



CS CAPSTONE DESIGN DOCUMENT

APRIL 12, 2019

APOLLO 11 3D ANIMATION

PREPARED FOR

OMSI

JIM TODD

Signature

Date

PREPARED BY

GROUP 49

THE APOLLOERS

JONATHAN ROPP

Signature

Date

SHANNON SANDY

Signature

Date

DEAN AKIN

Signature

Date

Abstract

The Summer of 2019 will be the 50th anniversary of the Apollo 11 moon landing and our group, 'The Apolloers', is working to create a 3D animation of the mission. This animation is being made for Jim Todd at OMSI so the animation can be displayed as part of their 50th anniversary exhibit. This document will show how we have organized the project and offer different design viewpoints so that stakeholders can see how the project has been structured.

CONTENTS

1	Overview	3
1.1	Scope	3
1.2	Purpose	3
1.3	Intended Audience	3
2	Definitions	3
3	Design Description	4
3.1	Design Stakeholders	4
3.2	Design Views	4
3.3	Design Viewpoints	4
3.4	Design Elements	4
3.5	Design Overlays	4
3.5.1	Textures and 3D Models	4
3.5.2	The Flight Path	5
3.5.3	Planetarium SDK	5
3.6	Design Rationale	5
3.7	Design Languages	5
4	Design Viewpoints	6
4.1	Context Viewpoint	6
4.2	Composition Viewpoint	6
4.3	Logical Viewpoint	6
4.4	Dependency Viewpoint	6
4.5	Information Viewpoint	6
4.6	Interface Viewpoint	7
4.7	Resources Viewpoint	7
5	Conclusion	7

Revisions

Section	Original	New
3.5.2 - The Flight Path	<ul style="list-style-type: none"> We will use a data set to create the flight path 	<ul style="list-style-type: none"> A data set would be ideal, but we will need to create our own
3.5.3 - Planetarium SDK	<ul style="list-style-type: none"> Unsure of how to show our animation in the planetarium 	<ul style="list-style-type: none"> OMSI uses a unique scripting language for implementation in the planetarium
4.2 - Composition	<ul style="list-style-type: none"> Go through mission from launch to splashdown 	<ul style="list-style-type: none"> Beginning and end with historic video, middle for interactive views
4.4 - Dependency	<ul style="list-style-type: none"> Mostly dependent on flight path and mission time 	<ul style="list-style-type: none"> Dependent on the story-line that we created (Beginning, middle, end)
4.5 - Information	<ul style="list-style-type: none"> Focused on flight path information 	<ul style="list-style-type: none"> How to make the scene realistic through size and positioning

1 OVERVIEW

1.1 Scope

This document focuses on the 3D animation for the Apollo 11 Mission that will be created by our group, "The Apolloers" for display at OMSI. The design principals of the project will be outlined here, as well as design elements, such as functionality, usability, and requirements. These principals and elements will outline the deliverables of our project as well as the resources needed to complete it. The context, stakeholders, and intended audience our project will be used in will also be outlined, which will provide supplementary reasoning for our chosen deliverables.

1.2 Purpose

The design and structure of this project will be outlined so that readers would be able to understand how the 3D animation is organized from a development viewpoint. In addition to the readers, this document will also serve as guidelines for our group to follow throughout the year. This document will be changed throughout the year as we refine our design.

1.3 Intended Audience

This document is intended for the stakeholders of the Apollo 11 3D animation, as well as any interested group that would like to know more about how the animation is structured. Some main takeaways of this document are the design, structure, and connectivity of the components of our animation, so that readers would have a general idea of how to make a similarly structured project.

2 DEFINITIONS

Term	definition
API	An Application Programming Interface is a set of protocols and tools that are used to build a software application. Essentially the 'building blocks' that a programmer uses to build an application.
Apollo 11 Mission	A spaceflight operated by NASA to land the first humans on the Moon, launched July 16th, 1969.
Graphics Pipeline	A conceptual model that describes the steps that a graphics system needs to perform in order to render a 3D scene.
High/Low Poly	The amount of polygons used to make a 3D model, impacting performance.
NASA	The National Aeronautics and Space Administration is a federal agency that focuses on research and development related to air and space.
OMSI	The Oregon Museum of Science and Industry, located in Portland, Oregon
OpenGL	An open-source graphics library API that is used to interact with graphics hardware to design 3D renderings.
SDK	A Software Development Kit is a set of tools that program developers use to write programs for an application.
Polygon Count	Refers to the number of polygons in a scene. The more polygons in the scene, the more time it takes for the scene to be executed, often resulting in lag.
SkySkan	A company that provides planetarium software and equipment to OMSI.

