Procedural Parkour

By:
Devin, Eli, Diwas and Kai

Source & Inspiration

Challenges

Main Challenges & Sections:

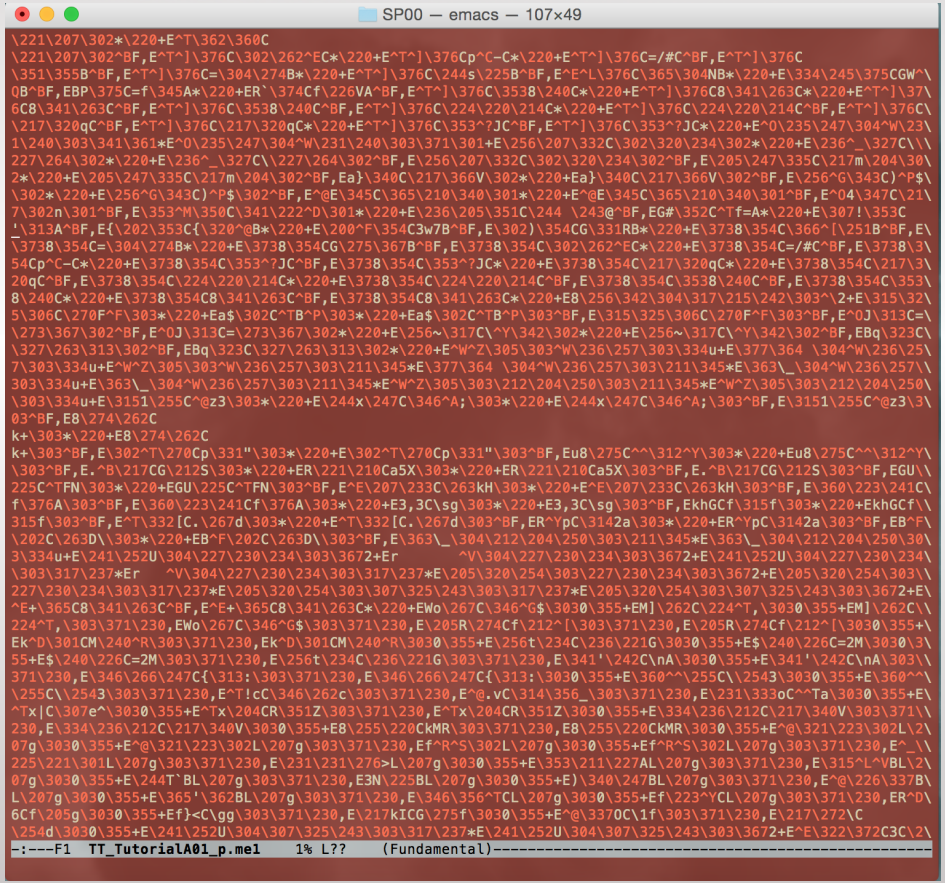

Modding Mirror's Edge


Creating Procedural Content


Generating Game Levels

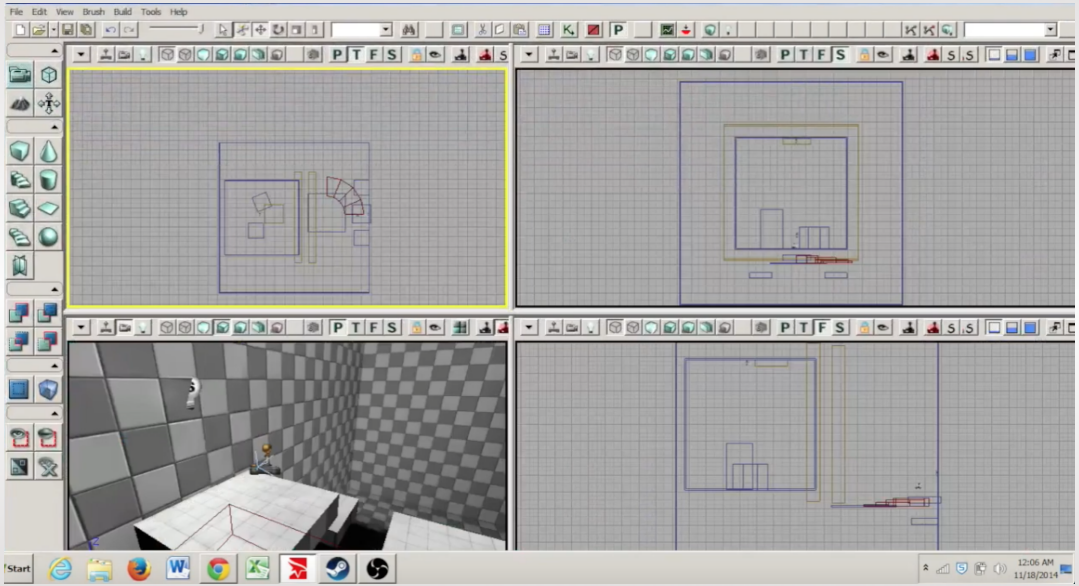
Challenge One: Modding Mirror's Edge

Mirror's Edge map files (.me1) are saved in binary format



Consulting the Modding Community

From online modding sites we found a method to open Mirror's Edge maps via an editor found in America's Army 3.



Early Level Editing Success

A simple level built by hand in order to understand how the editor works.

Procedural Content

We have now mastered the basic features of the editor, but we need to find a way to interface with the data that we procedurally generate.

