

# Virtual UNR Campus: The Specification Process

Sergiu Dascalu, Frederick C. Harris, Jr.  
Matthew Knaus, Robert Larmore, Gianpaulo Sorreta, Devin Connell

Department of Computer Science and Engineering  
University of Nevada, Reno  
Reno, NV, 89557, USA  
{dascalu, fredh}@cs.unr.edu

## Abstract

*The Virtual UNR Campus (VCam) presented in this paper is an interactive environment where users explore a 3D representation of the University of Nevada, Reno (UNR) campus. In essence, VCam gives users a visual reference of a real-world location and displays useful information such as building names and directories. Various 3D models of campus buildings lay the foundation for the virtual environment where one can freely explore from a first person perspective. The models are realistic and anyone who has seen the physical buildings can easily recognize their virtual representation. For example, a student could walk around campus and locate in which building his or her class is scheduled. Information is provided visually, audibly, or textually depending on how the user interacts with the environment. A person using VCam can listen to a computer character and choose options such as getting directions or following a guided tour. This paper presents the main aspects of the Virtual UNR Campus's UML-based specification, gives details of the current development status, and points to a series of possible extensions that we intend to investigate in the near future.*

**Keywords:** Virtual campus, virtual environment, 3D modeling, requirements specification, use cases, scenarios, UML.

## 1 Introduction

The Virtual UNR Campus (VCam), currently under development, is a 3D environment that can be explored by the user in a first person perspective. The focus of this project has been on creating a

simulation where information is retrieved through interaction with the objects in that environment. Due to its nature, a virtual environment has no boundaries and VCam is no exception. The functionality of this project includes displaying information on campus buildings such as directories, history, and location. The program allows users to interact and retrieve information from the environment which will familiarize them with the depicted location. Thus, via an interactive simulation, VCam makes available detailed information on a real-world location which in our solution is the University of Nevada, Reno (UNR) campus.

There are similar products commercially available such as Geosimcities [1]. This is a similar project to ours and has many excellent features such as photorealism and navigation. Some of the underlying features found in the GeoSim system will also be implemented in VCam, but not necessarily by using the same techniques. Similar features that will be implemented in VCam include multiple methods for navigation and the concept of modeling a real world location. We have developed VCam with the goal of being innovative and the ability to expand the solution without limitations.

For development purposes we have followed basic software engineering practices for representing the structure of VCam. The specifications of VCam's requirements as well as the descriptions of

use cases and scenarios have been written following the guidelines presented in [2]. We have found that by following a rigorous development process and by relying on UML as a modeling notation, both VCam's functionality and overall architecture have not only been better understood, defined, and represented but also better planned to accommodate future requirements.

This paper, in its remaining part, is organized as follows: Section 2 presents a brief description of VCam, Section 3 describes the main functional and non-functional requirements of the virtual environment, Section 4 includes details on use cases and scenarios, Section 5 presents the statechart diagram of VCam, Section 6 describes the current status of the VCam environment, and Section 7 completes the paper by pointing to several planned directions of future work.

## 2 General Description

VCam is an interactive 3D environment that provides the means for exploring a location without physically traveling there and viewing detailed information about objects in the simulation. The information is provided using text, video, and audio formats.

The two primary parts of the VCam project are the environment which includes building models and other various details and the information about each interactive object. Pictures taken with a digital camera have been used as a reference for modeling the buildings and for providing realistic details of campus. The resource given in [3] was used as a guideline for texturing and creating 3D models. In order to make the program run efficiently each building was drawn multiple times with different levels of detail. The users do not need to see the intricate details of buildings that are not within their viewing distance. Buildings within a close proximity to the user will be

more detailed than those that are farther away. VCam uses the Torque game engine [4] for integrating the various models into an integrated environment.

