An Online Art Museum for the IMGD Program

Major Qualifying Project Report completed in partial fulfillment of the Bachelor of Science Degree at

Worcester Polytechnic Institute, Worcester, MA

Submitted By:

Rafael Angelo

Keenan Gray

Peter Salem

Aaron Segal

Kevin Zhao

Project Advisors:

Charles Rich

Ralph Sutter

Abstract

The IMGD Art Museum is designed to exhibit artwork created by students in Worcester Polytechnic Institute's Interactive Media and Game Development (IMGD) program. Using an online form IMGD artists can submit media; specifically images, video, 3D models, animations, and audio. Submitted pieces are then curated and placed within the museum. Visitors to the museum can view the artwork in a virtual space, utilizing affordances not granted in a traditional museum. The museum and submission form are available at *alpheus.wpi.edu/~museum*.

Acknowledgements

Our team would like to thank everyone that made this project possible.

We thank our advisors Ralph Sutter and Charles Rich for being able to meet with our team on a weekly basis and provide us suggestions and directions for the project. Individually, we thank Ralph Sutter for helping provide the art direction for the architecture and color of the museum and ideas on how to improve the museum's layout. We thank Charles Rich for assisting us in project organization and management and giving us suggestions on how to approach certain difficulties.

We thank the WPI Interactive Media and Game Development department for providing us server space on Alpheus to host the museum and the submission page for the museum, as well as providing us with a paid GitHub account to store the museum and submitted art files.

Authorship

All team members contributed equally to the creation of the paper.

Contents

Figures

1. Introduction and Motivation

This project resulted in the creation of a virtual museum for the collection and display of artwork produced by IMGD students. The museum itself was created using Unreal Engine 4. Additionally, a web-based submission system was created to allow artists to submit their work. This project was completed by a five member team of Worcester Polytechnic Institute students. The team was composed of two IMGD majors and three double majors in IMGD and Computer Science.

There were three primary motivations considered while this project was in development. The first objective of this project was to build a culture of creativity within the IMGD community. Having their work displayed in the museum will hopefully inspire students to improve the overall quality of their work. Having pieces accepted to the museum will also reward students for their hard work.

The second motivation of this project is to serve as an additional point of exposure for artists' work and portfolios. WPI's Interactive Media and Game Development program strives to further the education of technical or artistic students in their pursuit of a career in video game development. The program has a strong emphasis towards hands-on, project-based learning. By the time an artist graduates, he or she will have worked on a number of artistic pieces: concept drawings, 3D models, animations, and even completed games. These works are integral to showing off the students' artistic skills to potential employers and are typically stored in an online portfolio. In addition to displaying a student's work, the museum will provide a link to the student's portfolio.

In general, game companies use “just-in-time” recruiting. Just-in-time recruitment involves recruiters finding new talent whenever there is a need of it, instead of following a strict recruitment schedule. The museum hopefully makes finding students with the necessary skills easier for potential employers.

Finally, the museum serves as a collection of quality artwork done by IMGD students. The museum collection can be showcased to young students interested in the IMGD program at WPI. The museum can give prospective students an idea of the work being done in the program and will also serve as a long term archive of quality work completed by IMGD artists.

The project was divided into several components that would need to be created in order to have a fully functional museum. First, artists would need a mechanism to submit their work to the museum. A museum curator would then arrange the submitted artwork within the museum itself. Finally, this museum would be made available to the public, who will be able to access the museum using their computers. Figure 1 shows a graphical process overview of the project that visualizes the roles of each user.

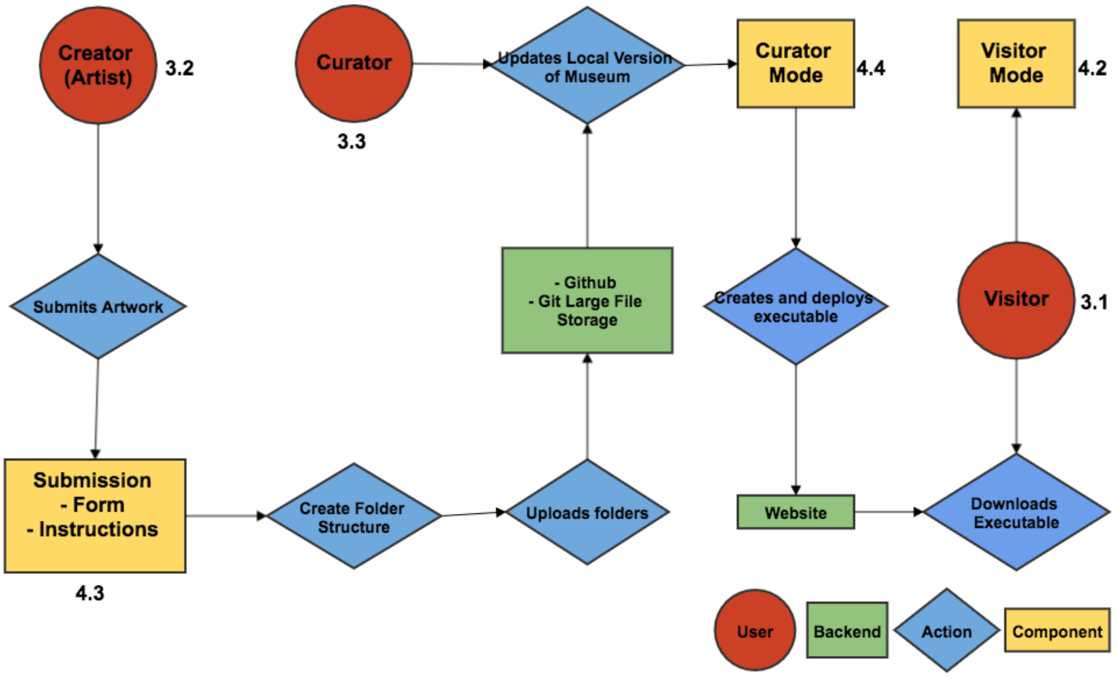


Figure 1 : Process overview – Numbers correspond to sections of this paper where more information about each process can be found.

The overview begins with the artist, who creates work and submits it to the museum through an online form. The online form uploads all submissions to an online repository. The curator then collects submissions from the repository and edits the museum through the use of curator mode. When curation is finished, the curator creates and uploads an executable version of the museum to the project's website. Visitors can then download the executable and explore the museum at their leisure.

The team selected Unreal Engine 4 (UE4), developed by Epic Games, as the engine used to construct the museum. The team decided to use UE4 because it is a free and powerful game development engine that could hopefully provide the tools necessary to complete this project. UE4 readily supports the asset types that the team desired to include in the museum. Also, the engine can be modified, allowing the team to extend the engine's capabilities according to their needs. The submission process will expose artists to Unreal Engine, which has the added benefit of helping students develop skills they would need in a game industry job.

The following chapters of this report will further explain the requirements of this project, the implementation, and challenges faced during development, as well as future directions for the museum.

2. Requirements

2.1 Persistence

The museum would need to hold student work from not only current students, but from students several years into the future. Specifically, the museum and all of its associated services needed to be created with a ten year lifespan in mind. As the museum collection grows, older artwork will be removed and archived to make room for newer pieces. Over time, different people will be delegated as the curator, so the curation system needs to provide a simple interface for modifying the museum.

To determine the required size of the museum for a ten year period, the team estimated the expected number of museum pieces that will be received in a period of five years. A five year cycle allows for an artist to have pieces in the museum from each year of their undergrad, before older work is archived. If the museum is large enough to support five years of work, the curator can keep the museum work up-to-date with work submitted by currently enrolled artists.

2.2 User Experience

The complete museum is utilized by three different categories of users: artists, curators, and visitors. Artists are users who submit their artwork to be displayed in the museum. A curator is a dedicated user that will use Unreal Engine 4 to layout the submitted artwork and fill the museum. A visitor is a user who enters the museum to interact with artwork and explore the museum. Each of these users will interact with the museum in different ways and their experiences needed to be simple and pleasant.

