Planetary Surface Flyover Movie Generator
(PSFMG)

CS 4961 Senior Design

Functional Requirements and Design Document

Prepared by

Shawn Anderson
Fidel Izquierdo Jr
Angel Jimenez
Khang Lam
Christopher Omlor
Hieu Phan

2 December 2016

CALIFORNIA STATE UNIVERSITY LOS ANGELES

Los Angeles, California

Planetary Surface Flyover Movie Generator
(PSFMG)

Prepared By:

Shawn Anderson
Fidel Izquierdo Jr
Angel Jimenez
Khang Lam
Christopher Omlor
Hieu Phan

Approved By:

___________________	_______________
Dr. Kang	**Date**
___________________	_______________
Richard Cross	**Date**
___________________	_______________
JPL Liaison	**Date**

Document Change Log

Changes		
Update	Date Released	Draft
10/14/16	N/A	First Draft
11/20/16	N/A	Second Draft
12/1/16	N/A	Fall 2016 Final Draft

List of TBD Items

Page	Item	Description	Status
Section 4	Detail Design	Level 2 DFDs	TBC Spring Semester 2017
Section 5	Implementation	Coding	TBC Spring Semester 2017
Section 6	Test Plan	Testing Plan	TBC Spring Semester 2017

Table of Contents

List of Figures

1.0 Introduction

1.1 Purpose

This document will provide a detailed outline of the software architecture for the Planetary Surface Flyover Movie Generator(PSFMG). This document will provide readers the details necessary to understand the system and its functionality.

1.2 Scope

This document intends to show how the design will accomplish the functional and nonfunctional requirements described in the Software Requirement Specification(SRS). This document will provide the details required to implement the software system by describing the architecture, subsystems, and interfaces.

1.2.2 Relationship to Other Documents

N/A

1.3 Context Diagram (DFD Level 0)

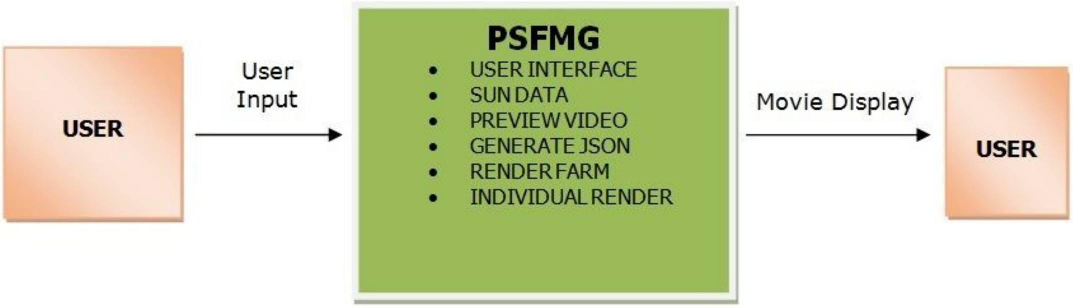

Figure 1-1: Level 0 DFD

1.3.2 Description and Major Functions of PSFMG

PSFMG will provide a user friendly interface for creating,deleting, and editing points on the surface of a planet and creating a path to create a movie.

1.3.3 Hardware and Software Considerations

1.3.3.1 User Hardware Requirements

N/A

1.3.3.2 System Hardware Requirements

Webhost requires a basic webserver with public accessible address and ports. Hadoop cluster for backend processing of data for movie creation.

1.3.3.3 User Software Requirements

HTML5 compliant web-browser that fully supports WebGL and a graphics card that supports 3D.

1.3.3.4 System Software Requirements

The software system has several different requirements for the various components of the backend processing.

- Cesium current version for hosting UI
- Nodejs for running webserver to host the Cesium UI
- Blender version 2.78 for creating and rendering the movie
- Python 3.5 for scripting automation of blender and running server for backend
- Spice Framework for calculating the position of the sun relative to position on planet
- Spiceypy Python wrapper for Spice Framework
- Hadoop for distributed processing of rendering tasks and animation

1.4 Documentation of the Development Process

PSFMG's detailed functional description is documented in section 2.0. Section 2 is a succinct software description document. The overall detailed functional description is based on higher level DFDs (above level 1). All major functional units are described in detail in this part of the document.

