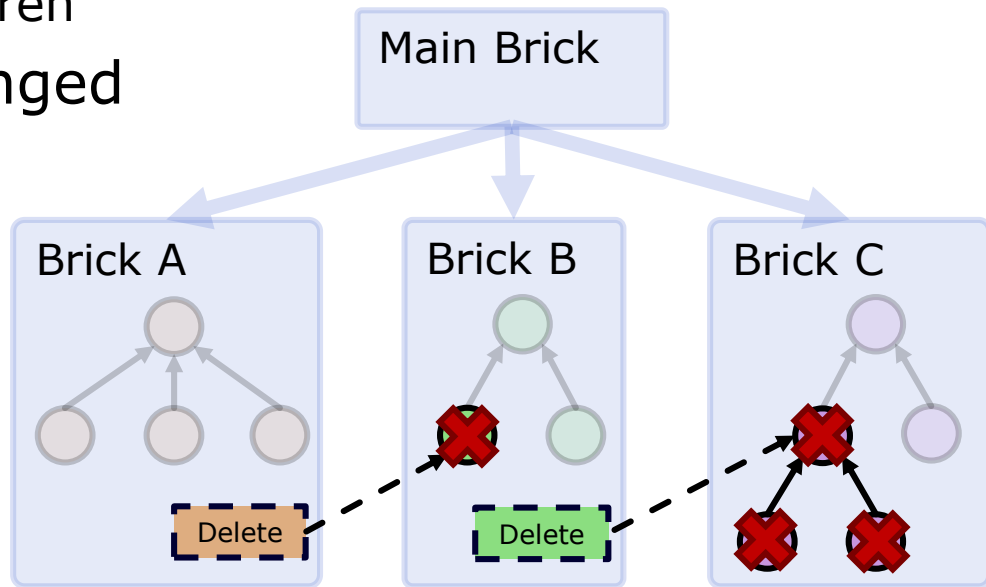
Override Deletes





Override Deletes

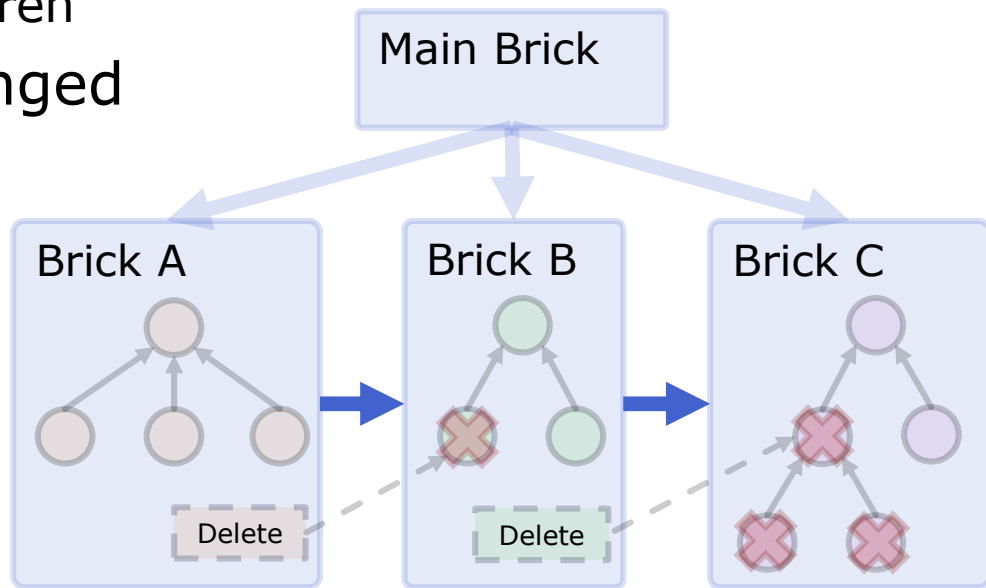
- Specify entity to remove
 - Implicitly removes children
- Original brick is unchanged





Override Deletes

- Specify entity to remove
 - Implicitly removes children
- Original brick is unchanged
- Ensure availability of entities to remove by adding brick references



