Generating Virtual Environments by Linking Spatial Data Processing with a Gaming Engine

Christian STOCK, Ian D. BISHOP, and Alice O'CONNOR

1 Introduction

As the general public gets increasingly involved in the development of policies that affect their environments, there is a need for systems that educate the public about important issues related to their environments and can support the making of complex decisions that impact their future. Well known software packages that build on geographical information systems (GIS) for this purpose are "What If?" (KLOSTERMAN, 2001), "INDEX" (ALLAN, 2001), and "Community Viz" (KWARTLER & BERNARD, 2001). In STOCK & BISHOP (in press), we have designed an envisioning system (EvS) that links GIS and virtual reality (VR) and seeks to involve the general public as a non-specialist audience into landscape planning processes. The idea of the EvS is that it is less of analytical but more of exploring and discovering character compared to typical decision support systems. The indicators the EvS provides are more suggestive in nature rather than the definite answers typically sought for planning purposes.

As spatial data becomes increasingly available to the general public and, at the same time, real-time graphics hardware becomes more powerful and affordable, more widespread use of systems that integrate GIS and VR for educational and planning purposes will become feasible (e.g., HERWIG & PAAR, 2002, on the use of game engines for landscape visualisation purposes). To build a system that derives a virtual landscape from accurate spatial data and delivers a reasonable realistic representation of an existing landscape, a fair amount of resources are needed. Firstly, one needs to build a real-time renderer that is powerful enough to handle large datasets of landscape terrain. Secondly, a matching virtual representation of an existing area has to be constructed, including vegetation and man-made structures. Both the implementation of a software package and the construction of a 3D landscape model are costly in time and resources are ideally minimised as much as possible.

We are currently building a system using a commercial low-cost game engine as a base for building a suitable online collaboration environment. To get a 3D dataset of an area of interest, users of the system will be able to log onto a map server, select an area on a 2D digital map and the server will send a 3D landscape model of the selected area back to the users. The generation of the 3D model will be done on the server and the system will be able to automatically create 3D models from anywhere in Australia on demand. The 3D models can than be explored by the user in the VR application. The VR application will also allow users to meet online in a landscape model and have online discussions about issues that are represented in the landscape (for example, a selected area could feature high levels of soil erosion), i.e. it will function as a collaborative exploration and discussion platform.

2 Renderer

Implementing a full featured VR application for landscape visualisation purposes from scratch is very resource and cost intensive and hence, a common approach from developers is to use existing software packages. In the past, one of the common solutions was to choose one of the existing visualisation packages that provide a programming interface for writing real-time visualisation software. Such a product is OpenGL Performer from SGI (http://www.sgi.com/software/performer) which is built on OpenGL and offers higher level graphics programming libraries, including functionality for using scene graphs, and level of detail (LOD) and continuous level of detail (CLOD) management. An open source alternative is the Virtual Terrain Project software (http://www.vterrain.org) which also provides typical functionality needed for landscape rendering, and also provides useful functionality such as geo-referencing. In contrast to OpenGL Performer, which only provides software libraries, the Virtual Terrain Project is a standalone application which runs without adding any code.

What these packages typically lack is an interface to allow networking between multiple applications. When building an online collaboration system it is necessary to have networking capabilities that allow different users to log into the system so they can explore the same landscape model. The application would ideally have a user interface that allows users to share thoughts, and discuss and express their feelings about issues that are visible in the virtual landscape.