2.3 Support for Multiple Media Types

The museum needed to be able to exhibit artwork of several different media types. Students in the IMGD program work in many kinds of media and the museum needed to support most of these. The supported media types are:

- Images
- Videos
- Audio
- Static meshes
- Animated models

The media supported in the museum is somewhat analogous to real world museums with paintings, photographs, statues, and video and audio recordings. The important distinction in the virtual museum is that all of the pieces are digital and can be interacted with in more ways than in a traditional museum.

3. Design of User Experience

The core philosophy of the museum's design is based on the requirements described above. The museum's design needs to be aesthetically pleasing and simple. Any of the three users (artists, curators, and visitors) that interact with our platform should be satisfied with its use. Furthermore, it is important to keep in mind that the focus of this project is to showcase the artwork submitted by IMGD students and not the museum itself. Therefore, when creating this project, the team had to develop an unobtrusive design that would not obscure student work, but highlight it.

The primary component of this project is visitor mode. Visitor mode is the system that allows users to view the artwork on display in the museum. The majority of people interested in this project are likely to be visitors.

3.1 Visiting the Museum

3.1.1 Museum Background

The museum designed for the IMGD program will be similar to a real world museum in many ways. The team did research into museology, the study of museums, and learned that the IMGD museum will need several specific museum-related features in order to achieve its purpose. The team researched several concepts from the book *Key Concepts of Museology,* in order to gain a better understanding of the depth of this project. The specific concepts the team identified as necessary to the museum are; architecture, a collection, museum objects, and exhibits.

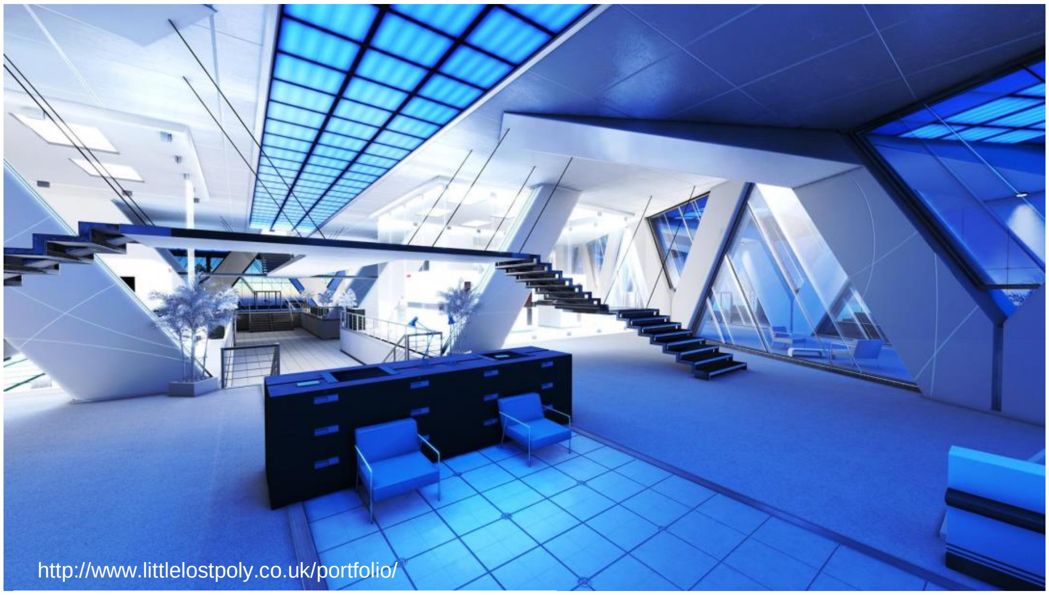
Architecture is the museum space that will be used for the display of artwork and for receiving visitors. For the virtual museum, the architecture includes the virtual space created by the team and the museum application through which visitors access the museum. *Key Concepts of Museology* describes a collection as "A set of material or intangible object which an individual or an establishment has assembled, classified, selected, and preserved in a safe setting" (Desvallées, 2010). The museum collection in this project is the collection of submitted pieces stored online. The museum collection includes images, videos, static meshes, animations, and audio.

Museum objects are portions of a collection that are prepared and displayed in the museum. *Key Concepts of Museology* has a very abstract definition of the museum object. The virtual museum designed for this project creates a more concrete definition of the museum object. The museum object in this case is a mechanism through which submitted artwork can be converted into a format that can be viewed by museum visitors.

Finally, museums often feature many exhibits. An exhibit is defined as, "the container or the place where the contents are on display" (Desvallées, 2010). The team calls the virtual exhibits "wings", which are individual sections of the whole museum. Each wing holds several rooms that have different shapes and physical design. The differences allow the curator to be very creative with positioning art assets and also allows the visitor to have a different experience in every room.

3.1.2 Museum Layout

The work of creating the museum began by gathering reference material. Reference images included pictures taken in real world museums, and screenshots from several games with interesting architecture. In particular, the team was inspired by interior level design of the game *Mirror's Edge*. *Mirror's Edge* features clean, open spaces, an interior from the game can be seen in Figure 2.



http://www.littlelostpoly.co.uk/portfolio/

Figure 2 : Reference image

After looking at reference and speaking as a group the team decided that large open spaces would be the best, allowing many different pieces of artwork to be visible at a given time. The visibility of any individual piece was important and the architecture could not obscure the vision of the visitor. Visitors should appreciate the artwork, as opposed to the architecture, so the architecture is both simple, and aesthetically pleasing to compliment the artwork. Another decision was to have multiple wings. This gives the curator creative control over the wing’s theme. The curator could organize by art style, art type, student year or other themes. Another decision was to use big open rooms, which allows for student artwork to be displayed equally. Figure 3 shows a top-down view of a wing with three large rooms.

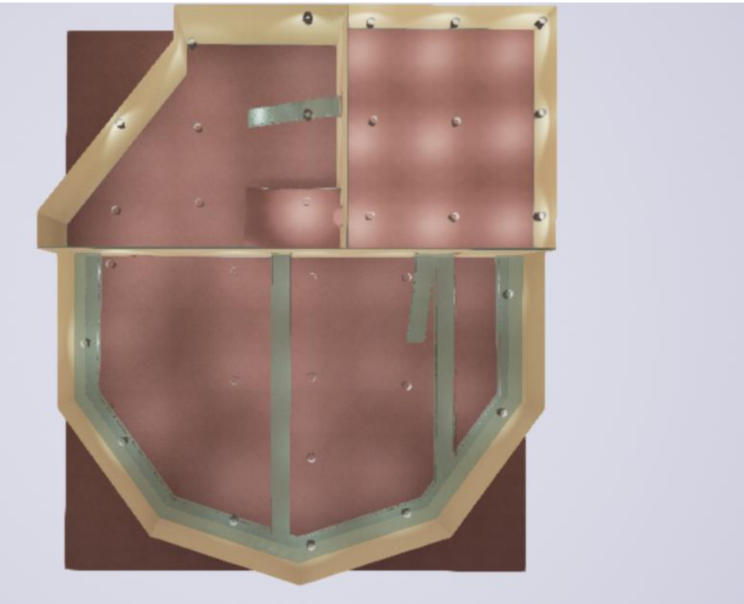


Figure 3 : Top-down view of one of the museum's wings

Many of the wings are tiered and all of the wings are easy to navigate. Easy to navigate wings help the visitor enjoy their time in the museum by ensuring that time spent there does not get stale as well as ensuring that the visitor does not get frustrated when they are exploring the museum. An example of what the visitor sees when exploring the museum can be seen in figure 4.