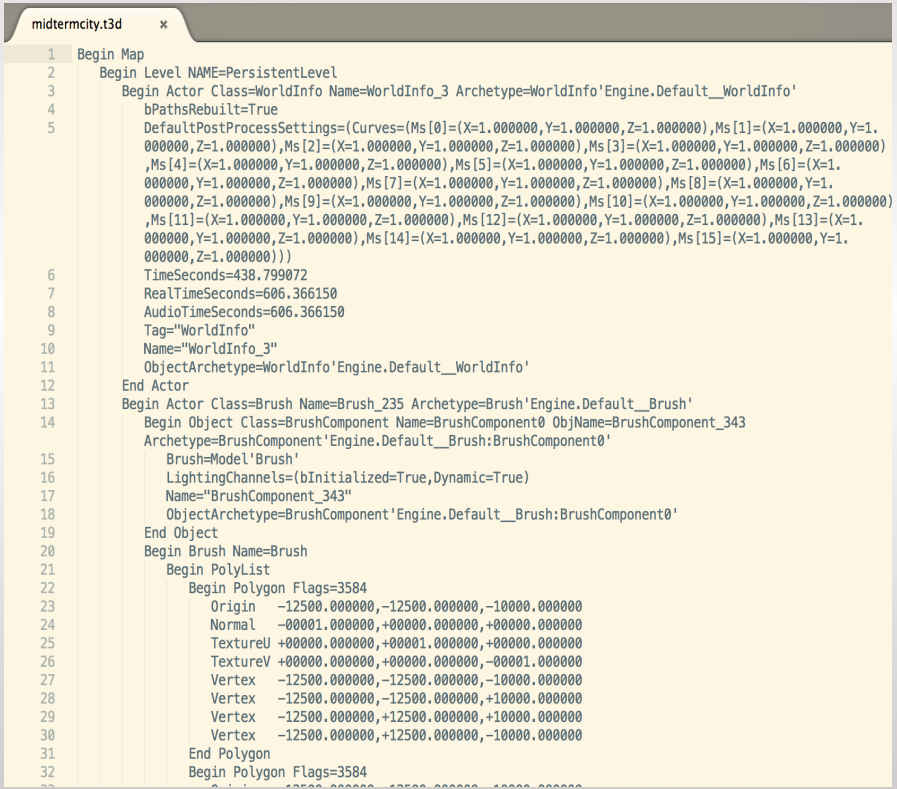
- .obj import tools are unruly and don’t include materials
- .me1(.umap) files are entirely undocumented
- We have to procedurally generate a more flexible intermediate file

T3D Files

T3D files are the answer.
T3D is the Unreal Text File.
Extremely readable

And most importantly, T3D files are importable and exportable from the editor.