Location brick



Location brick



Location brick + deleting brick



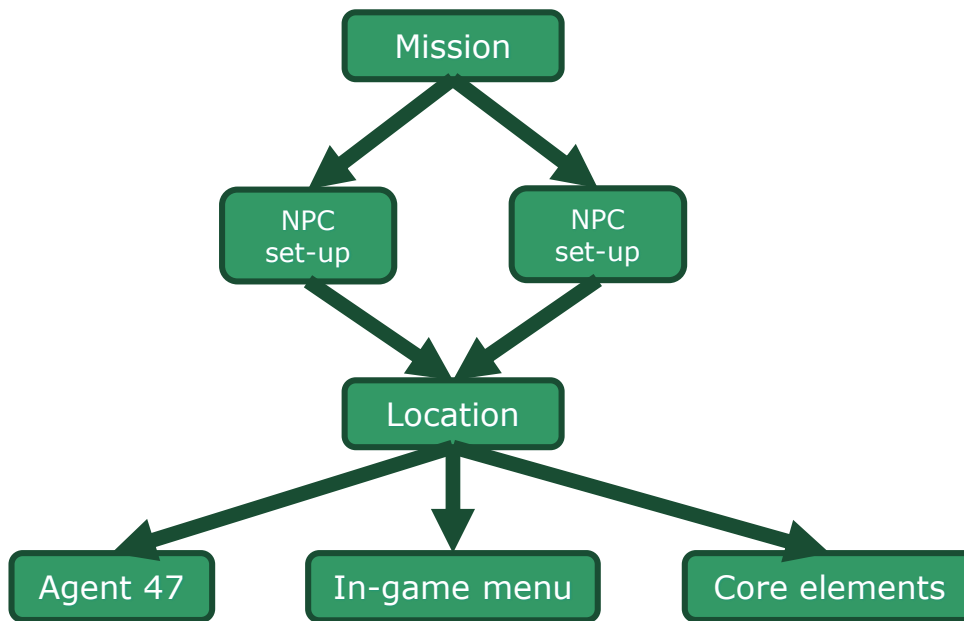


Dynamic Brick Configurations

- Dynamically change existing set-ups
- Toggled by players
 - User-configurable missions (Contracts)
- Toggled from a server
 - Seasonal content
 - Time-limited mission goals