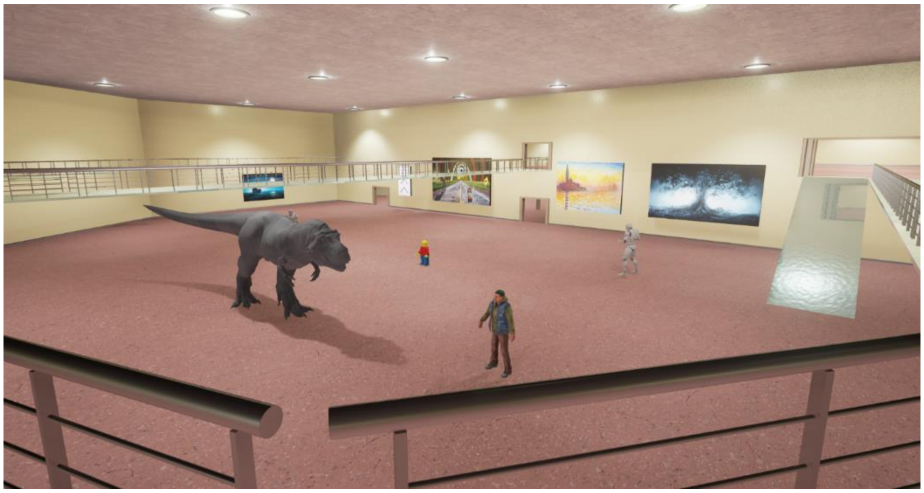


Figure 4 : The completed museum aesthetic

3.1.3 Visitor Features

The visitor uses first person shooter movement controls, WASD keys and the mouse. The control scheme was chosen because it is intuitive and a system that is commonly used in virtual applications. By using this system, the user will be immediately familiar with the navigation, and will be able to begin exploring the museum with as little delay as possible. The user interface within the museum is designed to keep the viewport of the user as clear as possible. Whenever a visitor walks close to a museum piece, the author and artwork information are displayed in 3D space. For 2D pieces, the text appears to the right of the image on the wall, much like a label placed next to a painting in a real-world museum. 3D pieces, including audio pieces, can be approached from any direction. The player should be able to read the label of 3D pieces, from any position. To accomplish this, the position and rotation of the label is adjusted as the player moves around the object. By adjusting the label's position, the text is always legible to the visitor. Figures 5 and 6 illustrate the difference in text label positioning for 2D and 3D pieces.

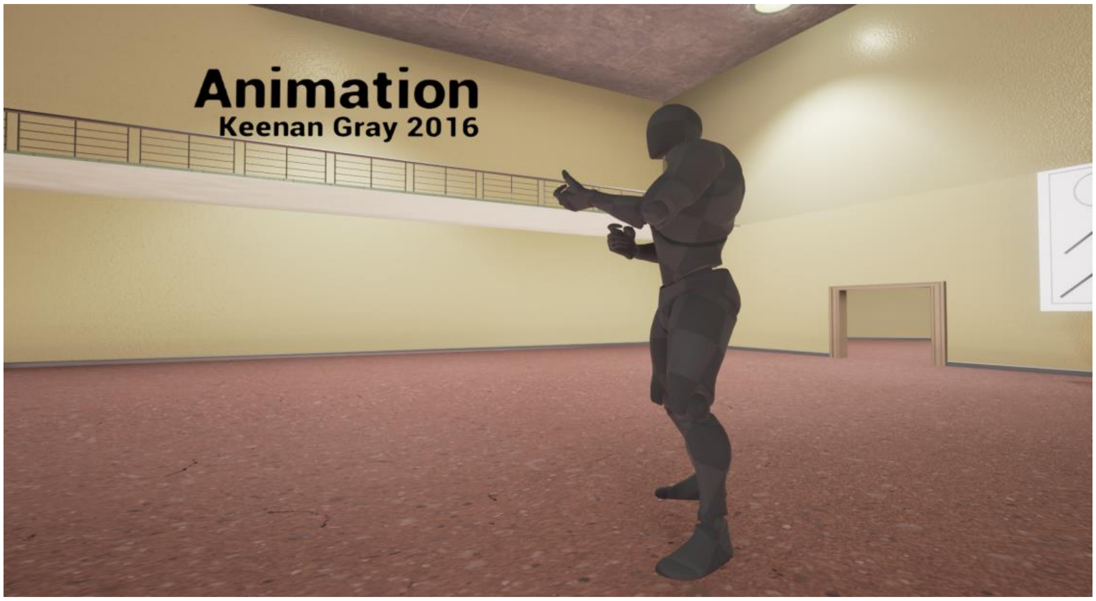


Figure 5 : Text label for 3D piece

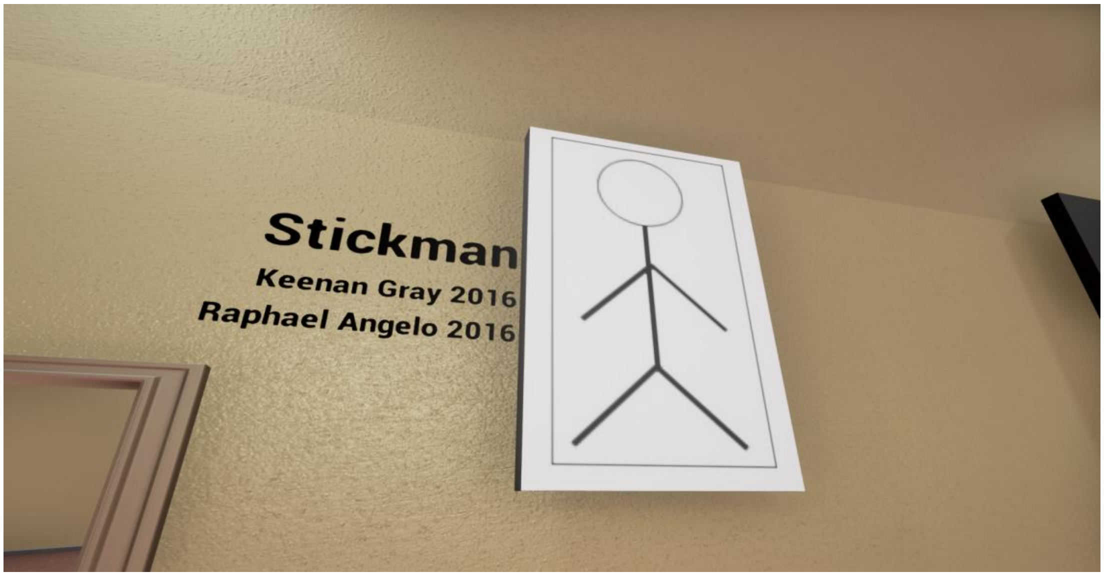


Figure 6 : Text label for 2D piece

For further interaction with museum pieces, the user can select a piece using the left mouse button. Selecting a piece brings up the interaction taskbar. The taskbar contains the title of the piece, the artists' names, and any associated files stored within the piece. For example, an animated model may have multiple animations that can be played. The menu will display the different animations that the visitor may choose from. Visitors can then cycle through various animations of the same 3D model.

With a 3D model or animation selected, the visitor has the ability to view the model's wireframe. Wireframe models are visual representations of 3D models created by drawing lines at the location of each edge connected to the different vertices. This is an additional way for visitors to get a closer look at the fine detail within each piece. The visitor can pause and play videos, animations, and audio.

The taskbar is hidden from view when a piece is not selected. A short animation plays when a piece is selected that brings the taskbar up from the bottom of the visitors view. The taskbar only shows the relevant information to the visitor. For example, if an image is selected, the pause/play button is not shown in the taskbar, and there is no prompt to display the wireframe. Figure 7 shows a selected piece, outlined in gold, and the taskbar menu.