VCam displays information when the user interacts with objects in the simulation. Examples of *objects* include bulletin boards, building directories, hallway posters, maps, and so forth. A symbol near an object lets the user know that he or she can view information about that object. When the user navigates his or her character into the information symbol a GUI dialog appears on the screen with the accompanying data about that object. The dialog is closed when the user navigates away from the information symbol. Displaying information without making the user type any text is intuitive and convenient. In addition to textual information, the user can also interact with the environment's help character (a "virtual guide," or interface agent, with human appearance and behavior). The guide, currently named Matt, is turned on when the user types the "h" key which will load the character and locate the user who needs assistance. When the guide approaches the user a GUI dialog is displayed with several options. The user can for instance select "directions" and the guide will audibly list all the buildings and wait for the user to select one from a list in the GUI dialog. When the user selects an item from this list, Matt the guide takes him or her to the item's location in the simulation.

## 3 Requirements Specification

Using the notations presented in [2], the requirements for VCam, structured as functional and non-functional requirements, are specified in the next two subsections.

### 3.1 Functional Requirements

The main functional requirements of VCam are the following:

1. The virtual campus shall simulate the UNR campus in 3D, including models of buildings, trees, cars, and people.
2. The virtual campus shall provide the user the ability to explore by flying.
3. The virtual campus shall provide the user the ability to explore by walking.
4. The virtual campus shall have a GUI interface.
5. The virtual campus shall provide the user the ability to select buildings and other objects to view details such as directories, building descriptions (including history of the buildings), bulletin board fliers, and similar items.
6. The virtual campus shall have AI for people.

### 3.2 Non-Functional Requirements

The most important non-functional requirements for VCam are the following:

1. The virtual campus shall have the ability to run on a Windows machine.
2. The virtual campus shall have the ability to run on a UNIX machine.
3. The virtual campus shall support interaction with the mouse and keyboard.
4. The virtual campus shall be written in C++ and C#.
5. The virtual campus shall utilize the Torque gaming engine.
6. The virtual campus shall be usable on most new computer systems.
7. The virtual campus shall include a facility for communication over LAN.

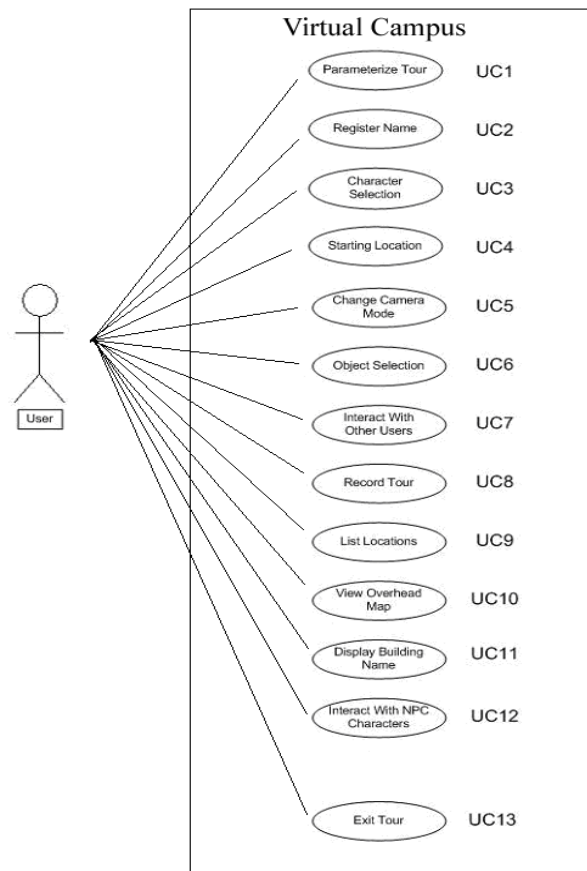
## 4 Use Case Modeling

As part of the formal modeling process, the functionality of VCam has been defined using use cases and scenarios. The entire functionality of VCam is captured in the use case diagram shown in Figure 1 at a high level of abstraction. Several parts of the use case diagram that describe VCam's

functionality are presented in separate use cases shown in Figures 2 and 3. Examples of specific ways of using the software are provided as scenarios in Figures 4 and 5. A use case identifies the actors involved in an interaction and names the type of interaction, whereas a scenario is just an instance of a use case [5]. We have relied extensively on use cases and scenarios to define the expected functionality of VCam and better describe its intended behavior.