Map->Actor(Brush)->Object.
Lots of redundant information

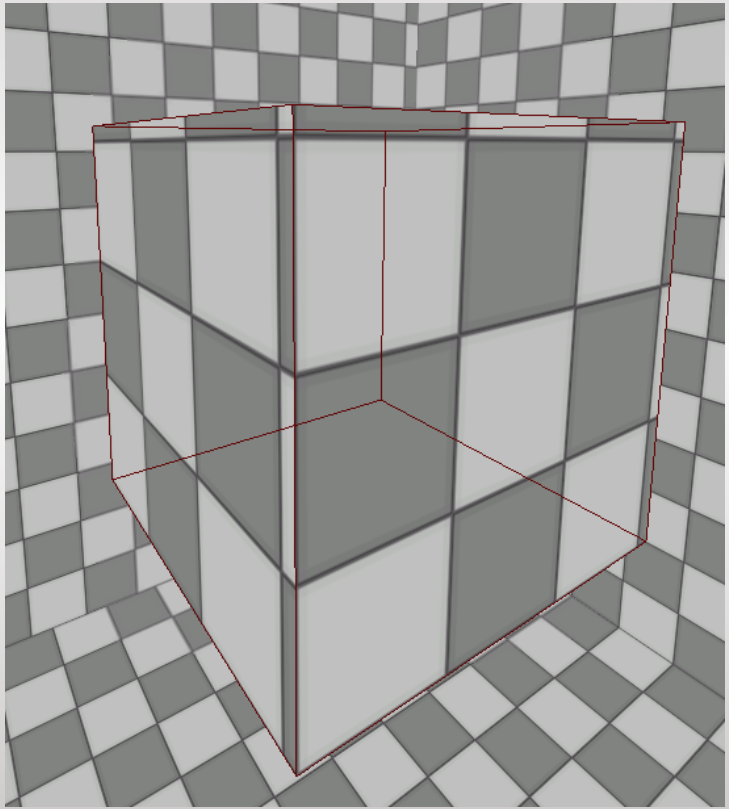


Basic Cube

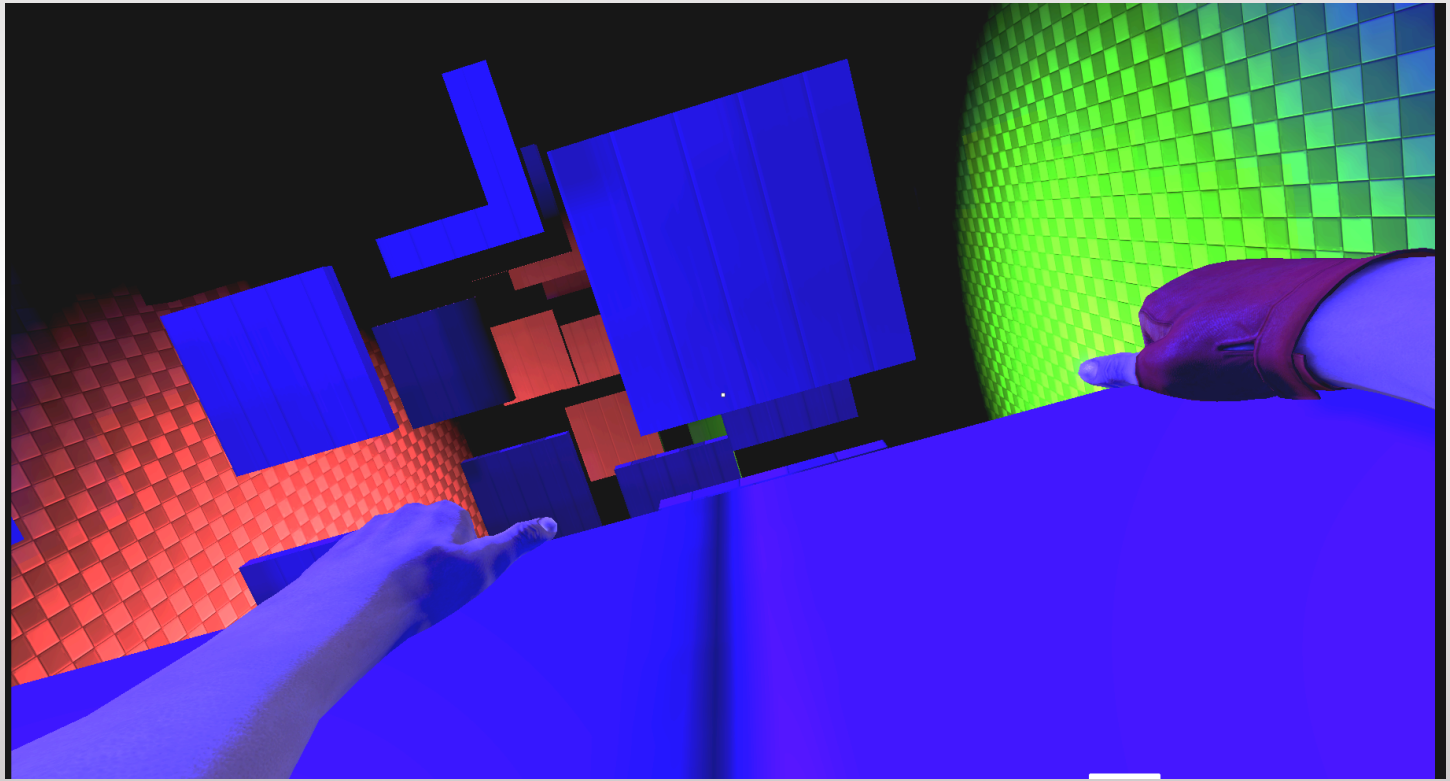
Two ways we can define a cube:
- Brush & CSG_Add
- Static Mesh

With only cubes, we can define:
- Platforms
- Wall-runs
- Springboards
- And MORE!
-