Requirements for PSFMG are captured in Section 3.0 of this document. This section includes both functional and non-functional software requirements, supplemented with more detailed information when necessary.

Section 4, PSFMG's detailed Design Description Document (SDD), will be completed in Spring Semester 2017.

Section 5, PSFMG's Software Implementation Document (SID), will be completed in Spring Semester 2017.

Section 6, PSFMG's Software Test Plan (STP), will be completed in a subsequent Spring Semester 2017.

1.5 References

N/A

2.0 DETAILED FUNCTIONAL DESCRIPTION OF PSFMG

2.1 Detailed PSFMG Functional Description.

The major tool used to design PSFMG is the Data Flow Diagram, DFD. The rationale for the selection of DFDs as the preferred design tool is its simplicity and versatility. In the future additional tools may be used if a stronger correlation from Design to Requirement to Implementation and Testing is required.

2.1.1 Level 1 DFD

PSFMG's major functional subunits are shown in the DFD Level 1 shown below:

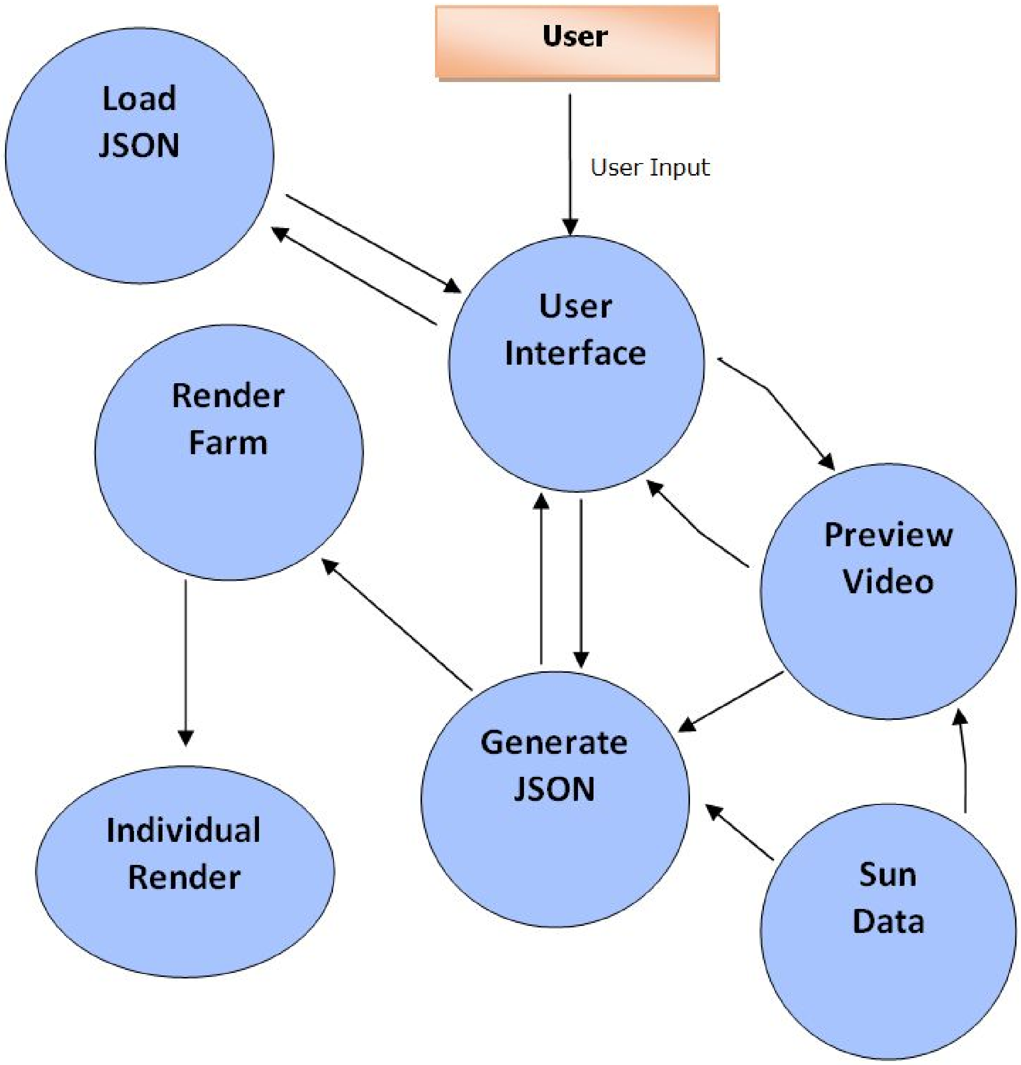

Figure 2.1: Level 1 DFD

2.1.2 Detailed Functional Description of PSFMG’s Major Units

User Interface - Module 2.1

Using UIM, user can choose to load in a JSON file or start choosing a patch from scratch. User can select camera angle offset, position, start time and time offset for each point. When user is finalized about points they select ,movie format, quality, and email to receive download link and JSON file.

Sun Data - Module 2.2

SDM will use NASA SPICE API to take a range of points and time in space to output a position, angle, distance and intensity of the sun's rays in the scene. SDM will be able to output this data by client request

Preview Video - Module 2.3

PVM will request associated sun points for each user specified points in the video. PVM will run a client sided render that will play a low resolution preview of the movie with scientifically accurate lighting that will be rendered in high resolution from the backend process. PVM will return to user interface to allow user to change points should they be dissatisfied with their current path

Generate JSON Start Backend - Module 2.4

GJSBM receives JSON file from UIM. Parses the file received into dictionaries and tuples, has the data ready to send back to other modules.GJSBM then sends parsed data.

