A Tool for Landscape Architecture Based on Computer Game Technology

Marc Herrlich
University of Bremen
Research Group Digital Media
Bibliothekstrasse 1, 28359 Bremen, Germany
mh@tzi.de

Abstract

Modern computer game engines have reached a level of visual quality that makes them usable for other tasks and applications than mere entertainment. In this paper we will present our prototype of an interactive tool for landscape architecture based on the well-known CryENGINE and the game Far Cry. We will present the steps necessary to automatically derive a valid Far Cry level for real GIS data and show some preliminary results.

1. Introduction

Nowadays computer games are an important part of many people’s everyday life. Over the last decade computer game technology has made great advances, especially concerning visual quality and realism, and modern game engines have reached a state where they compete or even surpass professional visualization tools in some areas.

While computer game engines have certain advantages, some of which we will present in this paper, it is clear, that they also have some disadvantages compared to professional visualization tools because in general they favor visual quality and real-time rendering over physical correctness.

Research in the field of applying games technology to other areas besides entertainment, e.g. educational or scientific tasks, is not altogether new. While it has been around for many years, recently the availability of cheap but powerful consumer hardware and the new level of maturity reached by the computer games industry have sparked new interest into this topic, also known as Serious Games or Serious Gaming.

In this paper we will present an approach to apply the visualization and interactional capabilities of modern game engines to other areas than mere entertainment. We will use the CryENGINE to build the prototype of a visualization tool for landscape architects. Our goal is to provide a tool that generates a quick but realistic visualization of an area based on real geographical data, allowing interactive movement through the landscape and real-time interactive modification of the terrain layout and vegetation placement.

One possible usage scenario could be a landscape architect having a meeting with his customers trying to present his ideas and how they will look in reality. As many people have difficulties in visualizing how the look of a landscape might change, e.g. with different arrangements of trees and other plants, it is often crucial to provide images or models of the target outcome to customers, which can be a very time consuming and expensive process. Game technology on the other hand can provide near photo realistic, interactive real-time visualization and - as we will show - even supports real-time modifications and alterations. As computer games are optimized for a broad audience, they run on normal desktop or even notebook computers, so it would be possible for the landscape architect to demonstrate different concepts live and to provide an interactive walkthrough for the customer right there on his notebook or on his customer’s desktop PC.

The rest of this paper is organized as follows. First in section 2 we will discuss related work in the area of Serious Games. In section 3 we will present details about the CryENGINE, its accompanying tools and the data used. We will also provide some reasons why we chose this particular engine for our prototype. In section 4 we will discuss the overall architecture and the design decisions made for the implementation. Furthermore, some preliminary results will be presented in section 5. Finally, in section 6 we will draw a conclusion and discuss open issues for future work.

2. Related Work

Using game technology for other purposes than entertainment, often called Serious Games or Serious Gaming, is a much researched topic. Giving an extensive overview is certainly beyond the scope of this paper. We will therefore only mention some very closely related works.
In [1] Bergen, McGaughey and Fridley discuss the benefits of using a data-driven approach to landscape visualization in the context of the Vantage Point visualization tool and how certain features of such a system, e.g. ground textures or lighting effects, contribute to visualization quality. Most of these features are readily available in modern game engines, such as the CryENGINE.

In [8] Herwig and Paar discuss the general benefits and drawbacks of using game engines and game technology for landscape visualization and planning. They present two examples of landscape visualization using two different game engines, i.e. the Unreal engine and Conitec’s A5 engine, embedded in a collaborative landscape planning process. They provide some preliminary results based on a feasibility study.

However in contrast to this paper they concentrate more on the general scenario of using game technology for landscape planning purposes. Furthermore their example applications focus on visual quality and appearance and do not include approaches towards automation of the process. The visualization and game levels respectively are mostly created by hand and they use additional commercial software tools like ArcGIS for data conversion. Moreover their sample applications do not include any real-time interactive editing features.

In [5, 6, 7] Germanchis, Cartwright and Pettit also discuss using game technology to visualize geographical data but they concentrate mostly on urban areas and on aspects of human wayfinding, navigation and spatial cognition.

