**System/Subsystem Design Description**

**for the**

**UMBC Virtual Tour 2.0 System**

**Document # CMSC447-05-FA2018-G03-SSDD-01A**

Revision A

20 November 2018

Table of Contents

[**0.1 20 November 2018: CMSC447-05-FA2018-G03-SSDD-01A 3**](#_Toc530229090)

[1 Scope 4](#_Toc530229091)

[**1.1 Identification 4**](#_Toc530229092)

[**1.2 System Overview 4**](#_Toc530229093)

[**1.2.1 Purpose 4**](#_Toc530229094)

[**1.2.2. Development History 4**](#_Toc530229095)

[**1.2.3 Deployment Locations 4**](#_Toc530229096)

[**1.3 Document Overview 4**](#_Toc530229097)

[2 Referenced Documents 5](#_Toc530229098)

[3 System-Wide Design Decisions 5](#_Toc530229099)

[**3.1 3D Engine Selection 5**](#_Toc530229100)

[**3.2 User Access Mode Selection 5**](#_Toc530229101)

[**3.3 User Characteristics 6**](#_Toc530229102)

[**3.4 Constraints 6**](#_Toc530229103)

[**3.5 Assumptions and Dependencies 6**](#_Toc530229104)

[4 System Architectural Design 7](#_Toc530229105)

[**System Functions 7**](#_Toc530229106)

[**4.1 System Components 10**](#_Toc530229107)

[**4.1.1 Virtual Unity Engine (VUE) CSCI 10**](#_Toc530229108)

[**4.1.2 Virtual Tour Interface (VTI) CSCI 11**](#_Toc530229109)

[**4.2 Concept of Execution 13**](#_Toc530229110)

[**4.3 Interface Design 13**](#_Toc530229111)

[**4.3.1 Interface Identification and Diagrams 13**](#_Toc530229112)

[5 Requirements Traceability 13](#_Toc530229113)

[6 Notes 14](#_Toc530229114)

[**6.1 Background and Rationale 14**](#_Toc530229115)

[**6.2 Glossary 14**](#_Toc530229116)

[A Appendixes 15](#_Toc530229117)

**0 Revision History**

**0.1 20 November 2018: CMSC447-05-FA2018-G03-SSDD-01A**

Release A contains the basic system/subsystem design description for the UMBC Virtual Tour 2.0 System.

**1 Scope**

This design description presents the designs used or intended to be used in implementing version 1.0 of a software application enabling virtual tours of the UMBC campus. The designs follow the the requirements identified in the Software Requirements Specification for this project (CMSC447-05-FA2018-G03-SSDD-02A).

**1.1 Identification**

Title: UMBC Virtual Tour 2.0

Abbreviation: VT2

Version Number: 1.0

**1.2 System Overview**

### 1.2.1 Purpose

The purpose of the VT2 system is to improve the existing basic UMBC virtual campus tour applications by importing the UMBC campus map and building information into the Unity game engine, enabling users to explore the campus freely in 3D. Additionally, the system offers other useful features, such the ability to highlight valid parking locations on campus based on user status. The intended users of the system are prospective students seeking to familiarize themselves with the campus environment and current students, faculty, and visitors trying to find their classes or event venues and seeking the best place to park.

### 1.2.2. Development History

Development of the system began in September 2018, with a prototype of version 1.0 of the system scheduled for completion in early December 2018. The project is sponsored by the UMBC Department of Computer Science and Electrical Engineering, and the development team consists of senior computer science majors at UMBC. If successful, the project will be acquired by UMBC and incorporated into university’s website in the future.

### 1.2.3 Deployment Locations

The only planned operating site for the software is the UMBC main campus located in Baltimore, Maryland. During the next phase of development, however, the software will be extended to include the UMBC campus at the Universities at Shady Grove, located in Rockville, MD.

**1.3 Document Overview**

This document is organized as follows: Section 1 identifies the scope of this document and lists the definitions, abbreviations, acronyms, and references used therein. Section 2 provides an overview of the system and a brief description of its architecture.

**2 Referenced Documents**

The following standards apply:

CMSC447-05-FA2018-G03-SRS-02A https://github.com/noahj1/UMBC-VT-2.0

IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications

MIL-STD-498 Military Standard Software Development and Documentation

UMBC Style Guide https://styleguide.umbc.edu/

Unity User Manual (2018.2) https://docs.unity3d.com/Manual/index.html

WebGL Manual https://docs.unity3d.com/Manual/webgl.html

**3 System-Wide Design Decisions**

This section details significant design decisions and other issues related to the design of the VT2 system.