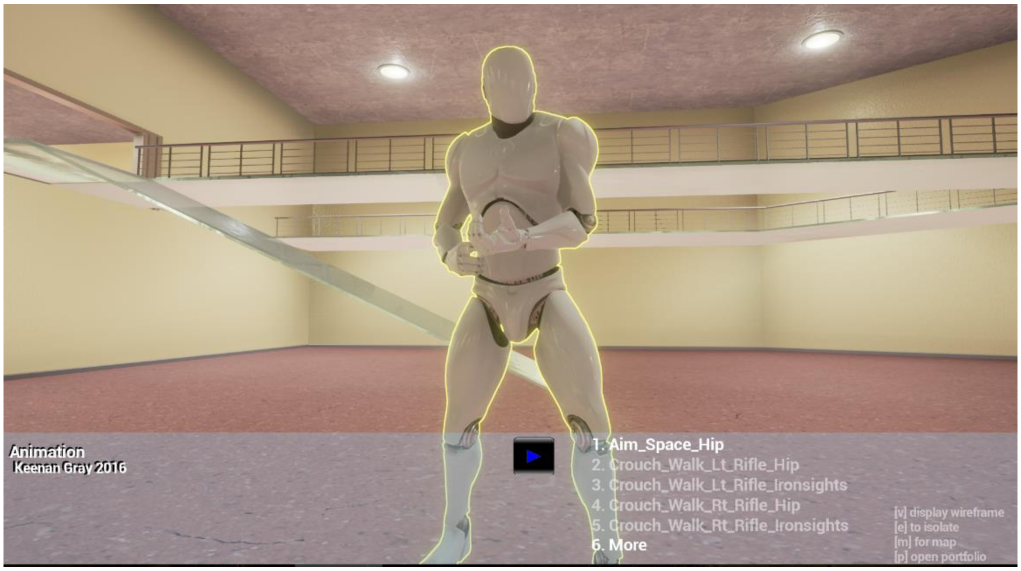
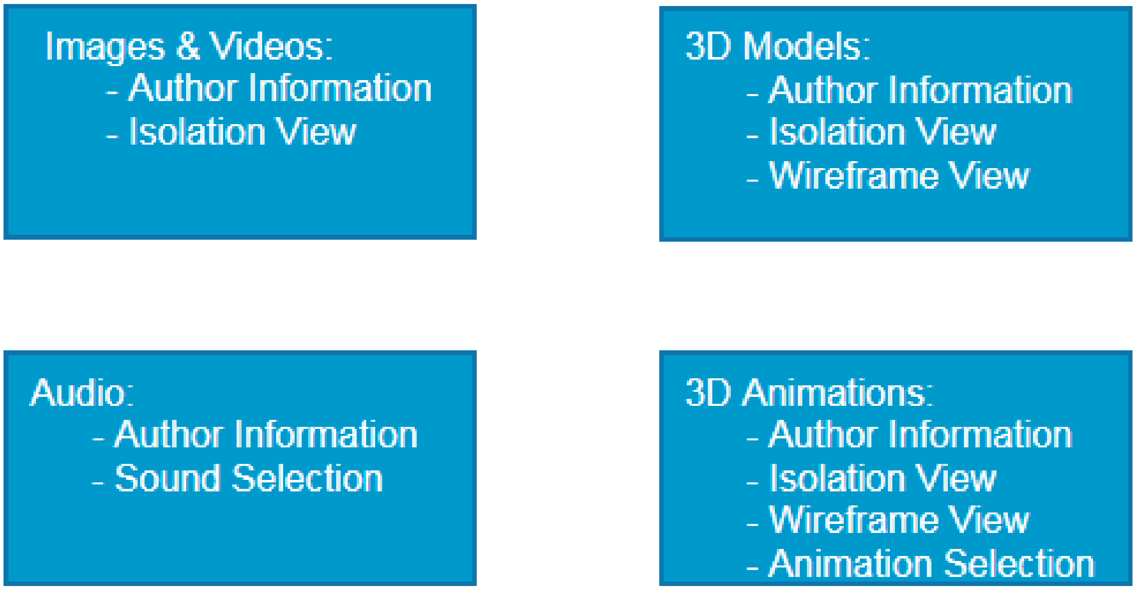

Animation
Keenan Gray 2016
▶
1. Aim_Space_Hip
2. Crouch_Walk_Lt_Rifle_Hip
3. Crouch_Walk_Lt_Rifle_Ironsights
4. Crouch_Walk_Rt_Rifle_Hip
5. Crouch_Walk_Rt_Rifle_Ironsights
6. More
[v] display wireframe
[e] to isolate
[m] for map
[p] open portfolio

Figure 7 : The taskbar menu

Finally, after selecting any piece, the player can press a button to open the artist's portfolio in a web browser. Since the audience of the museum is potential employers, it is important that visitors can access student portfolios, as they contain additional work done by the students and information that may be relevant to the visitor. Figure 8 summarizes the capabilities of the visitor for each selected object.



Images & Videos:
- Author Information
- Isolation View

3D Models:
- Author Information
- Isolation View
- Wireframe View

Audio:
- Author Information
- Sound Selection

3D Animations:
- Author Information
- Isolation View
- Wireframe View
- Animation Selection

Figure 8 : Visitor interaction capabilities

3.1.4 Isolation View

Another major feature of the museum for the visitor is the ability to transition into isolation view. Visitors are able to toggle this view for any asset type, with the exception of audio assets. As the image below demonstrates, the piece is seen in front of a blank background. In isolation view, the visitor can get a better look at a piece by rotating and moving the piece, and zooming in and out. Figure 9 shows an object that has been brought into isolation view. In the image, the taskbar is hidden so that the piece can be seen unobscured. The colored buttons in the bottom right can be used to change the background color if the visitor wishes.