### 4.1 Use Case Diagram

The use case diagram displayed in Figure 1 shows the interaction between a user and the VCam program. The use case diagram shown in Figure 1 depicts the interactions between the main user and the VCam software system.



**Fig. 1:** Use Case Diagram for VCam

First, the main user will parameterize the video, audio, and controller options of the tour. Then, a user name will be registered. From there the user will choose a character and the tour begins when the user selects a starting location. Once the user is in the virtual world he or she can interact with objects, list locations in the tour, display and record information, or exit the tour.

#### 4.2 Detailed Use Cases

Because of space limitations and the large number of use cases in our diagram, only two detailed use cases and two example scenarios are presented in this paper. Figure 2 shows the “Object Selection” use case, where the user interacts and obtains information from various buildings and objects in the tour. Figure 3 depicts the “Interact With NPC Characters” use case, in which the user interacts with computer controlled NPCs to take a tour or obtain information.

#### 4.3 Scenarios

For the “Interact With NPC Characters” use case shown in Figure 3, multiple scenarios are possible. For illustration purposes, Figure 4 presents the primary scenario and Figure 5 shows one of the many possible secondary scenarios of this use case.

### 5 State Chart

The statechart diagram of VCam is presented in Figure 6. This diagram has been developed iteratively along the guidelines suggested in [2]. Specifically, we have developed a statechart that shows the overall functionality of the system but this diagram could easily be broken down for individual reactive objects in the system. This statechart diagram was used as a guide for

<b>Use Case:</b> Object Selection
<b>ID:</b> UC6
<b>Actor:</b> Main User
<b>Precondition:</b> <ol style="list-style-type: none"> <li>1. The user is in tour</li> <li>2. No other objects are currently selected</li> </ol>
<b>Flow of Events:</b> <ol style="list-style-type: none"> <li>1. This use case starts when the user enters the virtual tour</li> <li>2. When the user navigates into an information point or clicks on a selectable object, a series of options appear on-screen for the user to choose from</li> <li>3. The user can click on various buttons to obtain information about the location or object and other histories</li> </ol>
<b>Secondary Scenarios:</b> The user cancels or exits interaction with object
<b>Postcondition:</b> Once the user exits information window, he or she can navigate elsewhere in the world

**Fig. 2:** VCam Use Case “Object Selection”

our implementation and gave us a graphical layout of the cycles and behavior of our system. Figure 6 shows VCam’s statechart diagram, which is a high level description of our software including details on transitions, cycles and response to external events.

### 6 Current Status

At this point in time, we have completed a virtual representation of the South UNR campus which includes over 20 buildings. We are working on making the virtual environment photorealistic and adding more functionality. By modeling details such as plants, kiosks and other items we will

<b>Use Case:</b> Interact With NPC Characters
<b>ID:</b> UC12
<b>Actor:</b> Main User
<b>Precondition:</b> <ol style="list-style-type: none"> <li>1. The user is in tour</li> <li>2. No other objects are currently selected</li> </ol>
<b>Flow of Events:</b> <ol style="list-style-type: none"> <li>1. This use case starts when the user enters the virtual tour</li> <li>2. When the user presses the ‘help’ key on the keyboard, a NPC will approach the player</li> <li>3. The NPC will prompt the user with options to take a full tour or not or for other information</li> <li>4. If the user selects the full tour, the NPC will proceed to the first location on the tour and the tour will commence</li> <li>5. If the user selects to obtain information, more menus are prompted to enable the user to get more information</li> <li>6. The user can leave the tour or NPC at any time</li> </ol>
<b>Secondary Scenarios:</b> The user cancels during the first prompt of the NPC User Walks away from the NPC during the tour
<b>Postcondition:</b> The user can navigate around the world and select and navigate into other objects

**Fig. 3:** VCam Use Case “Interact With NPC Characters”

accomplish our goal of making VCam photorealistic. In addition to adding detail to our simulation, we are working on enhancing the environment’s functionality. We have implemented the first version of a virtual guide character that talks to the user. The guide is able to take the user to selected buildings and read information about them. We are also working on modeling the inside of a building where the