Server Task Management - Module 2.5

STMM begins importing OBJ file while user creates movie path, parses JSON file and set configuration options and read point data. STMM will create scene from point data and map each frame to a time offset and render still image of camera view as a PNG file. STMM will reduce range of frames into MP4 animation and combine all movies from reducer using MP4box into single MP4 Animation. STMM will send absolute file path of final animation back to web-server

Individual Server Sided Rendering - Module 2.6

ISSRM will read JSON file and create a camera path, camera angle, object and sun path in the scene. ISSRM will render still images along the path in the scene and save into a folder. ISSRM will take the images and output a video of the specified filetype. ISSRM will send the smaller video to the main server. Final render server will do these tasks and receive the other mini videos and combine them together. ISSRM will transfer movie to FTP server for download and generate URL.

3.0 PSFMG Requirements

3.1 PSFMG Functional Requirements

Requirements Related to Design Module 2.1: User Interface Module	
Requirement No.	**Requirement Description**
3.1.1	UIM shall allow user to load in a JSON file that contains a set of points
3.1.2	UIM shall display the points contained in the JSON file
3.1.3	UIM shall allow user to modify the points loaded in from JSON file
3.1.4	UIM shall allow user to create a point on the map
3.1.5	UIM display a UI with textboxes for camera pitch, roll and yaw and camera longitude, latitude, altitude as well as time offset from starting point
3.1.6	UIM shall display a sprite on the map for each point with the associated data visible on the indicator
3.1.7	UIM shall allow user to delete an existing point
3.1.8	UIM shall allow user to modify an existing point
3.1.9	UIM shall allow the user to click a button to confirm that the user will render along these points
3.1.10	UIM shall generate a JSON file to be sent to user and stream the data to the render server

Requirements Related to Design Module 2.2: Sun Data Module	
Requirement No.	**Requirement Description**
3.2.1	SDM shall receive a range of points and time in space.