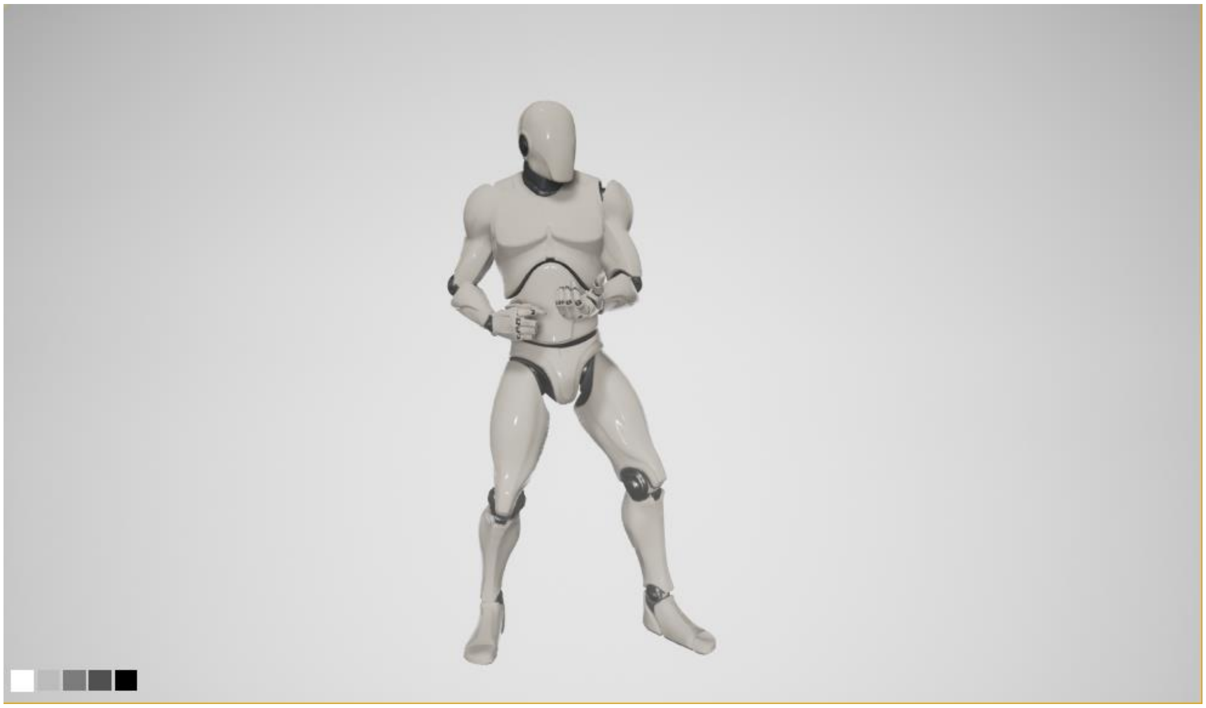
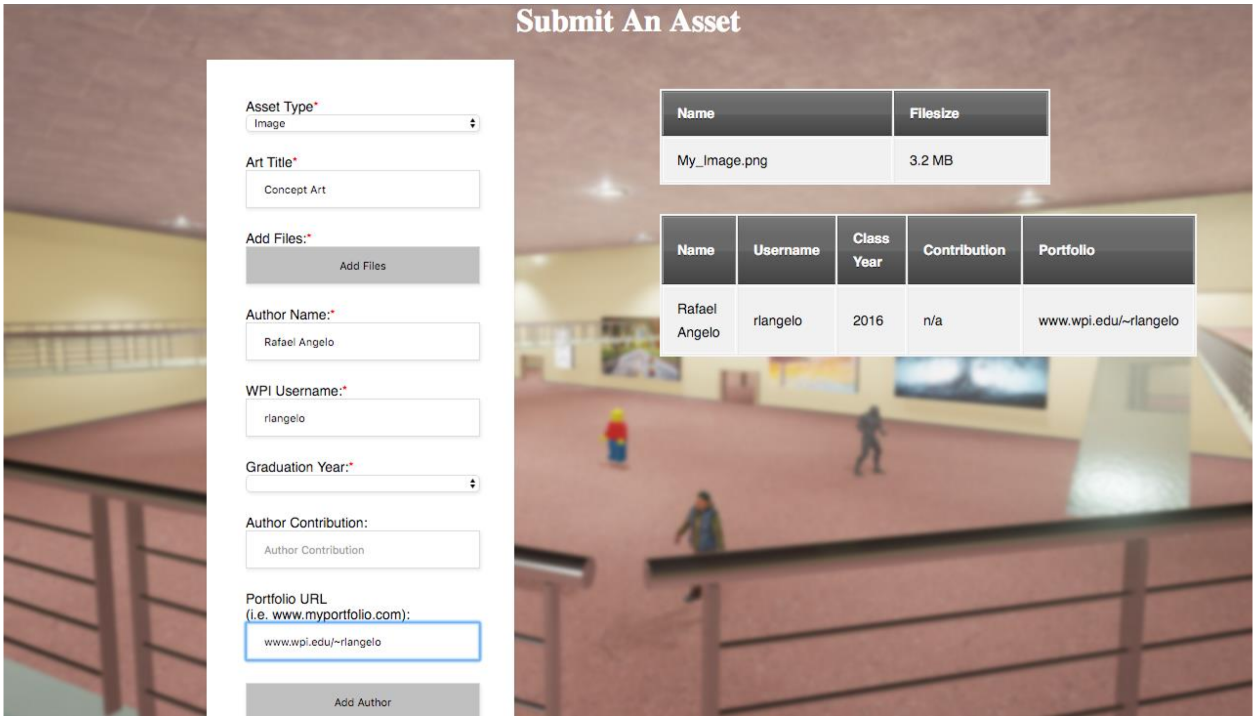


Figure 9 : Isolation View

3.2 Artist Submission Form

All the work within the museum is created by IMGD art students. An online submission form was created in order to allow artists to submit their art to the museum. Once an artist has completed a piece, either as part of a class or in a personal project, they would be able to submit the piece. The requirements specify that the user experience for artists should be pleasant, and with this in mind, the website design needed to be aesthetically pleasing and simple to use. Figure 10 is an image showing the submission form with some information filled out and a test file uploaded.



Asset Type*: Image

Art Title*: Concept Art

Add Files:*

Author Name:* Rafael Angelo

WPI Username:* rlangelo

Graduation Year:*

Author Contribution: Author Contribution

Portfolio URL (i.e. www.myportfolio.com): www.wpi.edu/~rlangelo

Name	Filesize
My_Image.png	3.2 MB

Name	Username	Class Year	Contribution	Portfolio
Rafael Angelo	rlangelo	2016	n/a	www.wpi.edu/~rlangelo

Figure 10 : The submission form

The submission system needs to be able to support any work created by students. The system needs to handle all media types supported in the museum. Some pieces, like images and videos, only require a single file to be submitted. Other pieces however, require multiple files. Static meshes, for example, have a mesh file alongside associated textures and materials. The submission system had to handle these cases. The submission system also supports multiple submissions from the same user and resubmissions. Finally, a reliable storage location was necessary in order to store and access submitted files.

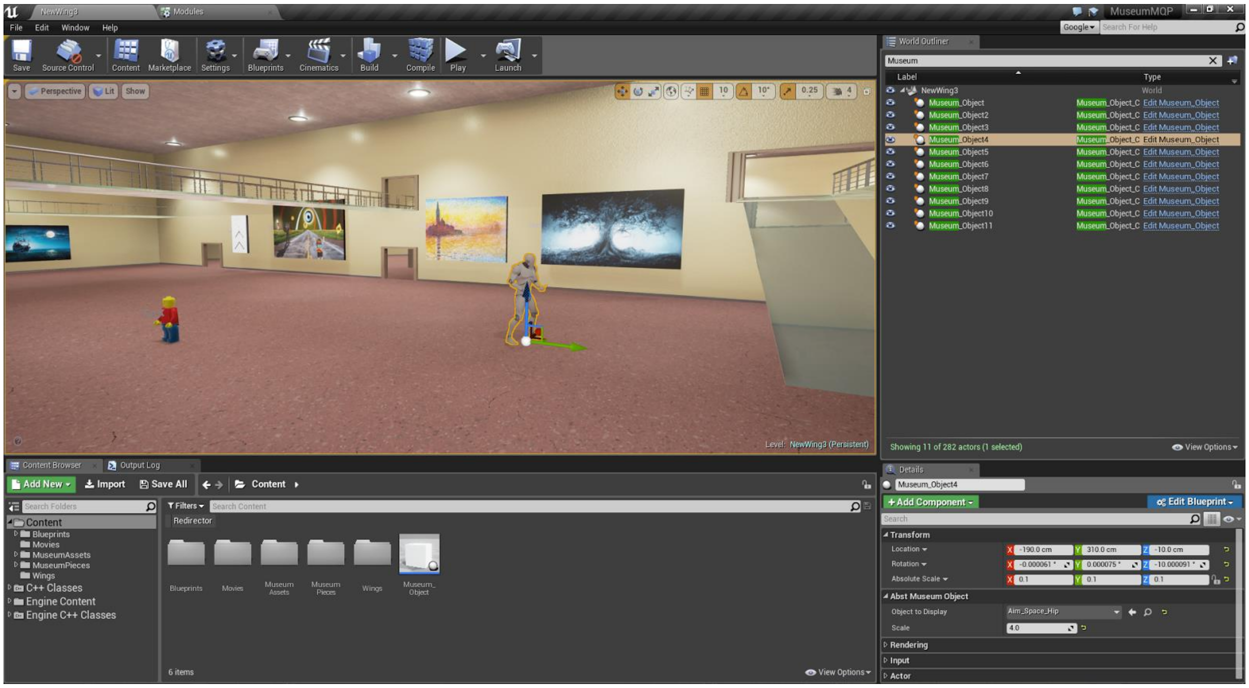
Artists will have a set of instructions they will need to follow in order to submit their art assets to be displayed in the museum. All submitted files need to be in a format compatible with Unreal Engine. Ensuring compatibility makes the submission process for static meshes and animated models more complicated than the process for audio, images, and video. Static meshes and animated models must be imported to Unreal Engine and migrated before they can be submitted to the museum. Having the artists migrate these objects from Unreal also allows artists to ensure that their models will appear correctly in the museum.