**3.1 3D Engine Selection**

To meet the requirement of rendering the UMBC campus in 3D, the development team decided to select a commercially available 3D game engine. Game engines are designed to render realistic 3D worlds and are therefore suitable for the needs of the VT2 project. The development team selected Unity (developed by Unity Technologies) to create the 3D UMBC campus environment for several reasons. First, by some measures, it is the most widely used game engine in the world today[[1]](#footnote-1). The system’s current developers are familiar with it, and future developers working on the project will be more likely to have experience with Unity than with a less well-known engine. Second, Unity provides strong support for browser-based access, as will be dicussed further below. Finally, Unity offers free licenses for students and academic institutions.

**3.2 User Access Mode Selection**

In order to satisfy requirements for flexibility, availability, and maintainability, the design employs a browser-based client-server architecture. The code for the customized Unity engine will be stored on a web server that users access over the internet through a web browser. The server will take advantage of the WebGL framework to enable rendering of the Unity engine’s 3D graphics within the client’s web browser. This process is explained in greater detail in section 4.

The alternative to a browser-based clienter server architecture would be to make the customized Unity world available to users for download. Users would first have to download a copy of the Unity engine, then import the customized world. While this approach would make the system available offline and avoid potential network bandwidth and congestion issues, it would have a number of serious disadvantages. For example, users who otherwise would likely have no reason to download and install the Unity engine (a 9 Gigabyte) would have to do so just to run the software and would have to worry about updating it to the latest version. Additionally, users would have to download a new version of the VT2 software every time the design team makes an update to it. For these reasons, the browser-based client-server architecture is the best option to meet project requirements.

**3.3 User Characteristics**

Users of this system are assumed to possess basic familiarity with internet browsers and websites, but no other special knowledge or skills are required. All user interaction with the system will take place through browser-based menus, and the system will prompt the user to take action with clear and simple instructions when necessary.

**3.4 Constraints**

According to the Unity manual (docs.unity3d.com/Manual/webgl-browsercompatibility.html), the WebGL framework that allows the Unity engine to be run within a web browser is not currently supported on mobile devices because most mobile devices are not powerful enough and lack sufficient memory to adequately support Unity WebGL content. The content may work on high-end devices, but by default, Unity WebGL displays a warning message when a user tries to load content on a mobile device. Therefore, version 1.0 of the VT2 system will not support mobile devices, and users should acess the system via desktop or laptop computers instead. Future versions of the system will provide support for mobile computing using a different framework. Additionally, users must access the system through a compatible browser as described in the Unity WebGL manual. Most modern browsers (Firefox, Chrome, Safari, Microsoft Edge) are supported.

The system relies on UMBC’s IRC office to provide the object files necessary for creating three-dimensional renderings of the campus buildings. As of late November 2018, the IRC department did not have renderings and textures for some of the buildings. As a result, version 1.0 of the VT2 system will only contain a subset of the 43 buildings that comprise the UMBC main campus.

Finally, version 1.0 of the software will not provide the user with directions to or from parking lots or buildings. It is intended that the system will provide this capability in a future release.

## 3.5 Assumptions and Dependencies

It is assumed that the CSCIs described in this version of the SSDD are the basic CSCIs necessary to meet customer requirements. Once the design team has successfully implemented these CSCIs, the team may proceed with implementation of “reach goals” such as direction-finding algorithms and mobile support upon agreement with the customer.

The VT2 system will also contain links to the main UMBC website (www.umbc.com). If the UMBC website were to become unavailable, some features of the software would cease to work, but the system’s core functionality would be unaffected.