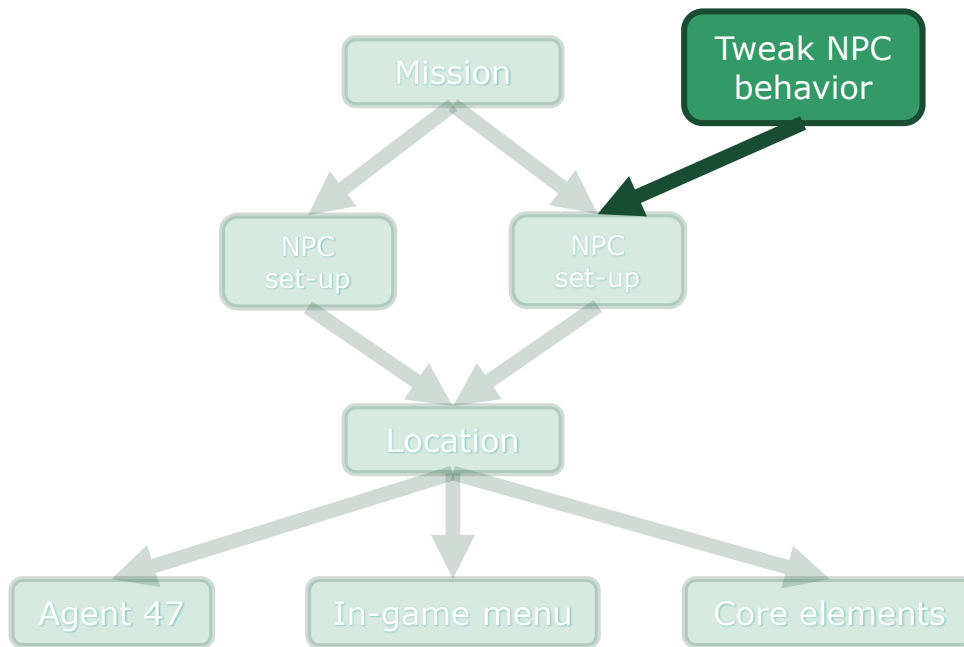


Dynamic Brick Configurations



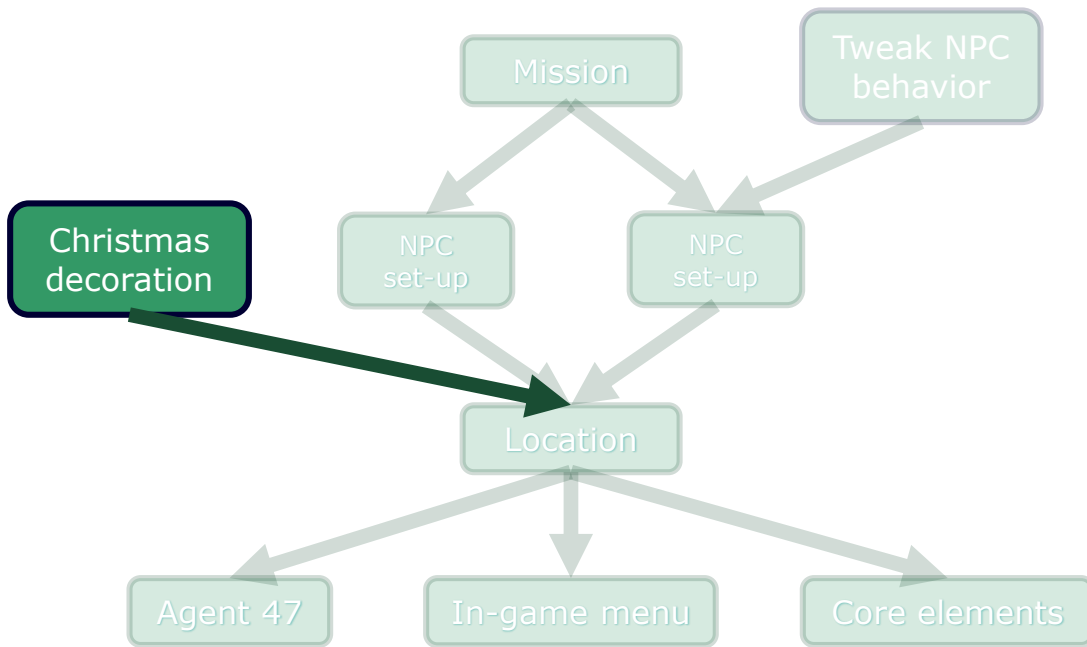


Dynamic Brick Configurations





Dynamic Brick Configurations





Overview

- Motivation
- Introducing "Bricks"
- **Impact**
- Wrap Up



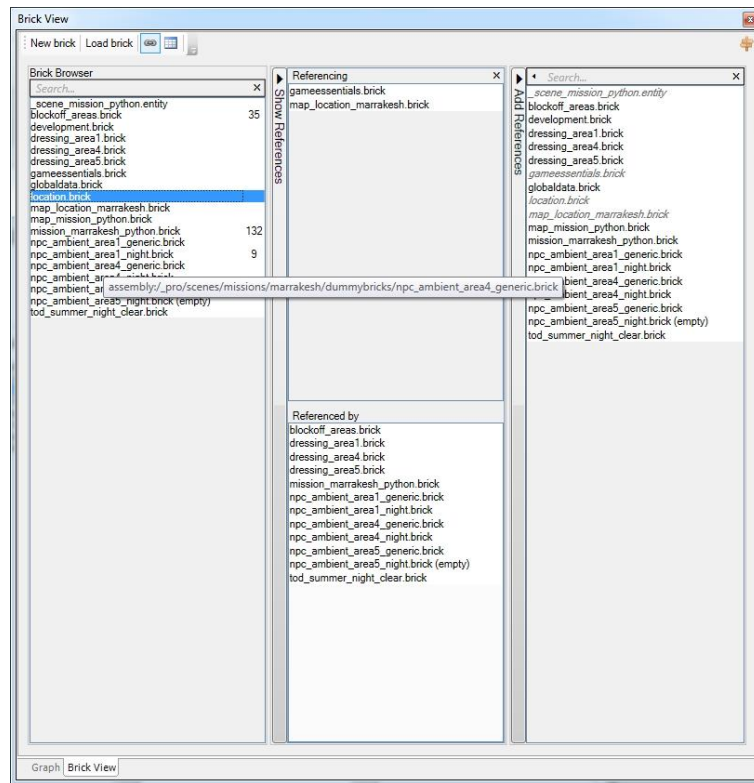
Implementation plan

- 1-2 level designers start using bricks
 - Get experience with tools and workflow
 - Discover missing tools and features
 - Iron out introductory hiccups
- Full team roll-out
 - Embraced by entire team
 - Brick tool engineers are heroes 😊