<b>Scenario for Use Case</b> Interact With NPC Characters
<b>Primary Scenario</b>
<b>Scenario ID:</b> S12.1
<b>Actor:</b> Main User
<b>Precondition:</b> <ol style="list-style-type: none"> <li>1. The user is in tour</li> <li>2. No other objects are currently selected</li> </ol>
<b>Flow of Events:</b> <ol style="list-style-type: none"> <li>1. When the user enters the tour, a designated help button can be pressed on the keyboard at anytime, as long as no other objects are currently selected</li> <li>2. Once the user presses the help button, a NPC that is already located in the world will find and travel to the current location of the user.</li> <li>3. Once the NPC reaches the user, it will prompt the user with several options</li> <li>4. The options include taking a full tour, general help information, and to exit the help screen</li> <li>5. We assume the user selects the full tour</li> <li>6. The NPC then proceeds to the first location of the tour and reads the information to the user through audio clips</li> <li>7. The user must follow the NPC to complete the tour.</li> </ol>
<b>Postconditions:</b> The user can navigate the world and select other objects or get help again

**Fig. 4:** Primary Scenario of the “Interact With NPC Characters” Use Case

user will be able to see information about what offices professors are located in and see what is posted on an interactive bulletin board. The user will be able to view a scaled

<b>Scenario for Use Case</b> Interact With NPC Characters
<b>Primary Scenario</b>
<b>Scenario ID:</b> S12.3
<b>Actor:</b> Main User
<b>Precondition:</b> <ol style="list-style-type: none"> <li>1. The user is in tour</li> <li>2. No other objects are currently selected</li> </ol>
<b>Flow of Events:</b> <ol style="list-style-type: none"> <li>1. When the user enters the tour, a designated help button can be pressed on the keyboard at anytime, as long as no other objects are currently selected</li> <li>2. Once the user presses the help button, a NPC that is already located in the world will find and travel to the current location of the user.</li> <li>3. Once the NPC reaches the user, it will prompt the user with several options</li> <li>4. The options include taking a full tour, general help information, and to exit the help screen</li> <li>5. We assume the user selects the full tour</li> <li>6. The NPC then proceeds to the first location of the tour and reads the information to the user through audio clips</li> <li>7. The user must follow the NPC to complete the tour.</li> <li>8. The user chooses not to follow the NPC and to go elsewhere</li> <li>9. The NPC will proceed without the user</li> <li>10. The user can select other objects</li> </ol>
<b>Postconditions:</b> The user can navigate the world and select other objects or get help again

**Figure 5** Secondary Scenario of the “Interact With NPC Characters” Use Case

bulletin board where he or she can click on fliers and view a larger image. More details on the design, implementation, integration, and testing of VCam are beyond the scope of this paper. For more information on VCam, the reader is invited to have a look at the project’s website [6].

## 7 Conclusions

The VCam simulation whose specification has been presented in this paper is an innovative solution where users can experience and interact with a location without having to physically travel there. The focus of VCam has been on creating an environment that users can freely explore and use as a visual reference VCam is a complex project that investigates methods for displaying information in a 3D environment and provides rich opportunities for research and development extensions.