3.2.2	SDM shall use Spiceypy to determine sun position and angle.
3.2.3	SDM shall send output to module requesting it.

Requirements Related to Design Module 2.3: Preview Video Module	
Requirement No.	**Requirement Description**
3.3.1	PVM shall play the video inside of the browser
3.3.2	PVM shall allow the user to go back to UIM to modify their points
3.3.3	PVM shall allow the user to click a button to confirm that the user is satisfied with the points and willing to render the movie

Requirements Related to Design Module 2.4 Generate JSON Start Backend Module	
Requirement No.	**Requirement Description**
3.4.1	GJSBM shall receive JSON file.
3.4.2	GJSBM shall parse the JSON file.
3.4.3	GJSBM shall send the parsed data other modules.

Requirements Related to Design Module 2.5: Server Task Management Module	
Requirement No.	**Requirement Description**
3.5.1	STMM shall receive list of file names saved on NAS
3.5.2	STMM shall read list of file names from NAS disk.
3.5.3	STMM shall get the total number of frame

3.5.4	STMM shall determine the number of tasks required.
3.5.5	STMM shall update jobconf with required tasks
3.5.6	STMM shall split file names based on number of tasks required.
3.5.7	STMM shall receive the jobconf from 3.5.5.
3.5.8	STMM shall set jobconf attributes using configure function
3.5.9	STMM shall receive the lists of split filenames from 3.5.6.
3.5.10	STMM shall map a key to each list of filenames.
3.5.11	STMM shall receive the key, value pair list of filenames
3.5.12	STMM shall create threads and send a portion of list to each thread
3.5.13	STMM shall receive rendered images from threads
3.5.14	STMM shall render images using hard coded render optimizations with blender API
3.5.15	STMM shall save rendered images to temp directory.
3.5.16	STMM shall use blender API to animate images into mini movie
3.5.17	STMM shall store mini movie on NAS
3.5.18	STMM shall return mini movie location to master node
3.5.19	STMM shall receive mini movie file location from slave nodes.
3.5.19	STMM shall animate the final movie from the mini movies with mp4box
3.5.20	STMM shall save the final movie on NAS

Requirements Related to Design Module 2.6: Individual Server Sided Rendering Module	
Requirement No.	**Requirement Description**

3.6.1	ISSRM shall receive the associated user data as well as planetary data to create the scene
3.6.2	ISSRM shall create paths inside a .blend file that will take each point as points along the path
3.6.3	ISSRM shall create paths for the solar path, camera path as well as load the terrain data
3.6.4	ISSRM shall render a still image for each frame along the camera path
3.6.5	ISSRM shall have data such as terrain and lighting be relative to the camera path
3.6.6	ISSRM shall save each still image be saved to a folder on the computer
3.6.7	ISSRM shall look for a specified directory
3.6.8	ISSRM shall use the images in that directory to render the partial video using an API call inside blender

3.2 PSFMG Non-Functional Requirements

3.2.1 Performance Requirements

The software was designed to be able to run at an acceptable timeframe on any hardware however the better the graphics card and the processor on the machine running the software the quicker the render times will be. The software is also designed to run on parallel machines so having a server cluster will help render videos in acceptable timeframes.

3.2.2 Safety Requirements

There is no specific safety requirements since this software will not used in a capacity to control hardware.

3.2.3 Security Requirements

The safety requirements is to be handled in house by JPL. The software will be transferring data insecurely over unencrypted channels. Any and all additional security including protection of data and backups are outside the scope of the software being developed.

3.2.4 Software Quality Attributes

The software is designed to operate at a small scale so issues with scalability can arise with high volume. The render quality and times are designed to be at a high quality even when zoomed in as well as render frames at an acceptable time even on minimal hardware.

3.2.5 Business Rules

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

A. ACRONYMS

PSFMG	**Planetary surface flyover movie generator**
UIM	**User Interface Module**
SDM	**Sun Data Module**
PVM	**Preview Video Module**
GJSBM	**Generate JSON Start Backend Module**

STMM	**Server Task Management Module**
ISSRM	**Individual Server Sided Rendering Module**