Architecture, Design and Conservation
Danish Portal for Artistic and Scientific Research

Aarhus School of Architecture // Design School Kolding // Royal Danish Academy

Game Engines as Dynamic Tools in the Design Phase

Kreutzberg, Anette

Publication date:
2011

Document Version:
Publisher's PDF, also known as Version of record

Link to publication

Citation for pulished version (APA):
Kreutzberg, A. (2011). *Game Engines as Dynamic Tools in the Design Phase*. Paper presented at First International Congress of Retevitruvio, Bari, Italy.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 23. Apr. 2024

ARKITEKTSKOLEN AARHUS
Designskolen Kolding
Det Kongelige Akademi

Game Engines as Dynamic Tools in the Design Phase

ANETTE KREUTZBERG,
Royal Danish Academy of Fine Arts, School of Architecture

Introduction

Novice architecture students usually have a problem grasping scale when sketching digital 3d models. Although modelled in 1:1, an object on the screen may be perceived as being much larger or smaller due both to the ability to zoom in and out and to the level of abstraction created by an absence of assigned materials and surface properties and a lack of objects of known size placed within context. Connections and the flow between spaces can also be difficult to imagine.
To perceive the scale of a digital architectural model, man relates to its spatiality. Scale can be defined through an eye-height corresponding to an upright human adult (160 - 170cm). By reducing the eye-height, the model may seem bigger and conversely, by increasing the eye-height, the model may seem smaller than it really is. With 3d modelling software, navigating in the digital environment is mostly free of constraints thereby possibly distorting our perception of proportion. This completely free movement can even be combined with simulations of camera lenses where manipulating the focal length also has an effect on the perception of scale: a telephoto lens pushes the model away thus a larger area is shown on screen and the model may therefore seem to be more extensive opposite a zoom lens, which pulls the model closer.
Scale and spatiality can also be evaluated through motion. Animated movement through a digital model is an effective method to explore and assess scale and spatial qualities including connections and passages. The main drawbacks with animations are time related: lengthy rendering time and time used for planning and setting up camera paths.
Previous works on spatiality and flow in video games[1] have inspired to introduce game development tools in the preliminary development and characterization of architectural projects. By adopting principles from first person shooter[2] action games, it is possible to navigate a first person controller with a fixed eyelevel and restrictions by way of physical simulations such as gravity and collision detection thereby setting the scale and enabling the exploration and evaluation of space and spatial sequences with immediate feedback.
A general problem associated with using most game develop-

ment tools is, however, the need for extensive programming skills, usually not part of the architectural curriculum. This study shows the Unity game engine to be easily comprehendible for architectural students and useful as a dynamic tool in the design phase as well.

Digital skills

Today architecture students need to have a well-developed digital culture and adequate computer skills to perform in the digital realm. They are introduced to a number of different digital tools covering modelling, construction, analysis and presentation[3]. During the first two study years, three mandatory courses are offered covering digital representation and graphic arts; digital 3d modelling; and light and materials in the digital model.
Specialized courses and workshops are offered on command, often in collaboration with study departments. Within the last four years several workshops covering movement in virtual environments have been held, introducing different game development tools[4]. Often students have found the software technically very difficult, which has largely been due to their lack of programming skills.

Unity Game Engine

The Unity 3d game engine was introduced to 22 undergraduate students at a workshop titled "Movement through the Digital Model" held at The Danish Royal Academy of Fine Arts, School of Architecture. The workshop included real-time movement in the Unity 3d game engine, animation in Autodesk 3ds Max Design and animated storyboards edited in Adobe Premiere Pro as methods to explore and communicate architectural form and space.
Prior workshops with architecture students using computer game technology[5] have made it clear, however, that it is essential for success that the chosen game engine must have a user interface which is easy to comprehend visually and that the import of geometry from 3d modelling software must be smooth. In this workshop the Unity game engine was introduced as a tool for investigation and communication during the initial sketching phase and not as a visualization and presentation tool. In the sketching phase, tactility is not essential whereas the spatial aspects of the work are in focus, meaning that in-depth knowledge of handling Material properties and Shaders[6]