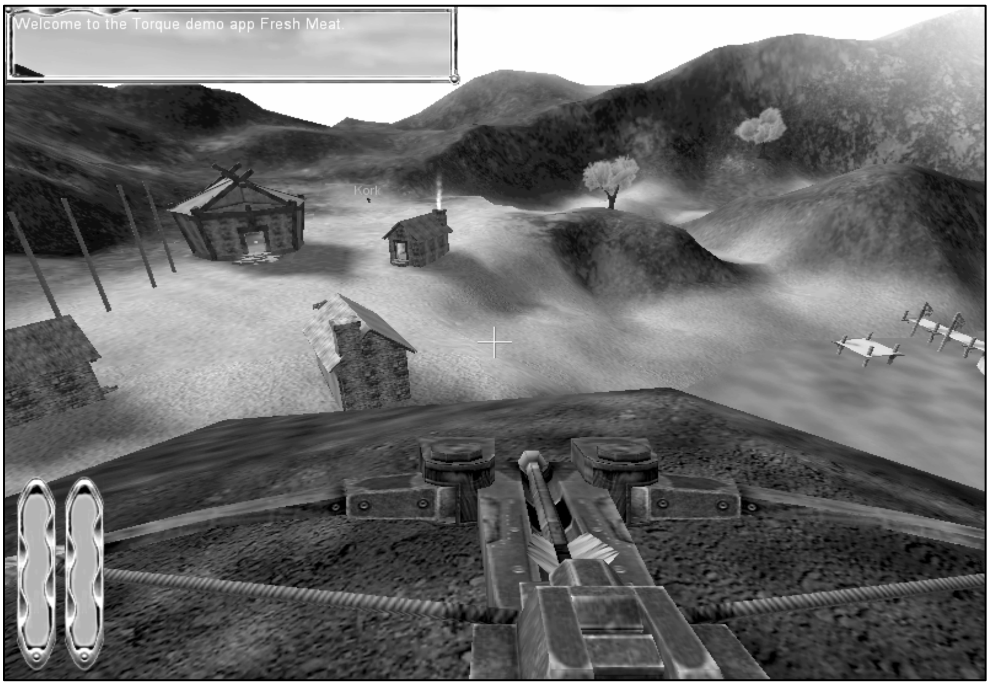

Fig. 1: The Torque Game Engine

Recently, several game engines have become available at low price and with a range of useful features. However, many game engines only allow for content customisation (i.e. adding landscape objects), but not for adding functionality on the code level, which is necessary for the purposes of our system. We are using the Torque Game Engine (TGE) from GarageGames (http://www.garagegames.com), which is currently available for USD 100, including the complete source code (see Figure 1). It features the typical functionality of visualisation packages, such as scene graph and LOD/CLOD functionality. It also supports a multi-user interface including server-client networking and avatar models, and a powerful scripting engine that allows customisation without making changes in the source code. TGE is a full blown application that can be used out of the box, while additional features can be added as desired. Typically, game engines also have some drawbacks. Since they are built to provide users solely with a gaming experience, they do not need to support large landscape models (game environments are usually very restricted in space). For example, TGE cannot support landscapes of arbitrary size. All in all, a game engine is usually limited to the purposes it has been designed for (e.g., some engines only support indoor environments). This means that almost always significant parts of the code will have to be altered to overcome limitations. On the other hand, when not using game engines often such code will have to be written from scratch anyway, so using a game engine may not necessarily lead to extra implementation work.

3 Generating 3D Landscape Models

To develop a realistic virtual landscape visualisation platform, one does not only need the appropriate VR software, but also the content to build and populate the virtual landscape (e.g. suitable tree models, terrain imagery, etc). Generating this content for a study area can be very time and resource intensive if everything has to be built manually from scratch. The system we are building will be able to automatically generate suitable 3D content models from spatial 2D data. Figure 2 shows the principle of how the system will be working. Users can log on into a server and explore a 2D map on their computer. They can than select an area of interest anywhere in Australia and, after the selection is made, on the server the mapping data will be converted to a suitable 3D model consisting of several files. The resulting files will be sent back to the user over the internet. The user will then be able to view the files in the custom built 3D visualisation software (built on TGE).