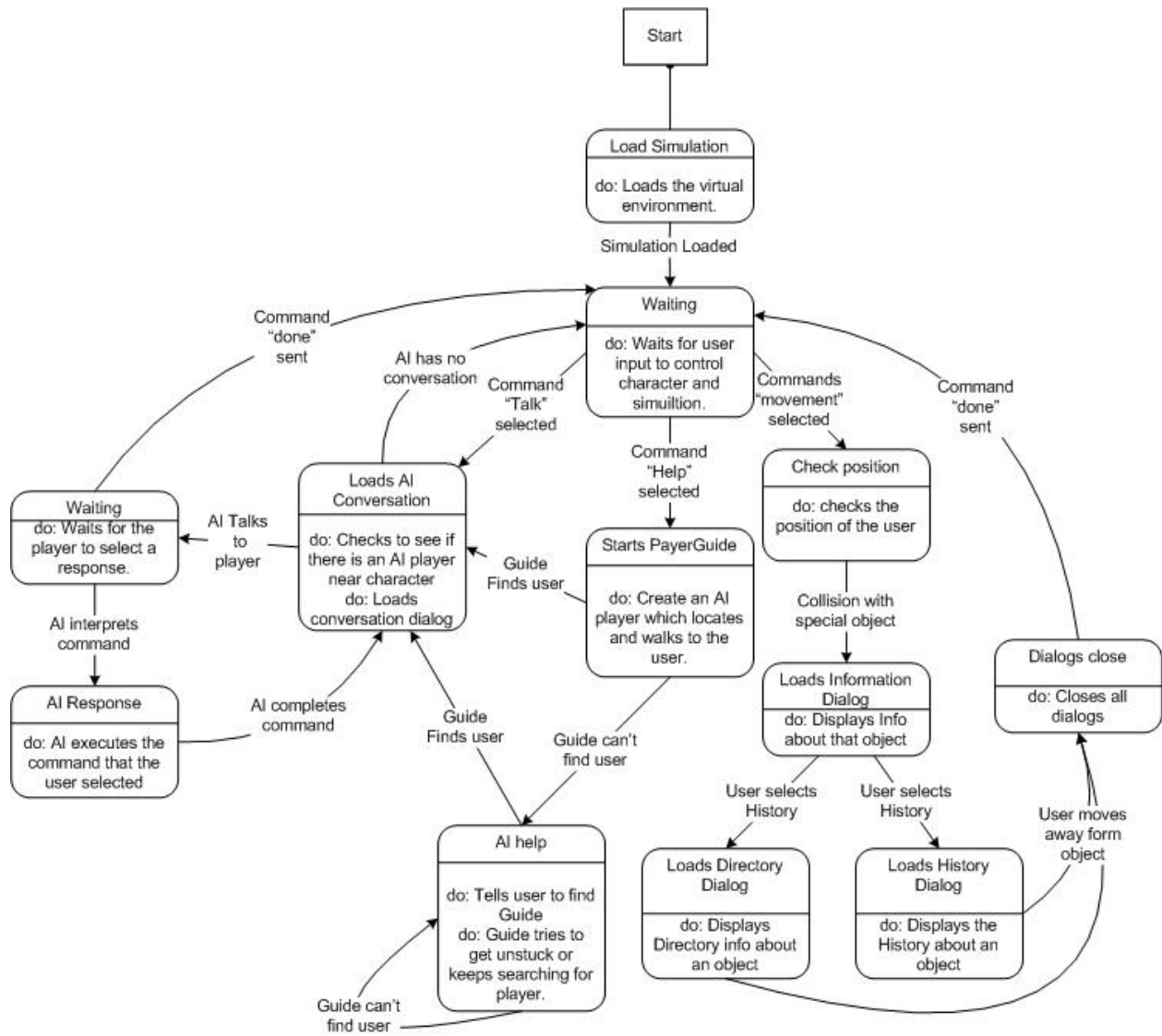
The possibilities for enhancing our simulation are countless and could involve industries such as tourism, real-estate or even government applications, including defense-related. For the tourism industry a location could be modeled and tailored to specific customers. When considered in the scope of real-estate VCam could be expanded to include property that is currently on the market and have contact information for interested virtual parties. Government applications could be defense-related or addressing the local level of city development. Finally, VCam could evolve into a full-sized virtual city incorporating on a larger scale all the possibilities listed above.

## References

- [1] Geo Simm Systems, accessed March 1, 2004 at <http://www.geosimcities.com/>
- [2] J. Arlow and I. Neustadt, *UML and the Unified Process: Practical Object-Oriented Analysis & Design*, Addison-Wesley, 2002.

- [3] 3D Café's Free Tutorials, accessed May 2, 2004 at [www.3dcafe.com/asp/tutorial.asp](http://www.3dcafe.com/asp/tutorial.asp)
- [4] Torque Game engine documentation, accessed April 20, 2004 at <http://www.garagegames.com/>

- [5] OMG's UML Resource page, accessed April 20, 2004 at <http://www.omg/uml/>
- [6] VCam/Virtual Reno Team 14's project website, available as of May 3, 2004 via <http://www.cs.unr.edu/~dascalus/cs426/>.



**Fig: 6** VCam Statechart Diagram