3 DESIGN DESCRIPTION

3.1 Design Stakeholders

The stakeholders for our project are OMSI, as well as the audiences who will be viewing our video. The educational and technical quality of our video will reflect on OMSI. The planetarium audiences will also be affected by the technical and educational quality of our video. If the video is more entertaining than educational, then the audience will not understand the impact and ambition of the Apollo 11 mission. If the video is more educational than entertaining, then the audience will be bored and less likely to pay attention to the content of the video.

3.2 Design Views

The Apollo 11 animation project can be broken into different views that can all be used to describe the animation. This document will look at the design of the animation through the different viewpoints listed below, and will explain those viewpoints in detail in Section 4.

3.3 Design Viewpoints

- 1) **Context:** In what context will this animation be viewed.
- 2) **Composition:** How the animation is separated into different entities.
- 3) **Logical:** What logic is constant throughout the animation, regardless of design decisions.
- 4) **Dependency:** How different entities of the animation depend on others.
- 5) **Information:** What data will be needed and how it will be used in the animation.
- 6) **Interface:** How developers should correctly use the animation.
- 7) **Resources:** What external entities are needed for the animation.

3.4 Design Elements

There are many elements of a 3D animation that all need to work with the other elements to produce a quality project. The programming platform OpenGL will be used for managing the elements relating to 3D graphics. Since we are building for a Windows platform, we will also make use of Windows system calls as needed.

The programming in OpenGL will dictate what 3D objects are in the project, where they are in the scene, and how they look. Many 3D objects in our project will come from an online repository, which we may need to add textures to, which would also be found online. Since the source for each object will likely be different, proper scaling will need to be applied to keep the size realistic. Once a 3D object is in the scene, different lighting features will be programmed in to account for the Sun and that light reflecting off of the other objects. Then to make objects move, key-frame animation will be used to interpolate the path between a start and goal point.

3.5 Design Overlays

3.5.1 Textures and 3D Models

Textures will be needed for some of our 3D objects, such as the Earth and Moon. Most of these textures will come from the NASA websites, which are readily available for public use. It is notable that these textures have been created with great detail, meaning that the textures may slow the program, or look out of place if near a lower quality texture.

The 3D model of objects such as the Lunar Module will likely be obtained from a model sharing site. We will attempt to find low-poly objects for our initial program so that the playback is not slowed down. High-poly objects may look better, but take many more resources to load. When our program is implemented at OMSI, Jim Todd should have industry quality models that can be used in the planetarium with less concern for polygon count.

3.5.2 *The Flight Path*

Ideally we would want a dataset of the whole flight and back. Using that data set, paths can be interpolated from between the sets of points, resulting in a realistic flight path. While this would be straight-forward to implement, it is unlikely that we will have access to such a resource. Instead, we will need to calculate our own flight path. This will still be done by using data points, but we must calculate each one, so there will be far less data points, meaning a less accurate flight path.

3.5.3 *Planetarium SDK*

One of our biggest stretch goals is to integrate our animation into the OMSI planetarium. We suspect that their planetarium use an SDK to interpolate subsections of the projection, but we currently do not have access to that SDK. Our contact here at OSU, Mike Bailey, has been in touch with SkyScan to determine how to best go about the process to implement an animation. It appears that there is a unique scripting language that is used for the planetarium. If the animation will go to the planetarium, we will use sample scripts to develop a prototype script resembling our animation for Jim Todd to test and finalize.

3.6 Design Rationale

The basic rationale of our project is that setting the scene correctly is the most important factor in making this animation high quality. The Earth, Moon, and Sun need to be in the right position and the view port needs to be in the right position to reflect that. The lighting from the sun needs to be accurate as it reflects and refracts off the lunar capsule, Moon, and Earth. The small details matter the most, such as the Earth having its iridescent glow that is given off by its atmosphere, or both planetoids having a dark and a light side. With all of this combined, our goal is to make the audience feel as if they are watching the Apollo 11 mission live.

3.7 Design Languages

C++ is going to be our design language solely because of our choice to use OpenGL as our API. OpenGL makes use of graphics libraries that are written in C++, meaning that we must also utilize that same language. C++ gives the programmer great control over what they implement, but because of that control, this allows the programmer to make a lot of mistakes. C++ is fast and efficient like its predecessor language, C, and has more back-end libraries that can help the programmer achieve what he is trying to do. Even though these extra libraries can slow down an application, the effect is minimal for a project with a scope as large as ours. Lastly, our group will need to be cautious with memory usage because C++ does not automatically clean up memory such as some other languages, so we will need to be sure to free any memory that we allocate.