**4 System Architectural Design**

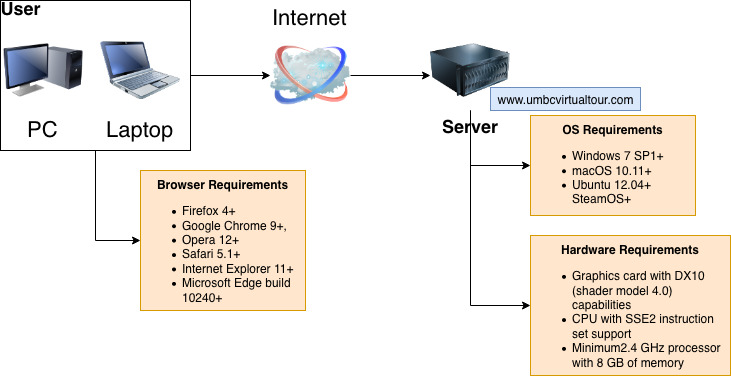


Figure 2 System Hardware Architecture and Requirements

**System Functions**

The system has four primary functions that correspond to the CSCIs specified in section 3:

1. **Virtual Campus Explorer (VCE):** This CSCI shall allow the user to freely explore the virtual UMBC campus map from a chosen starting point.
2. **Virtual Parking Finder (VPF):** This CSCI shall identify parking lots where the user is allowed to park based on the user’s status (faculty member, commuter student, residential student, visitor, etc.).
3. **Virtual Tour Interface (VTI):** This CSCI shall provide a browser-based interface for the VT2 system based on the WebGL framework. The interface shall provide the main menu for the system, allowing the user to select either the VCE function or the VPF function.
4. **Virtual Unity Engine (VUE):** This CSCI shall provide a customized version of the Unity Engine that includes accurate 3D renderings and textures of the UMBC campus buildings. It shall allow natural movement around the campus’ exterior spaces with motion and camera effects generally expected of first and third-person point of view video games.

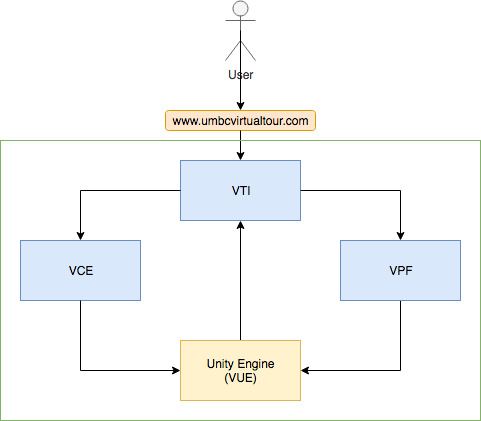


Figure 1 System Architecture

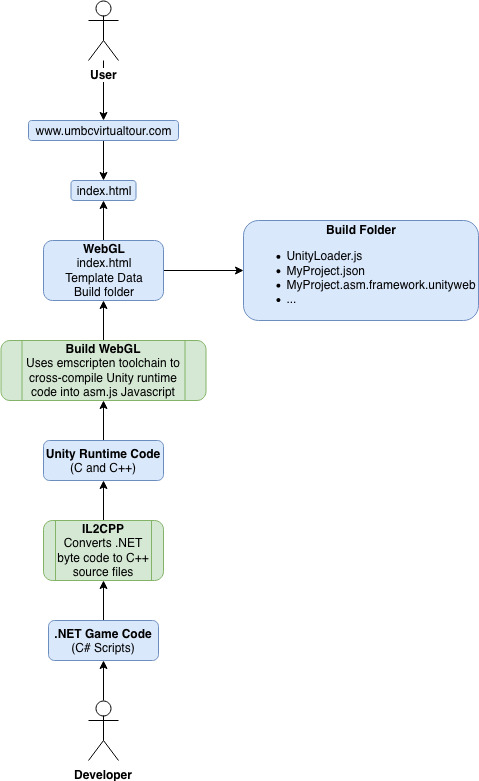


Figure 3 WebGL Framework

The VT2 system consists of a customized Unity engine with a web interface. The customized Unity engine—the VUE CSCI—is incorporates the VCE and VPF CSCIs, while the web interface is provided through the VTI CSCI.

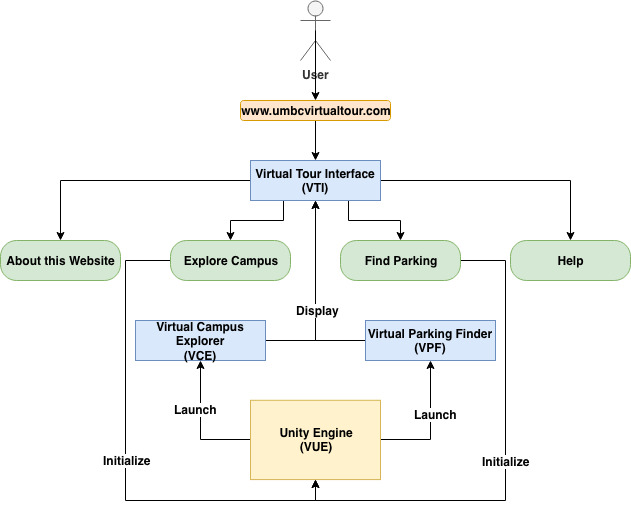


Figure 4 VT2 Overall System Architecture

The following sections explain these components in detail.