Unique Conventions:
- Without Vertex Indexing
- Vertical Z-axis



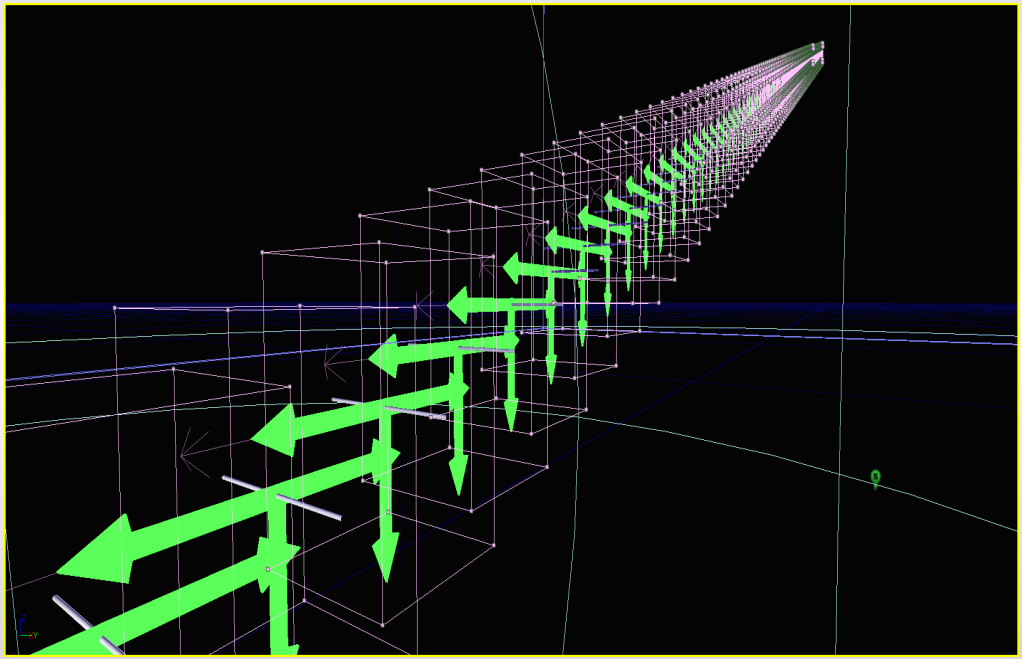
First Procedural Level



Volumes

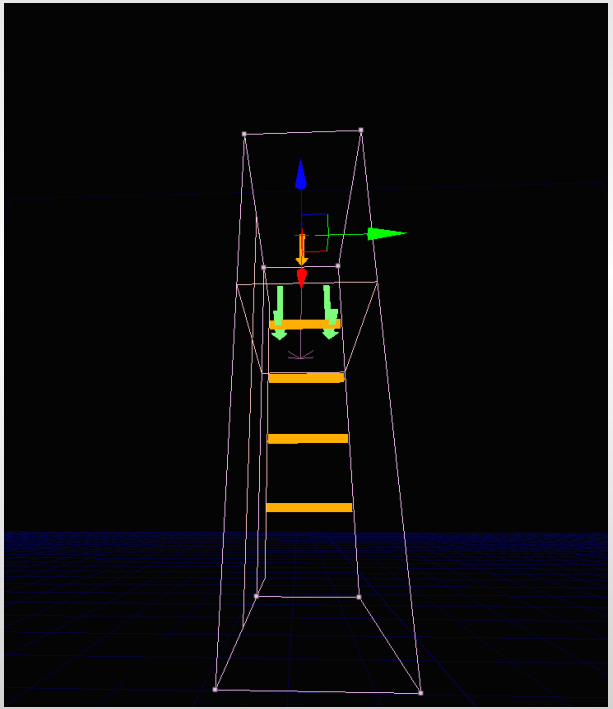
- Invisible meshes with triggers
- Transformation affects behavior

<u>Examples Include:</u>
- Swing, Ledgewalk, Zipline, Balance Walk, Ladder, etc.

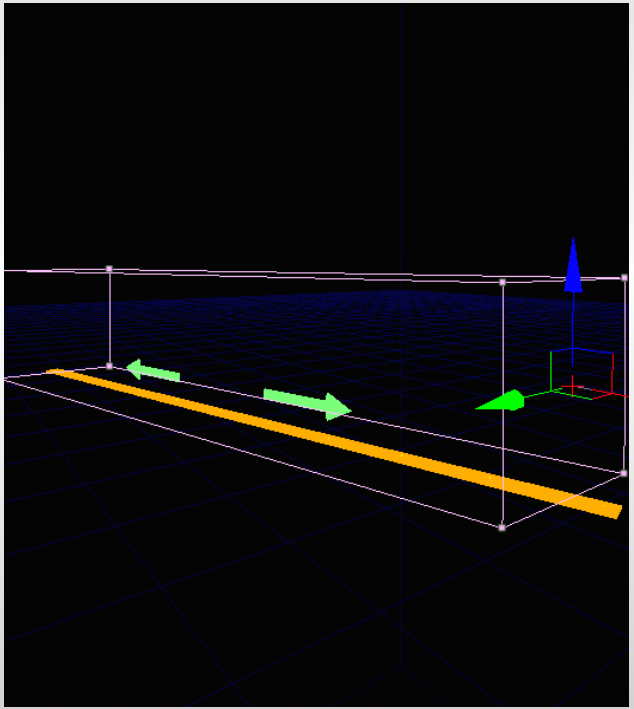


Swing Volume

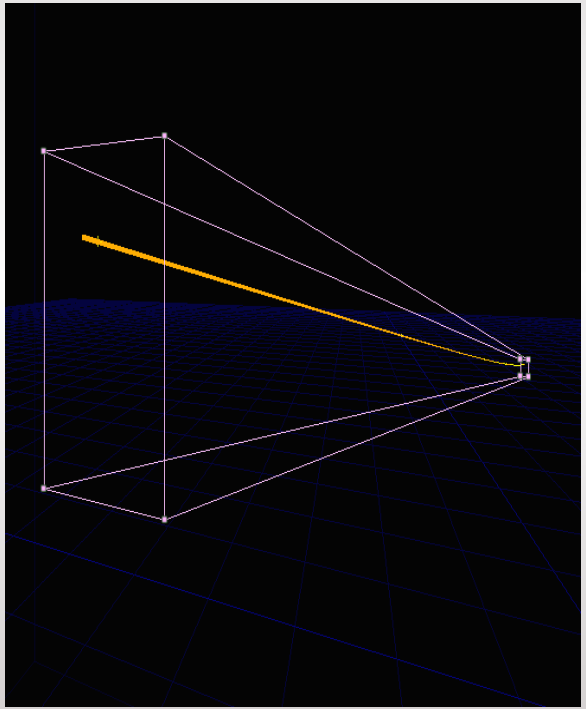
Volume Examples



Ladder



Balance Walk



Zipline

Content Creation Pipeline

Based on ObjectManipulator from DK3TM

↓

Add(Cube, Swing, Zipline…. etc.)