3. Tools and Data

3.1. The CryENGINE

The CryENGINE [3] is a commercial computer game engine developed by the German game studio Crytek. It was first employed in the game Farcry. The engine itself is accompanied by an interactive editor tool called CryEngine Sandbox [2], which is automatically installed together with the game and is free for non-commercial uses. When we talk about using the CryENGINE in this paper, we mean using the implementation distributed with the game Farcry.

The CryENGINE being a commercial game engine means that using it is, of course, not free in general and the source code to the engine is not publicly available. However, it is possible to create content, e.g. so-called maps or geometric models and textures, to use with the version of the CryENGINE embedded in the game Farcry. Additionally, some parts of the engine, i.e. mainly the parts containing the game logic and high-level functions, are available in source as a so-called modding SDK, thereby providing a means to modify some aspects of the game behavior or to add new functionality within certain limits.

The overall approach we follow in this paper is to convert geographical data into a representation readable by the CryEngine Sandbox, making use of its real-time editing features, and finally deriving a map that can be used with the CryENGINE and in the game Farcry respectively for interactive real-time exploration and visualization.

The decision for the CryENGINE was made mainly because it supports the visualization of large outdoor terrains. Being able to use sufficiently large outdoor areas is an important factor in our scenario, as landscape architecture in general deals with designing outdoor landscapes and not room interiors, as the word landscape already implies. The existence of the CryEngine Sandbox - or short Sandbox - also was a key factor in the decision process. The Sandbox is distinct from other editing tools because it features real-time interactive editing and modification, which includes a comprehensive set of tools for terrain shaping and vegetation placement, providing seamless switching between in-game and editor modes.

![Figure 1. CryEngine Sandbox](image)

Figure 1 shows a screenshot of the CryEngine Sandbox with a tree selected for editing. In the main area the landscape is displayed interactively using in-game graphics, with the selected tree denoted by a green circle, and on the right the property editor is located, where the user can make modifications, which are reflected on the selected object in real-time. It is also easily possible to move the tree around, to delete it, or to insert additional vegetation.

3.2. Data

In order to visualize the terrain and to place vegetation accordingly we need two kinds of input information. First, we need a digital elevation model (DEM), which describes the general shape of the landscape to a degree limited by the resolution of the available DEM. Second, we need a
segmentation of the terrain according to types of vegetation present in the respective areas we want to visualize, i.e. forest areas, field areas, meadow areas, etc. We will discuss the DEM first.

In this paper we only consider regular spaced DEMs, defined as a regular two dimensional grid with additional height information for each point of the grid, representing a height field of the underlying area, sampled at the corresponding grid points.

In the second type of data we need is the segmentation of the underlying terrain or landscape according to different vegetation types as we already mentioned above. For this purpose we use ESRI\textsuperscript{1} shape files \cite{4}, a standard commonly used in geographic information systems. Shape files allow to add meta data information to a polygonal segmentation of a geographic area, which is just what is needed for the conversion process.

Figure 2 shows an excerpt from an example DEM in the plain text format we use in our system. Each line describes a single grid point. The meaning of the different columns is as follows. The first column contains a consecutive numbering for each grid point, while the second and third column respectively contain the Gauss-Krüger coordinates \cite{10}. The fourth column denotes a quality measure for the grid point, which is currently neglected in our system and therefore not discussed further in this paper. Finally, the fifth column contains the height information measured in meters above sea level. The remaining four columns contain additional information, e.g. components of the gradient vector at each point, but they are also not considered in the current version of our system.

The second type of data we need is the segmentation of the underlying terrain or landscape according to different vegetation types as we already mentioned above. For this purpose we use ESRI\textsuperscript{1} shape files \cite{4}, a standard commonly used in geographic information systems. Shape files allow to add meta data information to a polygonal segmentation of a geographic area, which is just what is needed for the conversion process.

Figure 3 presents an example shape file, demonstrating a possible landscape segmentation. It shows different named areas like Field, Forest, or Grass. The required level of detail of the segmentation largely depends on the needed realism of the visualization in a concrete application scenario. In the presented prototype application the level of detail shown in figure 3 already produced convincing results (cf. section 5).