The best choice for an easy-to-use submission form would be to create a website, so that artists can access it from anywhere at any time. The submission form would be accessible on the project's website, which should be hosted on a server maintained by the IMGD program. The website where the submission form can be found should also be the same place that users will go to download the executable museum.

3.3 Curation

In order to maintain this project in the long-term, the museum needs a dedicated curator to edit the museum as necessary. On a regular schedule, the curator will update the museum with new artwork. Updating the museum is done using the Unreal Editor. The curator chooses where to place new submissions and which old assets to remove. It is the responsibility of the curator to layout artwork in the museum in an organized and appealing way.

Since the curator will be using the Unreal Editor to make changes to the museum, the curator needs experience using Unreal Engine 4. The engine is very powerful and has an incredible variety of features for game development. However, in order to curate and maintain the museum, the curator will not need access to many of these features. In order to aid the curator as well as to “protect” the museum from any unintentional changes, the team added a “Curator Mode” to the Unreal Editor. The curation mode is an unreal editor layout that allows for the easy manipulation of museum objects and removes from view any features that the curator will not need while editing the museum. Figure 11 shows the curator layout.



Level: NewWing3 (Persistent)

Museum

Label	Type
NewWing3	World
Museum_Object	Museum_Object_C Edit Museum_Object
Museum_Object2	Museum_Object_C Edit Museum_Object
Museum_Object3	Museum_Object_C Edit Museum_Object
Museum_Object4	Museum_Object_C Edit Museum_Object
Museum_Object5	Museum_Object_C Edit Museum_Object
Museum_Object6	Museum_Object_C Edit Museum_Object
Museum_Object7	Museum_Object_C Edit Museum_Object
Museum_Object8	Museum_Object_C Edit Museum_Object
Museum_Object9	Museum_Object_C Edit Museum_Object
Museum_Object10	Museum_Object_C Edit Museum_Object
Museum_Object11	Museum_Object_C Edit Museum_Object

Showing 11 of 282 actors (1 selected)

Museum_Object4

Transform

	X	Y	Z
Location	-190.0 cm	310.0 cm	-10.0 cm
Rotation	-0.000061 °	0.000075 °	-10.000091 °
Absolute Scale	0.1	0.1	0.1

Abst Museum Object

Object to Display: Aim_Space_Hip

Scale: 4.0

Figure 11 : Curator layout

The curator layout removes the “Modes” Panel from the Unreal Editor. The curator will only be adding new museum objects, and will not need access to any of the editor modes. The image shows the Level Editor, the Content Browser, the Details Panel, and the World Outliner. The details panel only opens when an object is selected in the level editor. The curator can select a piece of art in the museum and adjust its location, rotation, and scale using the details panel. Additionally, the curator can set a new “Object to Display” using the details panel. The world outliner usually displays all of the objects in a level. Since the curator is really only interested in the artwork on display, they can sort this panel by museum pieces. Using the world outliner, the curator can see the names of each piece currently on display in the wing.

Once the curator is satisfied with any edits to the museum, he will upload a new version of the museum that can be downloaded by visitors.

4. Implementation

Using Unreal Engine 4, the team needed to create the museum space, a player controller, and an object to represent each piece in the museum. To ensure good performance, the full museum was broken up into several smaller chunks, called wings. The Unreal project includes all of the wings of the museum, the submitted pieces, and all other files required for the curation of the museum.

4.1 Museum Architecture and Lighting

4.1.1 Architecture

Maya was the primary 3D modeling software used for the museum. The museum required architecture and Maya contains the tools necessary for modelling large, potentially complex spaces. Models generated in Maya can be easily imported to Unreal Engine. The team's artist created the original four museum wings, including railings and room trim, as well as architecture modules that could be used to construct additional museum space in the future. Once the museum size was decided, paper mockups of the rooms were created. The mockups were detailed with information about each wings scale, layout, and required art assets. These details were incredibly important when creating the assets in Maya and for importing them to Unreal.

The museum’s architecture originally built at one-tenth scale in Maya. This reduced scale made working in Maya more manageable. When building the rooms, the walls, floor and ceiling were created as separate objects, simplifying UV mapping. Each individual piece was able to have their own UV map which increases the resolution. Because the objects would eventually be imported to Unreal and reassembled, it was important to adhere to a strict naming convention. Once modelled, objects were UV mapped (flattened out in order to make texturing more reasonable and comprehensive) and given the proper names, then they were migrated to Unreal.

The pieces of each wing were scaled appropriately, and assembled into the complete wing. After the geometry of each wing was created and assembled, materials were applied to the models in order to finalize the appearance of the museum.

Materials are Unreal assets that can be applies to models to modify their appearance similar to how wallpaper modifies walls. All of the architecture in the museum has a material of some sort that gives it a unique and realistic look to it. Materials are created from image files that are associated with a specific object in the 3D space.

Every material created had five main parameters. The first being the base color. Base color affected the picture on the material. The floor texture looks like red concrete because of Base Color. The next two parameters are metallic and roughness. These two values work together. Metallic affects how much the material looks like metal, 0 being not like metal and 1 being like metal. Roughness affects how rough the material is. The roughness value affects the amount of light that reflects off of the surface of the material. Specularity, the next parameter affects how light reflects off of the material. Specularity gives the material a pattern to reflect light and breaks up the texture a little bit. And finally there is a normal map. A normal map makes it look like the textures have depth. Without a normal map, a stone texture will look very flat but with one the texture looks much more believable due to the texture having slight bumps and indentations similar to stone. All of these parameters are used to make the materials in the museum the most realistic.

To complement the materials and make them better, a program called Substance designer was used. Substance designer is a node based texture program similar to Unreal’s material system. However, Substance designer’s main purpose is to create materials and can produce higher quality materials than Unreal’s material system. Substance designer has more to offer in the ways of customizing the materials created. Once complete, the materials are exported from Substance designer into Unreal and then used to texture the museum.

4.1.2 Lighting

Many different aspects of the lighting were changed throughout the development of the museum such as, intensity, light color, cone angle, source length and at one point the museum space was shrunk to improve the lighting quality due to certain lighting parameter's max value. Three light types were used to imitate real lighting. Spot lights emit light pointing in a specific direction, point lights light up a light source to emit light in an area, and lastly a skylight is used to create the ambient lighting of a space.

4.2 Modularity

In order to support ten years' worth of student's artwork, modular rooms were created, allowing the curator to create more wings as needed. The modular rooms were designed with the same ideas in mind as the prebuilt wings, we wanted open space that compliments the artwork as opposed to take away from it. Each of the modular rooms can easily be connected to the other rooms. With these rooms, the curator can create many different wings with ease. All of the pieces required to assemble the modular rooms are in the same folder in the unreal project. The curator will need to select all the pieces of said room and drag them together in the unreal scene and all the parts will oriented in the correct position. Once this is complete, the selected pieces need can be grouped in order to move them as a whole easily.

4.3 Visitor Mode

In order to facilitate access to the museum itself, the team hosted the download link online using one of the IMGD department's servers. By going to the museum's URL, the visitor would see a homepage with a download button. Clicking on that button downloads the museum which can be run locally in order to view the exhibits.