## 4.1 System Components

**4.1.1 Virtual Unity Engine (VUE) CSCI**

The Virtual Unity Engine (VUE) CSCI is a customized version of the Unity game engine that provides an explorable 3D rendering of the UMBC campus. The VUE is based on Unity Personal version 2018.2.13 (available at https://store.unity.com/download). The following steps were taken to enhance this version of Unity to create the 3D map of the UMBC campus:

1. Export a map selection containing the UMBC campus from OpenStreetMap (www.openstreetmap.org/) as a .osm file.
2. Convert the OpenStreetMap .osm file to a 3D object model (.obj file) using OSM2World (osm2world.org/).
3. Import the 3D object model of the UMBC map into Unity, creating a basic campus map.
4. Obtain models (.obj and .mtl files) and textures (.png and .tga files) for 25 UMBC campus buildings from the UMBC IRC.
5. Import the models and textures into Unity and add them to the basic campus map.
6. Enhance the map with additional features such as grass, water, and trees to increase realism.
7. Complete the customized Unity engine by adding camera and object height mapping features to make movement and terrain more realistic.

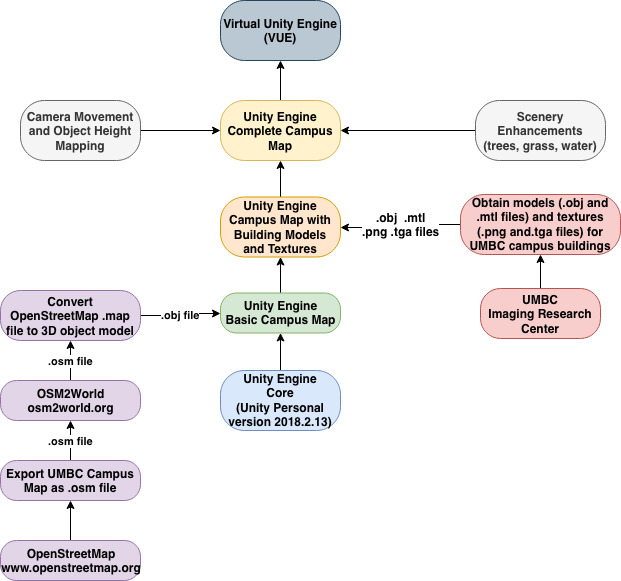


Figure 5 Development of Virtual Unity Engine (VUE) CSCI

**4.1.2 Virtual Tour Interface (VTI) CSCI**

The Virtual Tour Interface (VTI) CSCI provides a web interface to the VUE, VCE, and VPF CSCIs, as well as other basic website functionality. The primary feature of the VTI is a menu system with the following options:

1. Explore Campus
2. Find Parking
3. About this Website
4. Help

4.1.2.1 Explore Campus

4.1.2.2 Find Parking

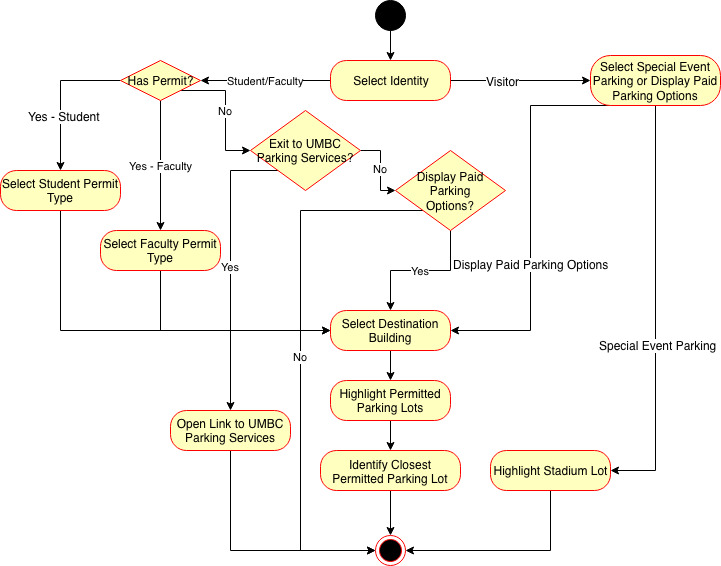


Figure 6 Activity Diagram for Find Parking

4.1.2.3 About this Website

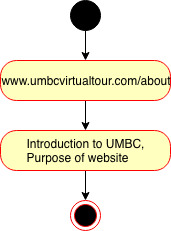


Figure 7 Activity Diagram for About this Website

4.1.2.4 Help

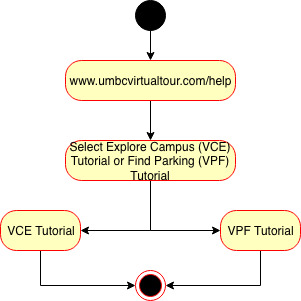


Figure 8 Activity Diagram for Help

## 4.2 Concept of Execution

## 4.3 Interface Design

### 4.3.1 Interface Identification and Diagrams

**5 Requirements Traceability**

**6 Notes**

## 6.1 Background and Rationale

A virtual campus tour is an important component of a university’s strategy for recruiting students. In an environment of intense competition for students of all types—domestic and international, in-state and out-of-state, and undergraduate and graduate—a strong virtual tour application can convince a prospective student to apply or visit the campus in person. Additionally, virtual campus tours can help current students and visitors navigate their way to their classes or special events.

UMBC currently has several websites that nominally offer virtual tours of the campus. The Undergraduate Admissions UMBC Virtual Tour (located at undergraduate.umbc.edu/visit/virtual-tour.php) provides 9 panoramic views of the campus, though it claims to offer 25 views. A virtual tour site for the graduate school (gradschool.umbc.edu/discover/vtour/) simply provides a link to the same site that hosts the panoramic campus views noted above. Additionally, a UMBC undergraduate student created a basic virtual tour mobile application for Android devices in 2014 titled, “Introducing UMBC Tours - A Virtual Campus Tour Experience for Android” (www.youtube.com/watch?v=zRI61jkUDT4). However, this implementation had extremely limited functionality and did not represent a significant improvement on the applications offered on the UMBC website.

The purpose of the UMBC VT2 software described in this design document is to dramatically improve the currently available UMBC virtual tour applications by importing the UMBC campus map and building information into the Unity game engine. It shall allow users to select any location on a three-dimensional map of the campus and allow them to explore it freely. It shall provide browser-based access to this system through a web application. Moreover, the new system shall provide several other useful features, including the ability to highlight valid parking locations based on user status. The system will primarily benefit prospective students seeking to familiarize themselves with the campus environment and current students, faculty, and visitors trying to find their classes or event venues and seeking the best place to park.

## 6.2 Glossary

API Application Programming Interface

CSCI Computer Software Configuration Item

GUI Graphical User Interface

HTML Hyper Text Markup Language

IRC Imaging Research Center

MTL File A Material Library (.mtl) file contains one or more material definitions, each of which includes the color, texture, and reflection map of individual materials. These are applied to the surfaces and vertices of objects and are stored in ASCII format.

OBJ File An object (.obj) file is a standard 3D image format that can be exported and opened by various 3D image editing programs. It contains a three-dimensional object including 3D coordinates, texture maps, polygonal faces, and other object information.

OSM Open Street Map

SIMD Single Instruction, Multiple Data

SRS Software Requirements Specification

SSE2 Streaming SIMD Extensions 2

TCP Transmission Control Protocol

TGA File A Truevision Graphics Adapter (.tga) file is a raster graphics file format that can store raw or compressed images.

UMBC University of Maryland, Baltimore County

Unity The Unity cross-platform game engine

VCE Virtual Campus Explorer CSCI

VPF Virtual Parking Finder CSCI

VTI Virtual Tour Interface CSCI

VT2 UMBC Virtual Tour 2.0

VUE Virtual Customized Unity Engine CSCI

WebGL The Web Graphics Library, a cross platform JavaScript API for rendering 2D and 3D graphics in a web browser

XML Extensible Markup Language

UDP User Datagram Protocol

**A Appendixes**

1. https://venturebeat.com/2017/03/01/game-engine-ceos-talk-past-each-other-when-it-comes-to-statistics/ [↑](#footnote-ref-1)