↓

Intermediate Storage

↓

T3D File (Irreversable)

↓

Unreal Editor

↓

Game

Challenge Roadmap

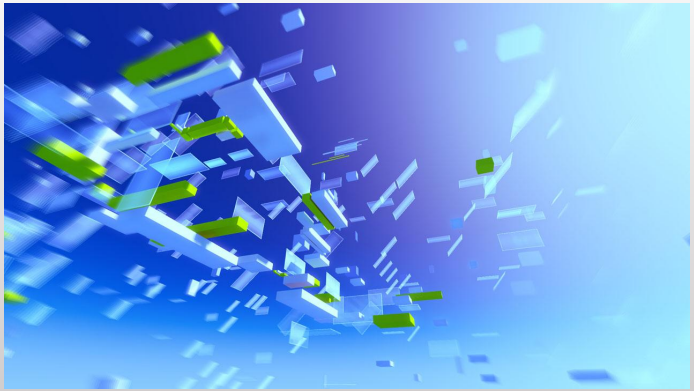
Modding Mirror's Edge

Creating Procedural Content

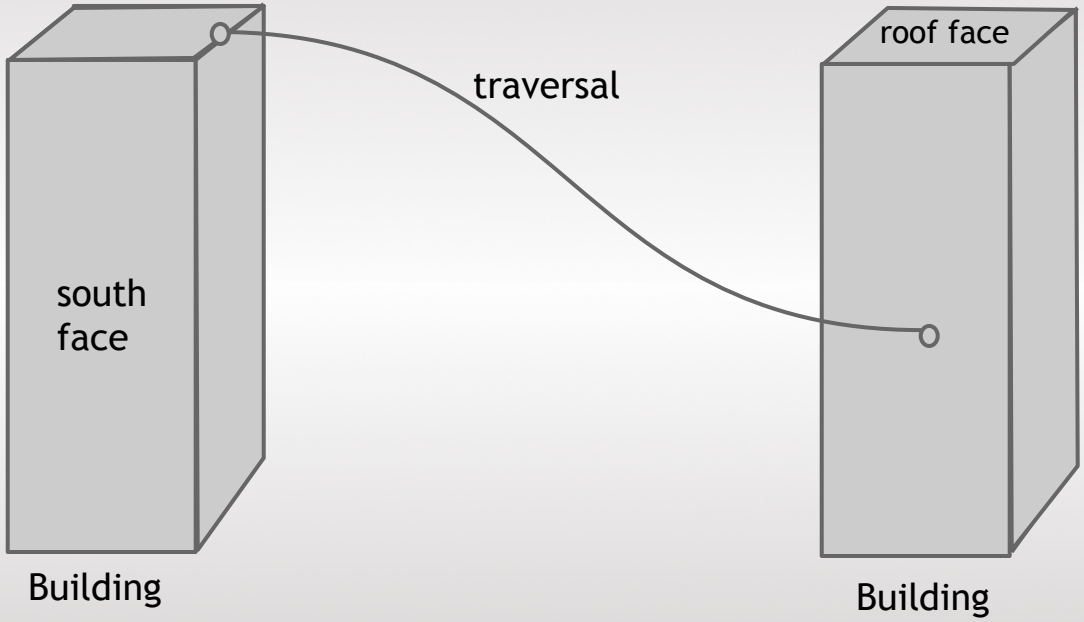
Generating Game Levels

Challenge Two: Creating Content

- Mirror's Edge DLC has an impressive free roam mode constructed with basic shapes
- We wanted to make something that would combine DLC-style gameplay with the sprawling cities of Mirror's Edge



Our Procedural Architecture: A Visual



Level:
- Buildings
  - Faces
    - Traversals

The natural question now is: “Given a set of buildings, what traversals are possible between their faces?”

Our Procedural Architecture

Steps in the process:

- compute each building's neighbors
  - ray casting over a voxel representation of our level
- determine possible traversals for each face
  - pick viable faces from neighbors
  - will be updated upon adding each new traversal
- pick traversals to guarantee playability
  - level-order traversal from a starting roof.

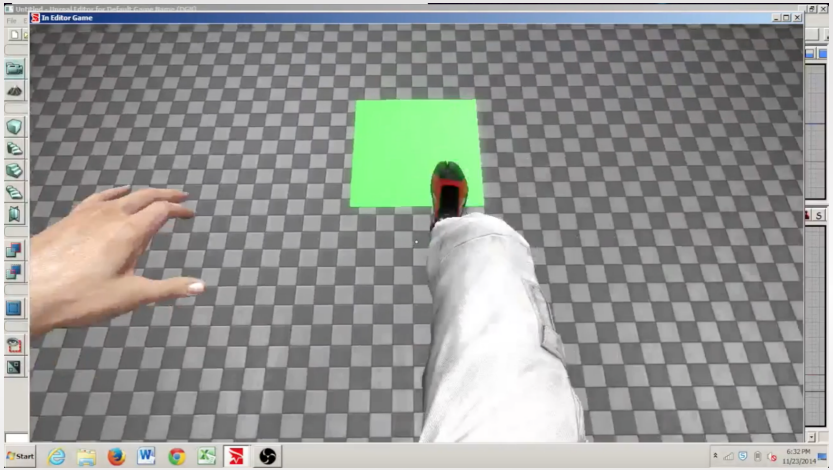
Path Generation

The next step in our algorithm is to generate a path for each traversal.

What are the core problems involved?