The implementation of visitor mode began with the design of the museum object. Each piece of art in the museum is an instance of a museum object. The museum object is designed to easily give the curator the tools he or she needs to add, remove, and edit artwork within the museum. The museum object holds a special field containing the object to display. The object to display is an Object - Unreal Engine's base game class. All game objects in Unreal derive from the base Object class so this field can hold a variety of things. For example, Curators can display an image in the museum by assigning a texture to this field. The selected texture is then applied to one side of a cube, which is scaled to the image's resolution. In the case that a different Object is selected, such as a static mesh, then a different operation is run, resulting in the new piece of art being displayed.

Since the museum object supports 2D artwork, 3D animations, and audio, it was necessary to incorporate a series of functions that would apply to the object. The different types of piece will be interacted with in similar ways but required specific, subtle changes. For example, each museum piece displays textual information when the player approaches it. The behavior of displaying the text is controlled by a trigger volume attached to the piece. The size and position of this volume is related to the size of the museum piece and its type. 2D pieces have their trigger set in front of them and scaled according to the images width. 3D pieces have a trigger box with the piece in the center of the box.

As mentioned, the trigger boxes cause text associated with the piece to appear when the player is nearby. A function offsets the text from the piece according to its type, allowing for all labels to be displayed correctly. 2D pieces have the label offset at a constant value. For 3D pieces, the label appears perpendicular to a line drawn between the player character and the object so that it can be read from all sides.

Clicking the mouse while looking at an object selects it and outlines the object with a solid colored line. This behavior was easy to achieve in Unreal thanks to an object attribute called the object depth buffer. The object's custom depth can be used to generate an outline using simple edge detection. Once this is done, a post processing effect applies the outline material to the edge of the mesh.

When an object is selected, the visitor can interact with the object in several ways. Firstly, audio, videos, and animations can be paused and played. Secondly, the user can cycle through objects with multiple associated files. A piece might have multiple files if it is a static mesh with multiple animations or poses, or if the artist submitted multiple versions of the file to show off improvement over multiple iterations. All of these interactions are handled by a collection of functions within the museum object.

Finally, a selected object can be isolated. Giving the user the ability to zoom in, rotate, and move the object in order to get a better look at the objects finer details. From a technical standpoint, isolation view stores the museum object's associated asset file and recreates the object in a new level. A new player controller class is used in isolation view, which allows for the view and object manipulation.

4.4 The Submission System

The form was developed using HTML5 and Javascript. HTML5 was used to program the general layout of the form, while Javascript added functionality to it. Once artists submitted their work using the form, it was necessary to store these assets somewhere until the curator got a chance to look at them. Since the project is using Github to store all of the museum's code and contents, using the same tool to store submissions worked best for the curator.

Using Github still presented some challenges with regards to the storage of the museum. Additionally, many of the files stored in the repository would be binary files instead of code files. Until recently, storing binaries files using Github made it difficult to merge branches with changes to the same binary file.

Minor changes to a binary object would result in many changes to the file's text and these changes could not really be understood by a human reader. Git recently, however, released an extension to its service that allowed for the versioning of large files. This extension, called Git Large File Storage (Git LFS), replaces large files, such as videos, with text pointers. The file itself is stored on a remote server and can be requested by contributors to the repository and downloaded in full.

The repository contains the entire Unreal project directory for the museum. This directory stores all the contents of the museum, the source code, and the necessary Unreal plugins. The files that are available in the Unreal Editor are all stored in a directory called "Contents" within the project directory. An additional folder, called "Inbox" was created in the project's directory. All submissions are pushed to the repository and stored inside the Inbox folder. Subfolders are dynamically created based on the first author's username. Within a given username folder, each separate submission is stored in another folder, named using the date and time of submission. Arranging folders by username and timestamp allows for multiple submissions from the same author and resubmissions to be easily viewed and understood by the curator.

A JSON file is created and stored alongside the files submitted by the artist. The JSON file contains the information inputted to the form. This file is used by the museum objects to display the relevant information to the museum visitor and link the artist's portfolios if this information is provided.

This setup is ideal for the curator, because he or she simply needs to update his or her local repository periodically, analyze the new submissions, and decide whether or not to add the submission to the museum. Every so often, the curator will update the local version of the Inbox folder and check for new submissions.

4.5 Curator Mode

The museum curator performs his or her duty by having access to the Github repository containing the museum Unreal project as well as all submitted artwork. The curator will need to periodically update his local repository to check for any new submissions. If there are new submissions, the curator will be able to open them in UE4 and check their quality and make sure they meet all of the submission standards (which are listed in the submission instructions that are available to students and located in the Appendix of this paper). Submissions are first moved from the inbox folder into the museum's content folder, which gives the curator access to the file within the Unreal Editor.

The curator lays out the pieces in the museum using the Unreal Editor. Using the editor, the curator adds new museum objects to the museum and sets their "Object to Display" field to the desired piece of art. The curator can then adjust the scale, rotation, and position of the newly placed piece. When done selecting pieces and placing them within the museum, the curator packages the game for windows and deploys the executable to the website. The most recent version of the museum will always be available online.

To simplify the curator's job, a special editor mode was constructed for this project. The curation editor mode simplifies the interface for the curator. This curator interface gives the curator the ability to add new museum objects, and edit their scale, location, and rotation.

The curator can play the game within the Unreal Editor to view the collection as it would appear in the final version of the museum. Any noticeable problems can be easily fixed before the museum is packaged. Packaging the game only requires the curator to click the package button and wait for the executable to build.

5. Post Mortem

5.1 Successes

This project resulted in a fully functional interactive museum with four completed wings. The museum is ready to be populated by student work. Each of the museum's wings are fully textured and the materials used to create the wings are available if the museum size needs to be increased in the future. The lighting of the museum is worth mentioning because it provides a realistic feeling for the environment and interacts with submitted pieces nicely.

A special 3D model was created as a visual representation for audio assets displayed in the museum. The user interface is clean and readable. The user interface includes the menu bar displayed after piece selection as well as the museum object's text labels. A promotional trailer was created in order to demonstrate the museum to artists to generate interest in submitting artwork for the museum.

The features as they have been described are fully implemented for the museum. An executable version of the museum can be downloaded from the project's website, however the museum is currently empty until more students submit work. The submission form can be accessed on the project's website and has been used already for several submissions. When enough artwork is available, the curator will be able to layout the pieces within the wing and deploy an updated version. Laying out the pieces within the museum should not be a challenge to the curator because of the specialized curation interface. Additionally, the museum and all submitted assets are stored in a central Github repository. The repository can be used as a backup for the entire project in case anything disrupts the museum's functionality.

This project has a large artistic footprint that was created by a single person. The museum fulfills its goal from an artistic point of view. The museum is attractive but the architecture does not detract from the work on display within the museum. If additional wings are created in the future, it is easy to apply the materials to the new wings to give each wing the same look and feel.

This project created a dependable system for receiving and maintaining student pieces within a virtual space. When visiting the museum, users can utilize the features implemented in order to interact with and analyze pieces. Work within the museum can be linked to a student's personal online portfolio, which should aid students in the pursuit of careers. Finally, getting work displayed in the museum is a goal that will encourage students to do their best work.

5.2 Failures

While the team considers this project a success, there are things that could have helped during development or improved the final product. Originally, the team hoped to host the museum online, so that it could be accessed and explored through a web-browser. Unfortunately, Unreal Engine's web support is in beta and does not support the full suite of features utilized by this project. Additionally, creating the look and feel of the museum wings using materials and lighting took up a large amount of the artist's time. With more research earlier in the development, the artist might have been able to focus more time elsewhere, instead of on the challenges he faced.

The submission of static meshes and animated models is more complicated than the team originally wished. Moving content between two Unreal projects is difficult and special care must be taken to ensure that the assets transfer correctly. Further investigation into the Unreal Art pipeline may reveal a better way of handling the submission of more complicated artwork.