in the game engine was not mandatory and could be kept at a very basic level.
The Unity game engine was chosen because of many reasons: it has a free version (with limited functionality), it works on both Mac and pc-platforms and it has an appealing drag-and-drop visual user interface that is easy to learn and does not require programming skills for basic usage. It supports direct import of 3D geometry from 3ds Max and, via the FXB[7] file format, supports files from SketchUp, Rhino and Revit, which are the 3d modelling software applications presently taught at the Royal Danish Academy of Fine Arts, School of Architecture. Furthermore, the Unity Editor can publish web player Builds, which makes it easy to distribute the interactive models if desired.
The lack of real-time shadows in the free Unity Editor used in the workshop was actually the only major drawback; the Unity Pro version would have offered this feature.

Unity Editor

The Unity Editor interface is organised into several windows: Scene View is the working view; Game View is used for testing in real time; Hierarchy Window shows assets used in the current scene; Inspector Window shows properties of selected assets; and Project Window shows all assets for the complete game: Scenes, GameObjects, Materials, Sounds and Scripts.
A Unity game consists of many parts: The Project is a folder with the complete "game", including all assets; Assets are 3D meshes, materials, textures, scripts, sound files, particles and PreFabs; Scenes are individual levels; Game Objects are every object in the game; Components give behaviour and "life" to GameObjects; Scripts become components when they are saved; A Prefab is a type of asset, a reusable GameObject stored in Project Window.
The relationships between the most common assets are as follows: A Texture is applied to a Material. A Material is applied to a GameObject with a Mesh Renderer Component. An Animation is applied to a GameObject with an Animation Component. A sound file is applied to a GameObject with an Audio Source Component.
Physical properties are applied to the geometry such as mass, gravity and collision. A First Person Controller (FPC) consists of geometry, camera and script, with a density similar to that of humans applied to the geometry. The force of gravity causes the FPC to be dragged down, and to avoid falling through floors or walking through walls, these need to have Colliders allocated.

Unity project structure

To make student adapt the very stringent workflow of the game engine, they need to be convinced of the importance of understanding the underlying structural and logical principles like not moving or renaming assets in the Project folder using Windows Explorer (PC) or Finder (Mac) – paths may be lost. The relocation and renaming of objects in the Assets folder should always be done from within the Project Window. Explanatory naming of objects facilitates the workflow considerably in Unity. The original 3d model should be placed close to 0, 0, 0 in the 3d modelling program to support easy import and System Units should be 1 cm to match Unity units.

Workflow

The workflow we tested can be described as follows: The Unity Editor and 3ds Max Design are both opened. From within Unity, a new Project is created including the import of an Assets Package prepared especially for this workshop. The Project Assets Folder is automatically created upon making a new Project. 3ds Max files containing the 3d models are imported (copied) into the Project Assets folder and colliders are added. Any Bitmap files for materials are placed in a folder called Materials at the same hierarchical level as the 3ds Max files or higher. Assets are added to the scene, combined with components and placed with Transforms. Editing the 3ds Max models located in the Assets folder will automatically update the corresponding assets in the Unity Editor. Terrain, SkyBox and sunlight are added to the scene. A First Person Controller PreFab is added and the exploration begins by activating the Play Window. For presentations, a Web Player Build can be made. When saving the project, the current scene is also saved as a game level.
Usually, architectural digital 3d models are constructed by students for visualisation purposes and include advanced materials and lighting setup. The MentalRay renderer included in 3ds Max Design supports very complex material types and shaders as well as Photometric lights. These material types and lights are not supported by Unity, but the diffuse map can be transferred to the Standard material, which is recognized by the engine, as can a light map placed in the self illumination channel. A maximum of two UVW maps can be imported together with the geometry usually used for controlling the diffuse map and the light map. Bump maps are created as Normal maps within Unity. Lights are created within the Unity Editor.