- Procedurally generate a playable path from Point A to Point B
- Include random features to provide interesting gameplay with variety
- Push the user to their limits without making traversal impossible

Path Generation: Reachability Constants



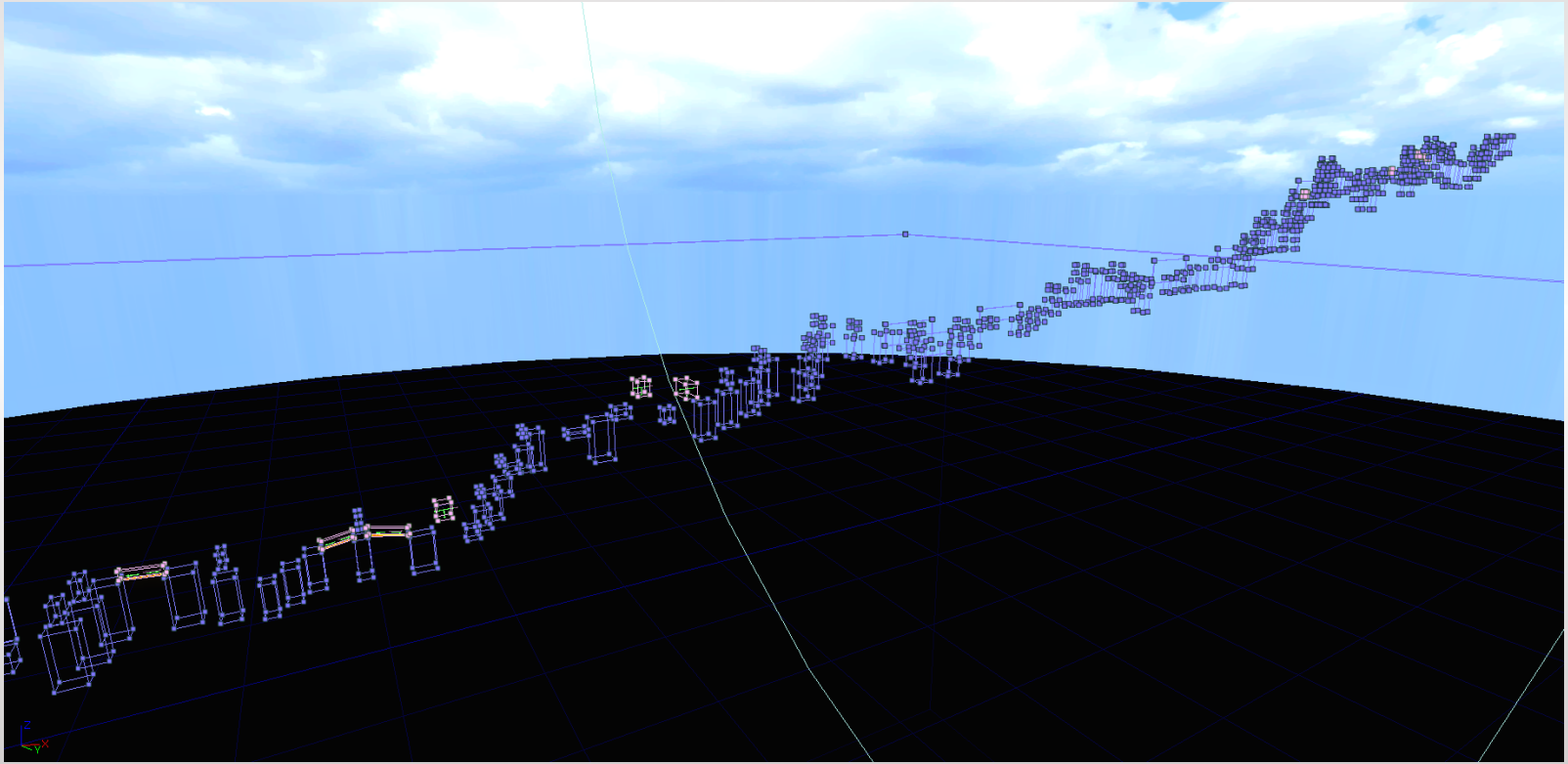
Basic Computed Constants

Height to Jump and Pull Up	320
Up Wall Scale	480
Damaging Jump	540
Death Jump	1000
Wall Run (No Jump)	1000
Wall Run (With Jump)	1600
Jump Distance (No Run)	400
Jump Distance (With Run)	550
Jump Distance (With Run And Full Fall)	1100
Turning Scale Jump	600

Path Generation: Technique

- Uses random sampling technique to create initial path between two points using only simple reachable cube platforms.
- Iterate over the initial path to add interesting elements like swings, wall runs, cube vaults, etc.
- Write to .t3d via the ObjectManipulator interface.