Basic Brick Management

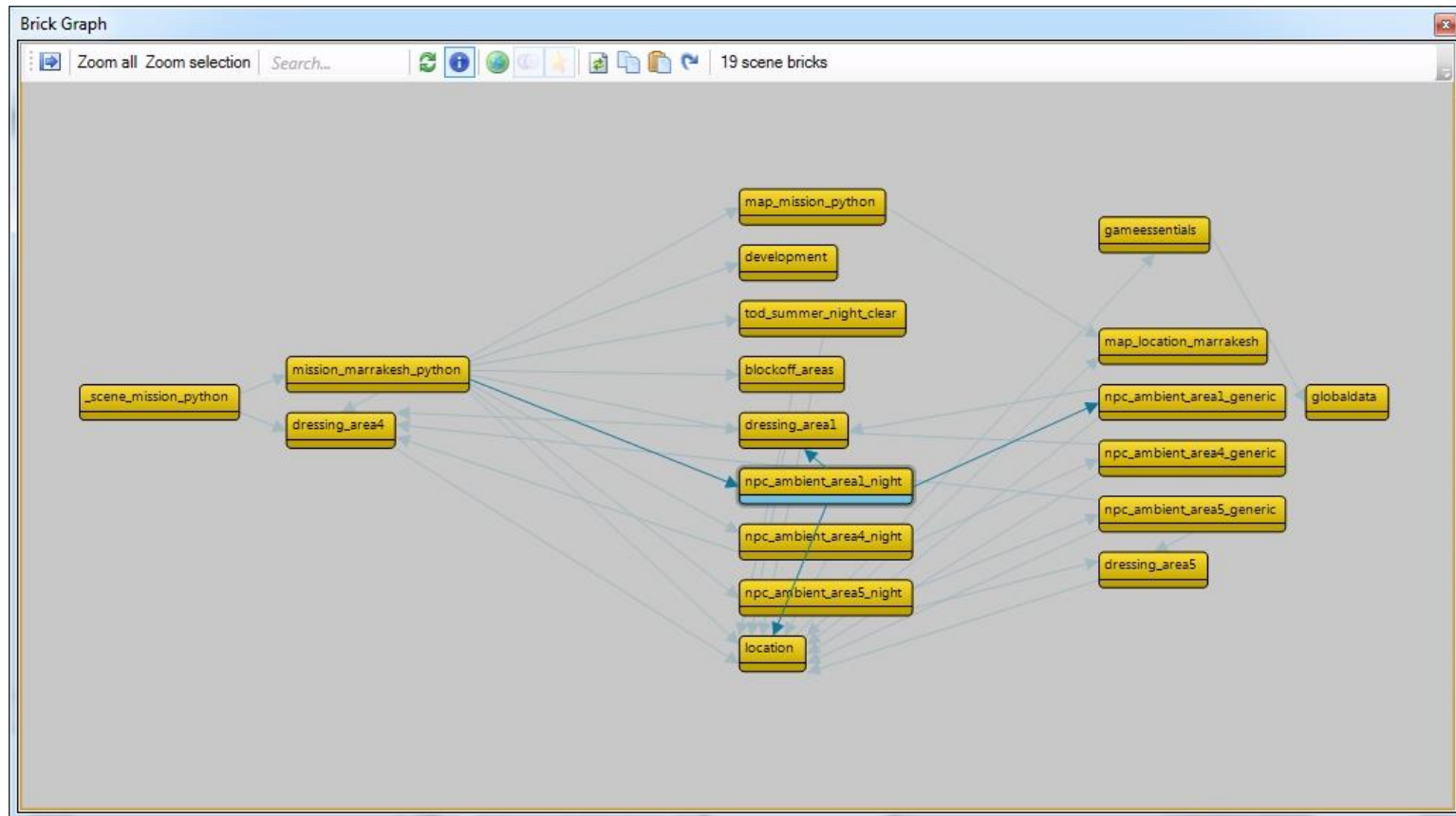
- New view: Brick View
 - Create new bricks
 - Load/unload existing bricks
 - Navigate brick references
 - Manage brick references
 - Manual process: connection by design – not chance





