

Building the City of Glass in Mirror's Edge™

Johansson, Daniel

Schmid, Jan*

Electronic Arts / DICE



© 2015 Electronic Arts Inc.

Abstract

Mirror's Edge (working title) is a reboot of the iconic DICE game from 2008. Using the *Frostbite™* game engine (used in *Battlefield 4™*, *Dragon Age™: Inquisition* and *Battlefield Hardline™*), we are creating a totally new and unique gaming experience on a proven technology platform.

To create the city of "Glass" that the story of the game takes place in, we have added several new tools to the *Frostbite* feature set. Creating art content while maintaining the possibility to make late changes to gameplay and level design is something that has been a challenge for us, and therefore something that has been a priority to solve when implementing the tools. The intended audience is someone who is familiar with computer games production and art pipelines.

1 Constructive Solid Geometry

Constructive solid geometry (CSG) is a well known and explored way of modeling geometry that has recently been implemented into the *Frostbite* engine. This allows us to rapidly create and prototype shapes for the game without the need for going into a DCC application. It's used by level designers and level artists create a rough 'whitebox' version of an entire area, as well as for creating placeholder versions of objects before a final version is made. Additionally we have come up with an authoring methodology for buildings where the blocked out CSG content is refined and used all the way up to the final asset.

2 Modules and Clustering

To maintain flexibility and efficiency when creating buildings we are using using a "Modular" workflow, which in essence means that we create sets of modular parts intended for a single building style. These can then be combined in various ways to create unique

versions of building using different variations of the same modular parts. To keep our instance counts down we have created a system to bake down buildings consisting of many objects to a single object, which is used sections of the city that is currently far from the viewer. This system is often used together with CSG to create more variation.

3 LOD & Material Reduction

Making a game such as *Mirror's Edge* requires us to maintain high frame rates and to have a memory footprint that fits into each target system. Due to these constraints we have devised a system that can bake down multiple shaders and textures to a single set of textures, allowing for much more efficient LOD steps. Additionally we are using several systems and methods to generate and optimize our LOD's automatically.

4 Texture Arrays & Interior Mapping

To further reduce the draw call counts we make use of a graphics hardware feature called "Texture Arrays". This allows us to efficiently group together several draw calls by combining their textures. "Interior Mapping" is a name we use for the multitude of shader and texture tricks we've employed to give our buildings some depth and believability, making it appear as if they had an actual interior behind the flat geometry.

5 Streaming

The setting in this game is a quite different environment from the outdoor vistas often seen in for instance the *Battlefield* game series. The amount of unique content visible in a single frame requires us to make much more extensive use of streaming technology than in previous titles.

We have also developed a system for streaming types of content that differ from the regular streamed content and do not have the same time or throughput requirements. This system is called the "Linear Media Streaming System" and is used to stream content such as movies for in-game billboards and complex geometry animations.

*daniel.johansson and jan.schmid at dice.se

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.

Copyright is held by the owner/author(s).

SIGGRAPH 2015 Talks, August 09 – 13, 2015, Los Angeles, CA.

ACM 978-1-4503-3636-9/15/08.

<http://dx.doi.org/10.1145/2775280.2775282>