The question still remains how to obtain suitable data, i.e. a DEM and a corresponding shape file segmentation, of a particular geographic area. In practice DEMs most of the time will have to be bought from a commercial company or public service. Shape files could either be derived from aerial photographs or - if available for the region of interest - also be bought from a service provider.

4. System Architecture

We will now describe in more detail the overall workflow and architecture of our prototype system. The process can be divided roughly into the following steps:

1. Conversion of GIS data
2. Manual refinement
3. Final level generation

In the first step a tool we developed is used to convert the aforementioned GIS data, i.e. the DEM and the shape file segmentation (cf. section 3.2), into a valid source file format for the CryEngine Sandbox. Therefore the DEM is converted into a suitable height map and vegetation is placed according to the shape file information. Currently the placement of vegetation inside the area denoted by the shape file follows hard coded rules and only works on the area level not on the level of single plants. To be able to create a realistic ground texture in the Sandbox, the converter also generates so-called surface masks. These are output as simple bitmap files and can then be loaded into the Sandbox. The Sandbox can then generate the final ground texture by using these masks together with surface descriptions, which are also automatically generated by our conversion tool and include a reference to a suitable texture for each surface type.

\textsuperscript{1}Environmental Systems Research Institute (ESRI) Inc.
Figure 4 shows how the generated height map looks like interpreted as a gray scale images. In the conversion process the Gauss-Krueger coordinates have to be mapped to level coordinates, heights have to be adjusted, and coordinates have to be matched with the shape file descriptions. Figure 5 shows a sample surface mask generated for a single type of area. Up to seven different area types, i.e. surface masks, can be handled by the CryENGINE to compose the final ground texture.

In the second step manual refinements to the overall landscape layout, i.e. terrain shape or vegetation placement, can be made using the standard CryEngine Sandbox tools (cf. section 3.1).

In the final step the CryEngine Sandbox is used to generate a valid Farcry level, which can then be explored in the game or directly in the editor. Adjustments can be made anytime by iterating steps 2 and 3.

5. Results

We have tested out prototype with sample data from certain areas in Lower Saxony and have received some very convincing results concerning realism and visual quality. Figure 6 shows the final generated ground texture and figure 7 the corresponding aerial photography of the same area. Notice that they match very well even though we used a very coarse shape file segmentation.

Figure 8 demonstrates the quality and realism of the visualization. Please note that for the implementation of the prototype we did not use any custom models or textures but instead only used the resources that come with the game and therefore the vegetation is not of the type you would find in reality in the sample area.

6. Conclusion and Future Work

In this paper we presented an approach to use existing computer game technology, i.e. the CryENGINE and the CryEngine Sandbox, to build a support tool for landscape architecture. We developed a converter tool to facilitate easy conversion of real GIS data into a suitable source format to exploit the interactive real-time editing capabilities of the CryEngine Sandbox.

Furthermore we compared the generated level ground texture with real aerial photographs to demonstrate the soundness of the conversion process. Finally we presented screenshots to account for the quality and realism of the resulting visualization.

In the future we would like to extend on the presented work. We would like to improve the conversion tool by using configurable rule sets instead of hard coded rules for vegetation placement and by supporting more fine grained vegetation placement control. While our results have already been good concerning the visualization of forest and field areas, landscape architects might also need to visualize more urban areas, which we do not support, yet.

Additionally we would like to do a thorough evaluation with more sample data under real-life conditions, i.e. deploying the prototype application to landscape architects and their customers, and we are also interested in comparing our system to commercial systems currently used in the respective field.

Another aspect would be to incorporate standards like CityGML\(^2\) [9] for input and output of the conversion process.

\(^2\)City Geography Markup Language
Acknowledgements

Part of this work was conducted at the Institute for Man-Machine-Communication at Leibniz Universität Hannover. I would like to thank Prof. Dr. Franz-Erich Wolter for making this work possible and Dipl.-Math. Karl-Ingo Friese for his advice.

The sample data was kindly provided by Dipl.-Ing. Roland Hachmann and Dipl.-Ing. Astrid Lipski from the Institute for Environmental Planning at Leibniz Universität Hannover.

References