Basic Brick Management

- Brick references define a graph
- Navigating a graph using lists view is hard
- New view: Brick Graph





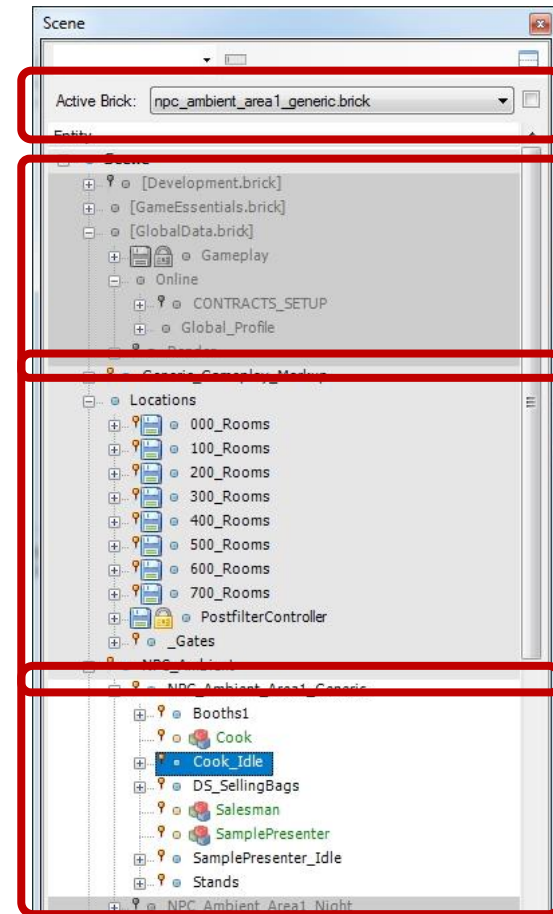
New Concept: Work Scopes

- Active brick
 - Selected by the user
 - New entities, property overrides and override deletes are added to the active brick
- Reachable bricks
 - Bricks with direct reference from the active brick
 - Entities in reachable bricks can be referenced
 - Cannot be edited
- Unreachable bricks
 - Bricks with no direct reference from the active brick
 - Entities in these bricks cannot be referenced
 - Cannot be edited



Updated View: Scene Tree

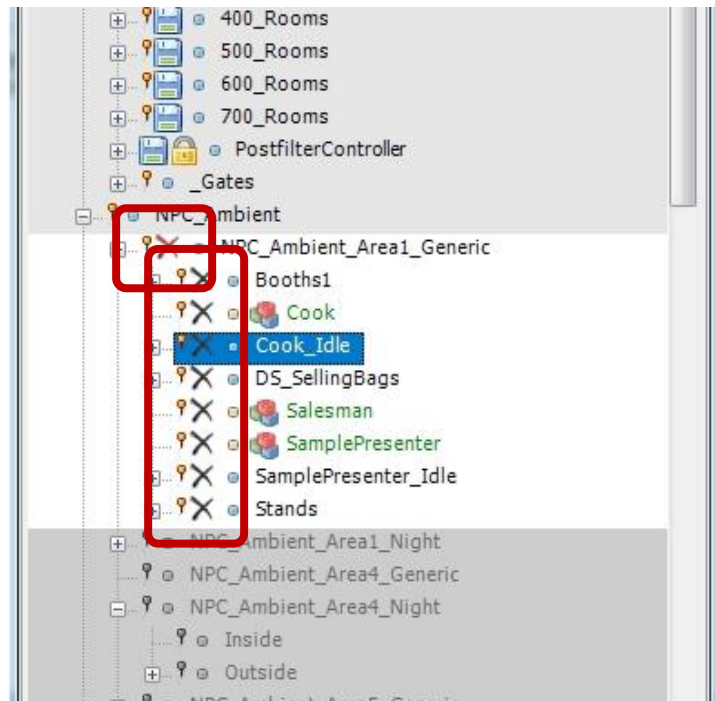
- Reflecting work scopes
 - Active brick selection
 - Entities in active brick
 - Original style
 - Reachable entities
 - Light gray background
 - Unreachable entities
 - Dark gray background
 - Gray scaled icons