5.3 Future Directions

The immediate next step for the museum is to receive enough art submissions from students in order to have a good sized collection for visitors to explore. The team request submissions from artists through emails, and by attending events in person. However, very few submissions were received in the weeks following these requests. Now that this project as reached its conclusion, it will be necessary for IMGD faculty to encourage students to submit their artwork to the museum. Some faculty may go as far as making submission a requirement for their assignments.

For the future management of the museum, the main concern artistically is the expansion of the museum in order to accommodate more pieces as the repository of artwork increases in size. This means that the curator, with potential help from IMGD students, will create new wings for the museum. In doing so, the curator will need to follow the artistic precedent of the museum, using the materials already created by the team.

With regards to technical aspects of the museum, there are additional features that could eventually be added to further the experiences of museum visitors. For example, a search system was discussed during the design of the project but never implemented. A search feature would allow visitors to view all work done by a specific artist. Searching would bring the visitor into isolation view and give them the ability to toggle through all of the artist's work.

The team worked to create the museum for an HTML5 build and have it hosted in the web browser. In Unreal Engine's HTML5 pipeline, many of the main features of the museum are not supported, which led to the team switching to an executable/installer approach. If possible, future work could be done on this project to get full web support working. When the engine is capable of building an HTML5 project without loss of functionality, the curator should quickly be able to make the museum playable in a browser. It should not be hard to make the museum browser playable, since a server is already in place to host the project's website.

Bibliography

About IMGD. (n.d.). Retrieved March 12, 2016, from

https://www.wpi.edu/academics/imgd/about.html

https://www.wpi.edu/Pubs/E-project/Available/E-project-030915-213100/

André Desvallées and François Mairesse (Eds.). *Key Concepts of Museology*. 2010.

Appendix A: About IMGD

Artistic Vision Meets Technological Know-How

Students who choose to study Interactive Media & Game Development (IMGD) at WPI learn in an environment that fosters a deep appreciation of both the artistic and technical aspects of IMGD--while focusing on developing in-depth skills in either the artistic or technical domain.

Our faculty's goal is to produce well-rounded grounded graduates by helping artists learn enough programming to be able to appreciate the challenges programmers face, and to help programmers become versed in the language of art so they can collaborate with their artistic colleagues.

LEARN BY DOING

Computer games may well be the most easily recognizable form of interactive media, but the IMGD program also offers digital whiteboard presentations, distributed virtual environments, and performance-based media. Game development involves all aspects of creating games--design, content creation, programming, testing, and project management.

Our program will challenge you to be creative, explore, collaborate, and push beyond boundaries to create entirely forms of new media that haven't even been imagined yet.

BUILD A PORTFOLIO

A key component of the IMGD program at WPI includes building a portfolio of unique works that will help give you a competitive edge as you pursue the career of your dreams.

Five things you should know about WPI's IMGD program

The IMGD program has 22 faculty members, including three, full-time Professors of Practice. Between them they have over 75 years experience in the industry. They are here every day, advising students, teaching courses and supervising projects.

About 90% of IMGD 2015 graduates are already gainfully employed at places like Blizzard, Riot, Microsoft, and Harmonix. The average salary for a game developer is $60,000 a year.

IMGD students don't just study things (though there's plenty of that too) - they MAKE things every day in groups that include both technical people and artists. It's super hands-on, creative, and collaborative.

59% of Americans play games. 48% of game players are women and most games (56%) play online with other people. The average game player is 31 years old. Games aren't just for kids.

Games are being made and used for education, health care, training and many other applications in addition to entertainment. IMGD students can partner with scientists and engineers to create learning experiences that are only possible in games.

Appendix B: Creator/Artist Submission Instructions

Creator Instructions

3D models/animation submissions:

1. Export piece from native program to a .fbx file
2. Import piece into the Content directory of a blank unreal project with these steps
   a. **Note: all files should be in the root content folder, do not create any subfolders**
3. (Optional) Place the unreal asset into the scene to ensure that it looks correct.
4. Right click asset in the Content browser -> Asset Action -> Migrate
5. Select Destination -> Place in easily findable file location
6. You will get a warning that the destination for the asset is not a content folder. Click "Yes" to place the asset anyway.
7. The resulting folder will contain all the files ready for submission.
8. Go to the submission form and add each file from the migrated folder.
9. Fill out the submission information form.
10. Submit.

2D Images:

1. Make sure the file(s) you are submitting is one of the following accepted file types:
   a. .jpg
   b. .png

2. Go to the Asset Submission Website and choose the Image option from the drop-down list and proceed to the next section.
3. Fill out all of the information on the form and select the file(s) that you are submitting.

Videos:

1. Make sure the file(s) you are submitting is a .mov file
2. Go to the Asset Submission Website and choose the Video option from the drop-down list and proceed to the next section.
3. Fill out all of the information on the form and select the file(s) you are submitting.

Audio:

1. Make sure the file(s) you are submitting is a .wav file.
2. Go to the Asset Submission Website and choose the Audio option from the drop-down list and proceed to the next section.
3. Fill out all of the information on the form and select the file(s) you are submitting.

Appendix C: Curator Instructions

Required Software

Unreal Engine 4 (v4.10.4)
Visual Studio 2015
Git Bash for windows.

1. Pull the Github repository

“git pull origin master”

2. Review Inbox

Look at newly added files

Video, audio, and images can be viewed using curator’s choice of program

Adding these submissions to the project

1. Check the “Content/MuseumPieces” directory to see if the student has submitted before.
   a. If not, create a folder with the student’s username
2. Copy the timestamped submission folder into “Content/MuseumPieces/<username>”

To view animations/static meshes, copy all files within the timestamped submission folder into the Museum “Content” directory. These files must be in “Content” not a subfolder. Remember these files since you will need to move them within the Unreal Editor in a moment.

Adding these submissions to the project

1. Open the Unreal Project “MuseumMQP.uproject”
2. Using the content browser open “Content/MuseumPieces” and check to see if the student has submitted before
   a. If not, right click and create a folder with the student’s username
3. In the Unreal Editor select the Content folder in the content browser
4. Select all of the files that were just added to the project.
5. Drag to move these files to the username folder created.
6. Select Filters->Miscellaneous->Redirector
   a. If any files appear, right click them and select “fix up”
   b. Toggle redirector off

Now the files are in the Unreal Project and organized properly

3. Make Changes to the museum

1. Open the Unreal Project “MuseumMQP.uproject”
2. The museum wings can be found under “Content/Wings”
3. Open any of the wings to begin editing
   a. Typing “Museum” in the search box of the world outliner will display all of the museum objects currently being displayed in the museum
   b. Selecting an object and pressing “f” will focus the camera on that object
4. The names of these objects can be changed in the world outliner in order to make the curator’s job easier
5. Under the Content directory is an actor called “Museum Object”
6. Add these objects to the scene in order to add more artwork.

7. Select a museum object in the scene to edit it.
    a. In the details panel, select the “Object to Display” field and type to search for new artwork to display.
    b. Adjust the scale of the object in the museum by inputting changing the “Museum Scale” value of the object.
        i. **Note: Do not adjust the “Absolute Scale” of the object in the details panel - this will scale the entire actor and could cause the text label to appear incorrectly.**
8. You can adjust the object rotation or location using the scene editor or the details panel.
    a. **Note: do not adjust the scale of the object in the scene editor, this will scale the entire actor and could cause the text label to appear incorrectly**
9. When satisfied with an object’s appearance in the museum, play in editor can be used to ensure everything looks correct.

4. Add, Commit, and Push local changes

“Git add -A”
“Git commit -m “Museum updated <date>”
“Git push origin master”

5. Build the Museum

1. Select “File->Package Project->Windows ->Windows 64 bit
2. Select a destination for the build folder
3. When the build is finished find the destination folder
    a. Rename the folder to “IMGD_Museum”
    b. Open the folder and ensure that the .exe runs and all changes are present.
4. Zip up the folder
5. Replace the Current zipped folder on Alpheus with the new folder.