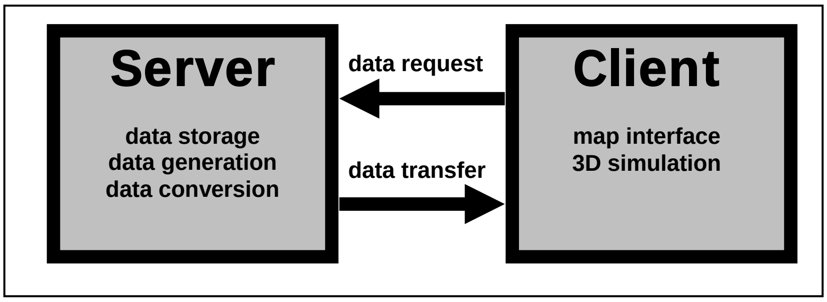

Fig. 2: User interaction with the system

The 3D data conversion tool is working as a module in ESRI ArcMap and can convert raster layers and shape files into a format that TGE can read. The GIS data we are using includes the GEODATA 1:250 000 dataset from Geosciences Australia and the Vicmap 1:25 000 dataset from the Department of Primary Industry, Victoria. The databases will be combined into a federated database for seamless integration and will be hosted on a server. The data will be used to build the 2D map that will serve as an interface for the users for selection of an area. It will also be used to generate terrain models including elevation data, roads and rivers, and manmade structures. In case where no mapping information exists (e.g. tree locations), algorithms will be implemented to generate the necessary additional data. For example, in the case of vegetation, the ecological vegetation classes (EVC) will be used to determine the location of individual species (e.g. species A, B, and C are typically found in EVC 1) and spatial species distribution (e.g. species A occurs in clusters and typically together with species B). Furthermore, we are building a vegetation content library based on existing vegetation textures. This library will include all major Australian species and will be linked to the EVC classes. Based on the mapping info, the algorithms for generating extra data, and the content library, automatically generated 3D models should reflect the existing conditions as realistically as possible. Users of the final system will be able to virtually explore any area in Australia in a real-time 3D environment.

Fig. 3: Landscape around Mt Beauty, SE Victoria

4 Collaborative Environment

Apart from just visualising existing landscape conditions, the system will also allow for importing scenario based landscapes that were generated with the help of scientific environmental process models (which need to be provided from experts as GIS layers for conversion to 3D models). For example, users could load a scenario that is based on a hypothetical development ten years into the future where the issue of soil salinity has been neglected which manifests itself in soil bleaching and dying vegetation. The VR application will have multi-user functionality that will allow multiple users to explore a virtual landscape at the same time. The users will be visible in the environment as avatar based characters. The system will have a user interface that will show all logged users in iconic form in the sidebar (see Figure 4). Name and Affiliation of all users will be visible to other users. The application will also be able to capture images via a webcam, so users in front of their PC can be captured in real-time, similar to a video conferencing application. As an interface for discussion the system will feature a text based system, but it will be also possible to link microphones to the system and have discussions via streaming audio. Users can use this platform to explore landscapes, find environmental issues that are relevant to them, and discuss their opinions to form strategies for overcoming them. Ideally, the system will provide facilities to allow for collection of feedback (e.g. via a voting system) and to record opinions for later evaluation. The system will need some sort of a facilitating mechanism so discussion can be conducted in an organised manner. Research and user testing will have to be done on how to best implement such a collaborative environment.

Fig. 4: Collaborative environment

5 Summary and Outlook

We are building a collaborative environment that is built on TGE, a commercial gaming engine. The system will consist of a VR application that features real-time realistic landscape visualisation, and a server-client model that allows users to specify an area on a 2D map and automatically generates a realistic 3D landscape model based on area selection anywhere in Australia. The VR application allows multiple users to explore the environment and use it as a discussion platform. In the long term, this system can educate the general public about the consequences of their interactions with their own landscape, and offer a tool to develop policies that will provide beneficial landscape management for a desired future. In the future, this system could be deployed on mobile devices, and via GPS tracking and augmented reality people actually present in a study area could be integrated into the collaboration system.

6 References

Allen, R.K. (2001): INDEX: software for community indicators. In *Planning Support Systems: integrating geographic information systems and visualization tools*, eds. R. K. Brail, and R. E. Klosterman. Redlands, CA, ESRI Press: 229-261.

Herwig, A., and P. Paar. (2002): Game Engines: Tools for Landscape Visualization and Planning? In *Trends in GIS and Virtualization in Environmental Planning and Design, Proceedings at Anhalt University of Applied Sciences,* eds. E. Buhmann, U. Nothhelfer, and M. Pietsch. Heidelberg, Wichmann: 162-171.

Klosterman, R.K. (2001): The What if? planning support system. In *Planning Support Systems: integrating geographic information systems and visualization tools*, eds. R. K. Brail and R. E. Klosterman. Redlands, CA, ESRI Press: 262-284.

Kwartler, M. and R. N. Bernard (2001): CommunityViz: an integrated planning support system. In *Planning Support Systems: integrating geographic information systems and visualization tools*, eds. R. K. Brail and R. E. Klosterman. Redlands, CA, ESRI Press: 285-308.

Stock, C. and I. D. Bishop (in press): Helping rural communities envision their future. In *Visualization for Landscape and Environmental Planning: technology and applications*, eds. I. D. Bishop and E. Lange. Taylor and Francis, Oxford: 145-51.