Path Generation: Result



Challenge Roadmap


Modding Mirror's Edge


Creating Procedural Content

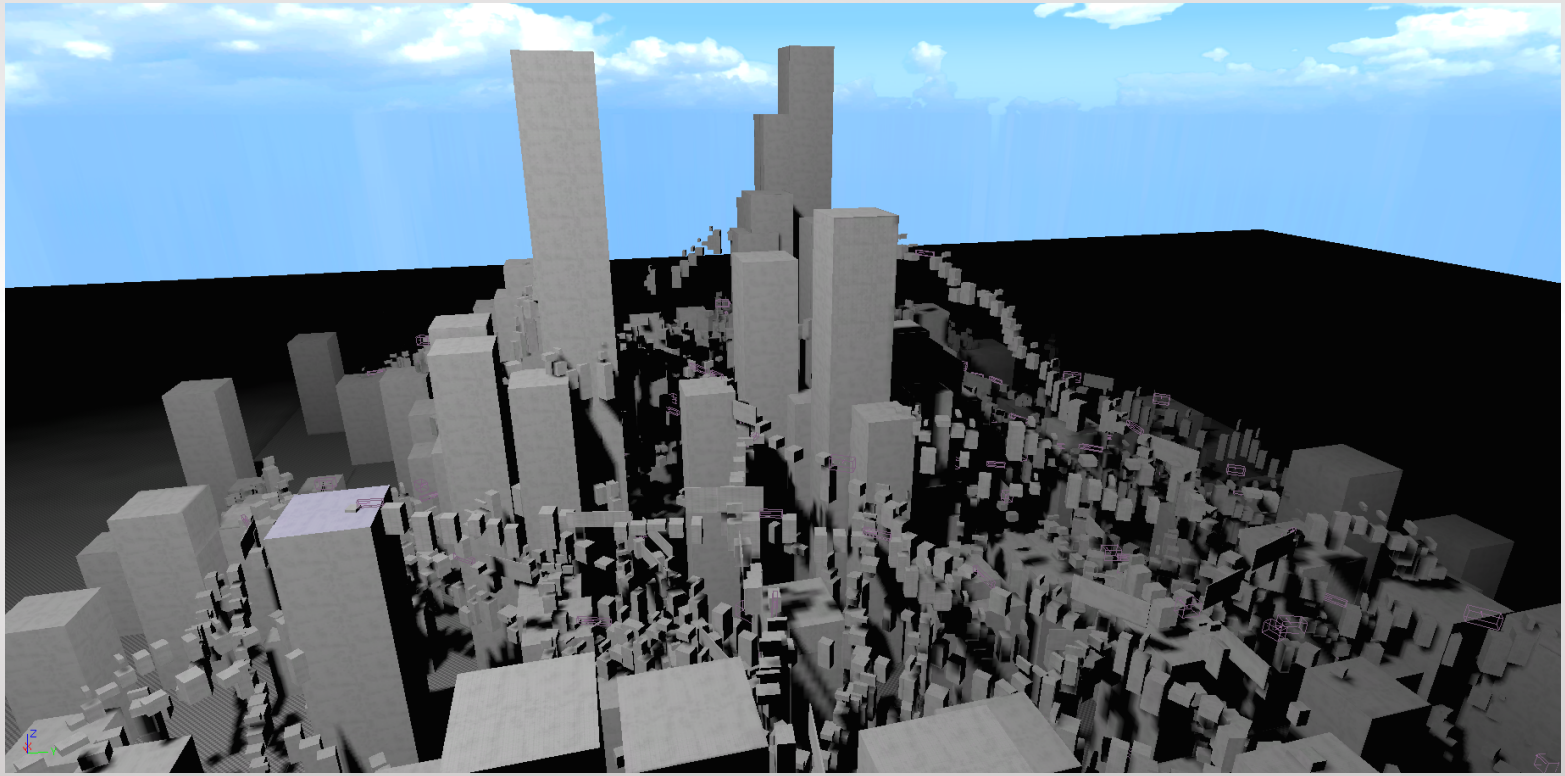

Generating Game Levels

Challenge Three: Generating Game Levels

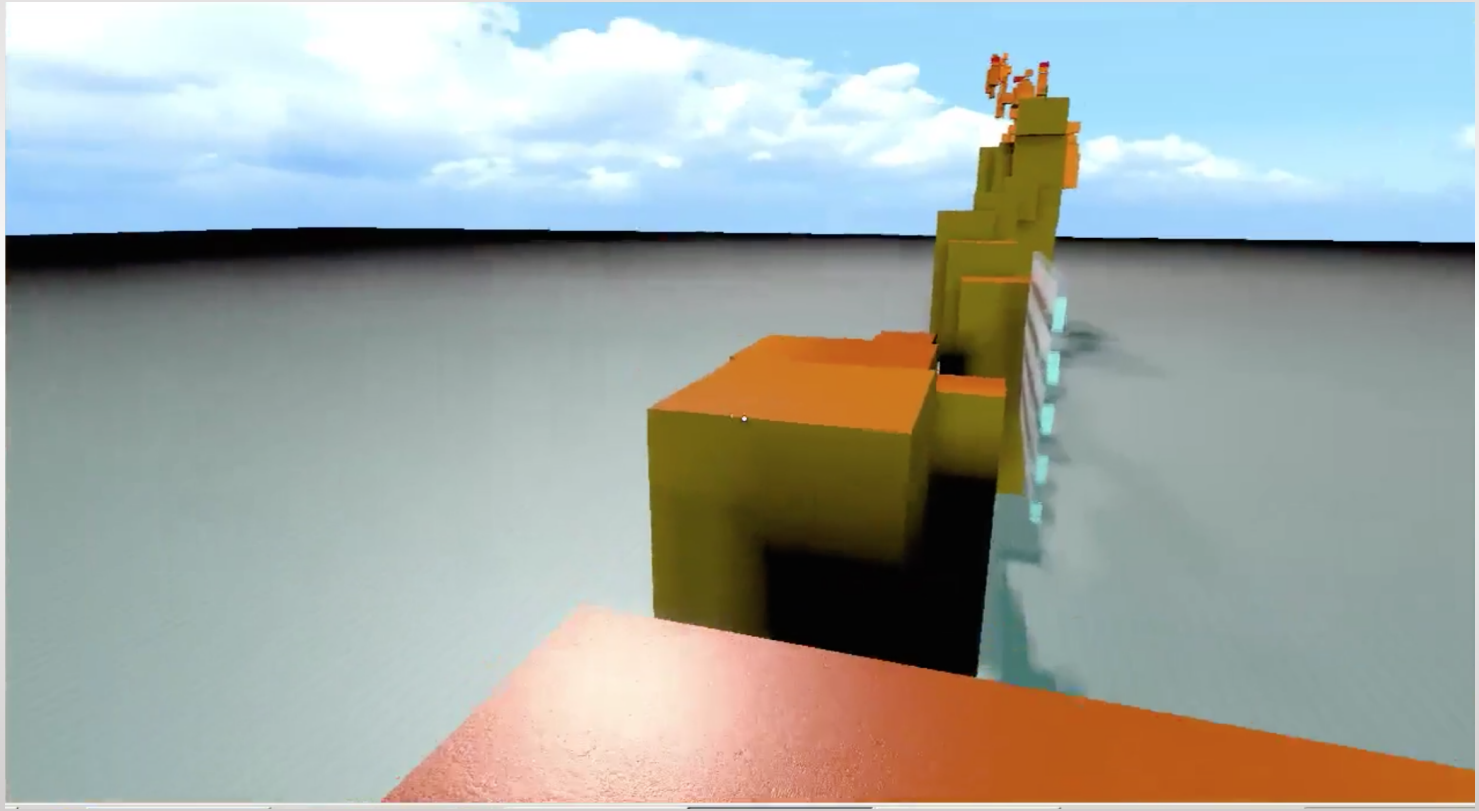
Our code is meant to be a tool for level designers.