Workshop preparation

Introduction to real-time movement was the first part of the workshop and students worked with their modelling software and the Unity 3d game editor simultaneously. When editing their 3d geometry, they could instantly check the effect in the real-time virtual environment. In this workflow, students were able to detect inexpediencies in their architectural designs otherwise overlooked and could act upon the discoveries appropriately.

The workshop was planned in collaboration with one of our study departments. It was placed in the middle of a semester project where students were working with architectural designs for dwellings as both single units and combined in a complex of 6-8 units. The students brought with them a common digital context model and their own individual digital sketches of housing models.

The aim was to teach the principles and use of a game engine for exploratory purposes within two working days followed by two days focusing on refining the pipeline and working with the game engine to qualify their own projects through the exploration and analysis of shape, spatial sequence and flow. The last three days of the workshop introduced the animated storyboard for the documentation and communication of their projects by using animations and cinematic visualization.

A small workbook with instructions and explanations was written for the workshop both to assist students with the pipeline and to be used as reference material after the workshop. Furthermore, an Assets Package was prepared containing the most important assets needed. To reuse assets between projects, Unity can export Assets Packages including metadata about the specific assets (import settings, links to other assets, etc.). The Package contained: A PreFab First Person Controller with all properties preset ready for use; a Skybox to wrap around the entire scene displaying a distant view of the sky including the Material used to render the Skybox containing 6 Textures; and, finally, a glazing Material.

Workshop course

The first day of the workshop was preceded with a short presentation on course content and format where after the Unity 3D website was presented showcasing numerous samples of games and demo projects. Previous workshops had indicated that it should not be taken for granted that all students were experienced gamers, and since it is important to comprehend basic

game mechanics and controls in order to work satisfactorily with game engines, it proved particularly successful to introduce the principles by having a play session. Game mechanics are defined as methods invoked by agents for interacting with the game world and how actions are mapped onto input devices (Sicard, 2008).

Students were asked to first explore the Tropical Island demo from the Unity website[8] to practice navigating with a mouse and WASD (or arrow-) buttons. The Tropical Island demo is an adventure-action genre game with the view perspective of the playing character controlled by the player. This perspective is meant to give the player the feeling of immersion in the virtual world: a feeling of actually being there. The demo also demonstrates many of the features supported by the Unity Game Engine, including real-time shadows, ambient and directional sound and camera effects like Lens Flare, etc. Students were encouraged to help each other with the techniques and to try out the other demo games on the Unity website. Inexperienced gamers could try out computer demos with simple game mechanics and controls and also had the opportunity to look over the shoulder of fellow students before trying out the more complicated game mechanics, inspired by the way children in groups learn to play games when only one computer, console or device is available.

The experienced gamers amongst the students soon switched over to action games like "Avert Fate"[9] and "3rd Person Shooter"[10]. Since all computers in the IT lab have speakers, the sound effects simulating gull caws, gun shots and explosions raised the noise level significantly. The sound effects also helped to generate interest in the various game demos, and the students watched each other play and were inspired to try themselves.

Some of the demo games like "Character Customization"[11] and "Butterfly Demo"[12] appealed especially to tentative female students. In Character Customization the only interaction is clicking buttons to change the appearance of the character. The Butterfly Demo is presented as a demonstration of soft shadows and depth of field, but the very slow and laid back movement in the game mechanics made it suitable for hesitant learners to practice navigation with a combination of the mouse and keys.

In the afternoon, the Unity Editor was presented and the user interface and the rather rigorous and logical design of the file structure and naming conventions were reviewed. The Fantasy Island project was used as a demo project once more to demonstrate the functionality and structure inside the Unity Editor. Start-up of a new project was reviewed and the students made

their first attempts at importing their own 3ds Max models into Unity. One student very quickly downloaded a car racing demo project, including the working files, and personalized it with his own and other downloaded models, switching the cars to Eames chairs and racing them through the common student context model!
On the second day, the workflow was refined importing 3ds Max models into Unity, returning to 3ds Max to change and edit the models and then scrutinising the updated model in Unity.