4 DESIGN VIEWPOINTS

4.1 Context Viewpoint

This animation will be viewed primarily by OMSI visitors. This audience can range from young school children on field trips, to very technical industry leaders. We will want the user of this program to be able to change the viewpoint of the scene, regardless of if the user is a visitor or a staff member running the program.

4.2 Composition Viewpoint

Our project will consist of a beginning, middle, and end. The beginning and end will consist of historic video related to the Apollo 11 mission to educate and provide context. The middle will allow the operator to take control of what viewpoint to see and be able to look around that view with the mouse. 3D objects, animation, audio, textures, lighting, shaders and more will all come together to create the scene. While there are not direct relationships between all these parts, they all add to the overall detail and quality of the animation.

4.3 Logical Viewpoint

Our logic will be determined by how far into the animation the operator is. The beginning and end of the animation will be shown in a sequential fashion with little to no variation. When a certain action completes and/or a time is reached, new actions will take place. Conversely, during the middle of the animation, our program will allow changes in viewpoints. The simple logic will result from different keyboard buttons being pushed. These presses will not only change the view, but reset the previous and new scenes, this way the operator will always know what to expect when they press those keys. A stretch goal would be to allow some animations to be returned to the point in time when the last view was changed. This would be useful for a longer animation that the operator may want to pause and come back to later.

4.4 Dependency Viewpoint

Different parts of the animation will not need to be loaded for certain viewpoints. For example, the lunar surface with an astronaut does not need to be loaded if the view is from the Earth. Each object will be given a 'flag' of sorts that we can set depending on what each view needs loaded. This will help with performance and prevent objects appearing out of place. Along these same lines, certain audio clips will be played when a view is chosen or when a certain time is reached.

Other than when to load objects, our program will not contain many actions that are dependent on others. With a graphics project such as this, all assets are loaded into memory at the beginning of running the animation so that each asset can quickly be retrieved for display to the screen. The program mainly determines when and how to display data from the computer's memory.

4.5 Information Viewpoint

General knowledge of the Apollo 11 mission is needed to create an accurate representation in an animation. The sequence of events and a timeline are necessary, but also details such as the true size of all objects in the scene and the positioning of those objects. Because of our audience, extra care must be taken to make the animation as realistic as possible since there may be members of the audience that are quite knowledgeable about Apollo 11. When possible, all

information will be acquired from an official NASA source and if that is not possible, other reliable sources will be used, such as news outlets and scientific journals.

4.6 Interface Viewpoint

The 3D animation will be compiled into an executable file (.exe) so that the user can open the whole animation from one file. Then, the user will be able to start and stop the animation using a keyboard, and change their viewpoint by clicking and dragging their mouse, or choose set viewpoints using the keyboard. If the user does not want to change the viewpoint, the viewpoint will be set to default viewing positions that our group has chosen. The user will be able to view the video from outside of the spacecraft, inside the spacecraft, on the moon, and on Earth.

4.7 Resources Viewpoint

To obtain the audio of the transmissions between the Apollo 11 crew and Tranquility Base, we will use NASA's archive of audio files from the Apollo 11 mission. Using Windows system calls, these files will be played approximately at the same stage of the mission as they were recorded in. Our group will not be implementing every audio file in the Apollo 11 audio archive, but instead we will aim to enhance the animation. The audio clips will need to be educational, but not include too much jargon for the audience.

Similarly to audio, our group will also need to obtain 3D models and textures from online repositories. Models will include the Command Module Columbia, and Lunar Module Eagle, an astronaut, and more. Multiple textures will be needed for the Earth and the Moon at different levels of detail, depending on how close the view port is during the animation. Finally, the background will need an accurate star-map to emphasize the vastness of space. Although, the star map needs to toggle because the stars cannot be seen from the surface of the Moon.

5 CONCLUSION

A 3D animation consists of many different elements that all need to work together based on the implementation in a given API. We will be using OpenGL as our primary API that will manage the main graphical elements of our group's Apollo 11 3D animation. The animation will include historical context of the mission and what the mission was like on the Moon with full functioning elements such as 3D objects, texturing on those objects, lighting for the scene, etc. With a completed animation, direct users of the animation will be able to use a mouse and keyboard to navigate through the animation and view the scene through arbitrary viewpoints. Lastly, the main goal of this Apollo 11 animation is to engage audiences at OMSI and encourage curiosity in regards to the vastness of space.