- Therefore to prove our process, we must show that it is a beneficial software for content creators.
- This is best proved by creating a level, or rather many levels, by ourselves to show both the speed of creation and diversity of results.
- The following slides show some of our procedurally created content generated within one days work.

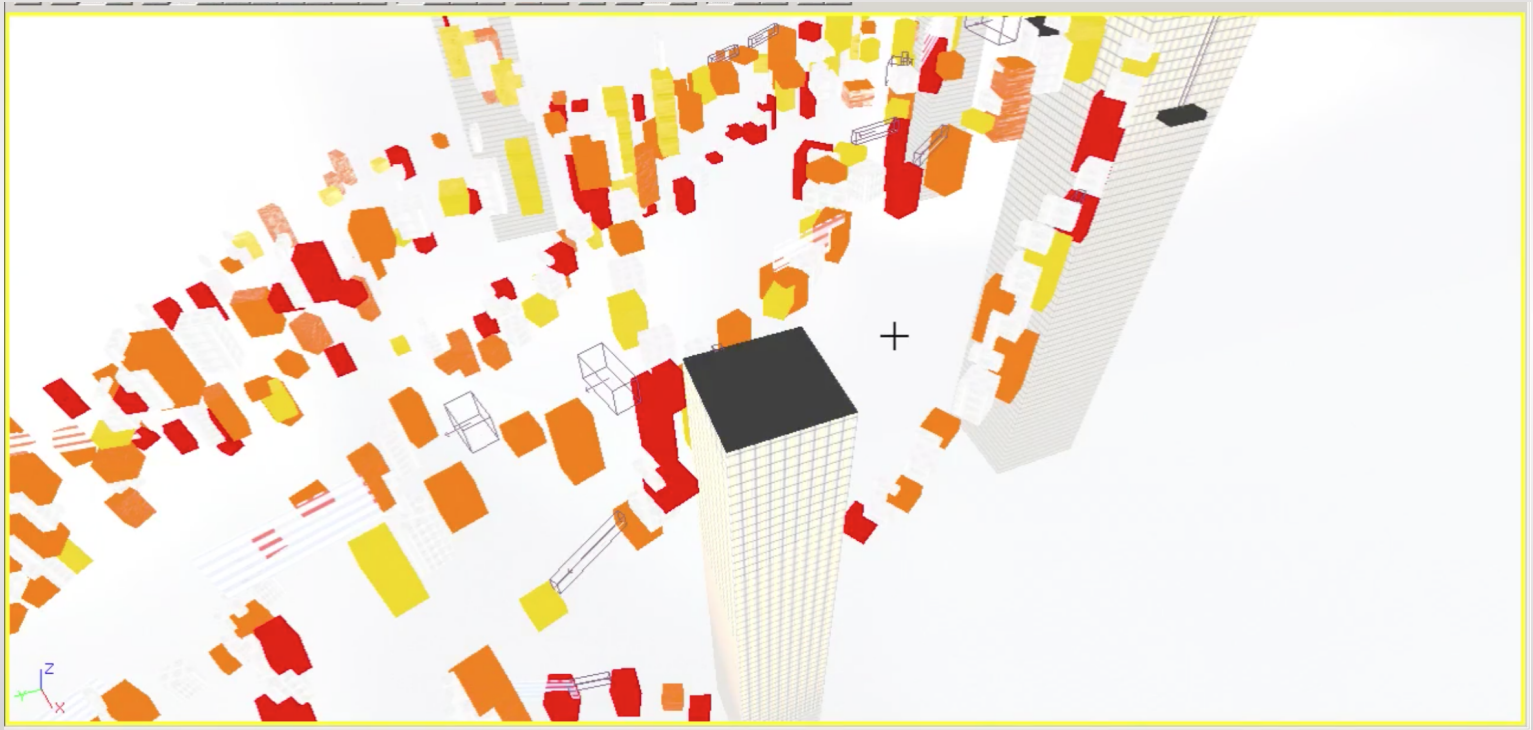
Procedural Level: Proof of Concept



Procedural Level: Textures



Procedural Level



Result

Acknowledgements

Special Thanks to:

- DICE
- Kevin Macleod
- Jeff Stewart
- Morgan McGuire
- Sam Donow
- DK3TM midterm team