The student projects were at an early development stage, and many details had not been decided on. There were major realisations for some of the students: When a model is assigned attributes like Collision Geometry, you cannot just walk through walls- there must be an opening large enough for the First Person Controller to pass! Obvious perhaps, but still not something you need to decide on to navigate in the original 3D modelling software, where you just move the camera or the eye of the beholder to go where you please. Similarly, it was also necessary to introduce stairs and ramps to move from one floor to another. The replacement of windows according to view likewise became an issue as students navigated around inside their models.
The third and fourth workshop days were used to further develop their projects. On the third day, all students received tuition from their study department teachers in the workshop lab and the Unity Editor was used spontaneously as a communication tool to show results and discuss the projects.

Workshop evaluation

The students were all very excited about Unity from the very beginning. Some were quicker than others to learn the techniques and to use them creatively, but overall, the general attitude was an acknowledgement that this workflow can be used and refined for something sensible. In the final written evaluation of the workshop, students stated:

- I think that the course might be best placed after a modelling course as there would be a good model with which to start. I would like the possibility to go into greater depth; it was a bit of a rush as the 3d model wasn't ready.
- The way the course was conducted and the hands-on approach of Unity as a tool worked really well.
- Nice to try your hand at new tools and media which can be introduced to your processes in the studio.

- A pleasure to become acquainted with new methods and programs which can be combined with previously learnt processes, programs etc.

Conclusions

The well-developed digital culture amongst our new students today promotes the introduction of new methodologies for the design process in the three dimensional, digital realms. However, since students must pick up and master numerous digital tools, an assessment must be made as to whether the extra workload is balanced by the benefits.
Using game development tools, it is possible to control free movement through the digital model, and the study shows that when applied in the early sketch phase, the spatial realizations thereby obtained qualify the architectural projects in a way that would not otherwise be possible. Furthermore, the game engine environment has proved to be useful in conveying the architectural intentions in projects in the context of individual supervision sessions.
It is important to comprehend basic game mechanics and controls as well as the underlying programming structures to work satisfactorily with game engines. Not all students were experienced gamers and none of them had previous experience in programming, but nevertheless, the techniques were taught in just two days with only two more days for refining the pipeline process. With regard to the use of the Unity game engine, the study proved it to be a dynamic and helpful tool in the design phase thereby justifying the extra workload and students have been encouraged to use the engine to qualify future projects and should be awarded accordingly.

Notes

[1] IT-University of Copenhagen, Msc study programme in Media Technology and Games, 3d Game Art
[2] The game world is displayed from the first person perspective. A perspective always used with avatar-based interaction models in which the virtual camera displays the game world from the point of view of the avatar's own eyes (Adams, 2009)
[3] The mandatory IT courses offered in the bachelor programme at the Royal Danish Academy of Fine Arts, School of Architecture are conducted and arranged by the IT Educational Unit at Institute of Design and Communication. The institute's area of study encompasses architectural design (with its various design-related specialisations), architectural tools

and methods for description and visualisation (for use in processes of documentation, analysis, design), and communication and information technology and its possible uses in the design process

[4] Glacier Engine by IO Interactive, TurnTool by TurnTool, Source by Valve Corporation, Unity 2 by Unity Technologies

[5] Det virtuelle rums arkitektur 2007, Arkitekten i spil 2008-09

[6] There is a close relationship between Materials and Shaders in Unity Shaders contain code that defines what kind of properties and assets to use Materials allow you to adjust properties and assign assets

[7] FBX (Autodesk) is a Platform-Independent 3D Data Interchange Technology that can be used to Transfer custom data between applications

[8] http://unity3d.com/gallery/live-demos/index.html#tropical-paradise

[9] http://unity3d.com/gallery/live-demos/index.html#avert-fate

[10] http://unity3d.com/gallery/live-demos/index.html#3rd-person-shooter

[11] http://unity3d.com/gallery/live-demos/index.html#character-customization

[12] http://unity3d.com/gallery/live-demos/index.html#butterfly

References

ADAMS, E., 2009: "Fundamentals of Game Design". *New Riders Press*.

SICARD, M., 2008: "Defining Game Mechanics". *Game Studies,* vol. VIII.