Updated View: Scene Tree

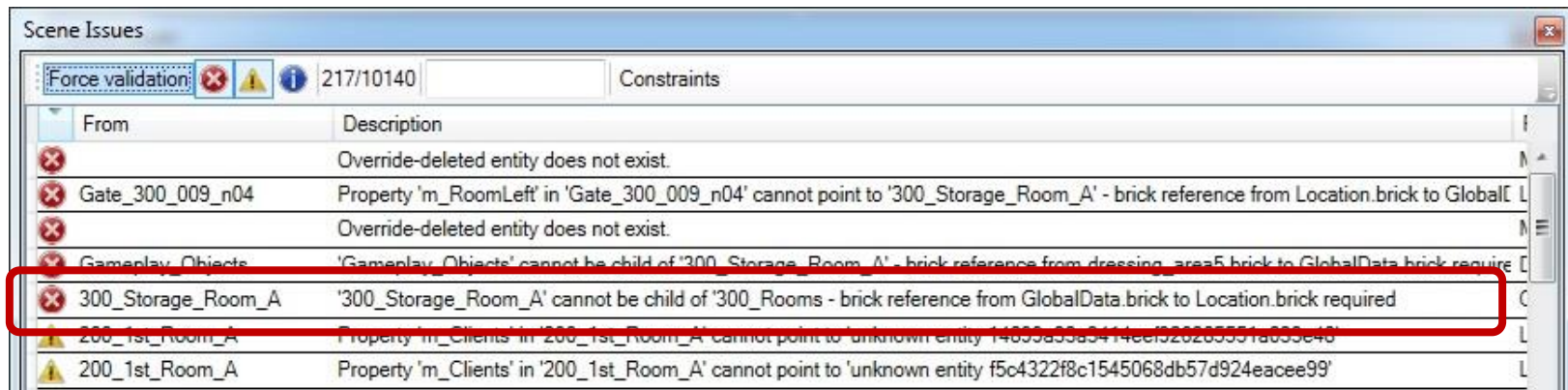
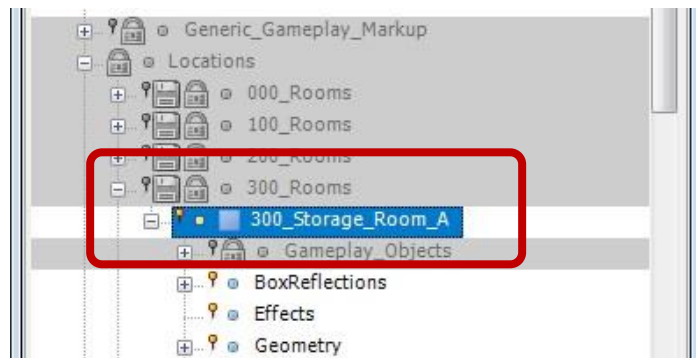
- Reflecting override deletes
 - Directly deleted entities
 - Implicitly deleted children





Centralized constraint system

- All operations constrained wrt. work scope
 - Move entities between bricks....
 - ...uh oh... disable constraints, please
- New view: Scene Issues
 - Issues related to bricks ("What did I break")
 - Bonus: Can show misc. legacy issues
 - Automatic bulk fixing of issues





Effects on Production

- Challenges
 - Cultural change in the way levels are made
 - Added complexity to level design
 - Harder to get an overview
 - Initial level construction is slower



Effects on Production

- Benefits
 - Production speed improves as the brick toolbox grows
 - Making level-variations is much faster
 - Prototyping is easier
 - Allows for set-ups that were not possible before



Overview

- Motivation
- Introducing "Bricks"
- Impact
- Wrap Up



Credits

- Tobias Sicheritz, Mi'pu'mi Games
- Kasper Fauerby
- Søren Seeberg
- Torbjørn V. Christensen



Q&A

?