2017-2018

Julio Ivan Davila Carasco & Kyriakos Lite

Universita La Sapienza

Interactive Graphics

A Polygonal Adventure

Contents

[Introduction 3](#_Toc490412833)

[Description of the Environment 3](#_Toc490412834)

[The Mainland 3](#_Toc490412835)

[The main character 4](#_Toc490412836)

[The Secondary characters 4](#_Toc490412837)

[Libraries, Tools and Models used in the project 5](#_Toc490412838)

[Tools 5](#_Toc490412839)

[Blender: 5](#_Toc490412840)

[Libraries 5](#_Toc490412841)

[Three.js: 5](#_Toc490412842)

[Technical aspects of the Project 6](#_Toc490412843)

[Collision Detection 6](#_Toc490412844)

[The translation and rotation mechanism 6](#_Toc490412845)

[The follow mechanism 6](#_Toc490412846)

[Background Environment 6](#_Toc490412847)

[Implemented interactions 7](#_Toc490412848)

[Bibliography 8](#_Toc490412849)

# Introduction

The “Polygonal Adventure” is a game made within the course of Interactive Graphics with purpose to realize the famous river crossing mathematical puzzle. Within the environment, the user is called to transport the Wolf, the Sheep and Cabbage across the river using the bridge but there are certain constraints that the user has to take into account. To begin with the user can transport only one of the aforementioned characters at a time. Also if Wolf is left alone with Sheep he eats it and if Sheep is left alone with Cabbage he eats it too.



Figure Polygonal Adventure Starting Interface

# Description of the Environment

## The Mainland

The game takes place at a floating island in space which was developed mesh by mesh for the purposes of this project using blender (LionStudiosTM, n.d.). As the name of the game indicates the environment is polygonal and can be seen below:

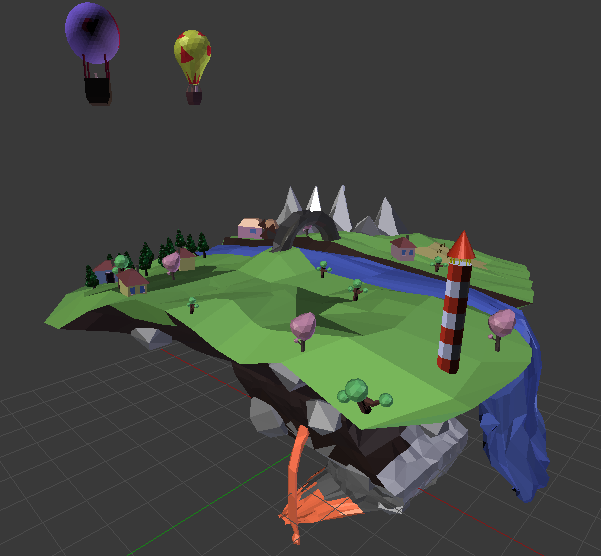


Figure Polygonal Island in Blender Environment

## The main character

The player is in control of the main character of the game shown below:



Figure Main Character

And gets to guide him around, and command the wolf, sheep and cabbage to follow and unfollow him with the keyboard keys as explained later in this document in order to solve the puzzle.

## The Secondary characters

The wolf, the sheep and lastly the cabbage are loaded on the one side of the island at the beginning of the game and the player has to guide them at the other side of the river with the correct sequence in order to solve the puzzle and win the game.

|  |  |  |
| --- | --- | --- |
| **Table of Secondary Characters** | | |
| **Wolf** | **Sheep** | **Cabbage** |
| Figure The Wolf in Blender Environment | Figure The Sheep in Blender Environment | Figure The Cabbage in Blender Environment |

# Libraries, Tools and Models used in the project

## Tools

### Blender:

Is a widely used free and open source 3D development suite. It supports the entirety of the 3D pipeline needed by a developer to develop a 3 dimensional model, rig, animate and many more functionalities.

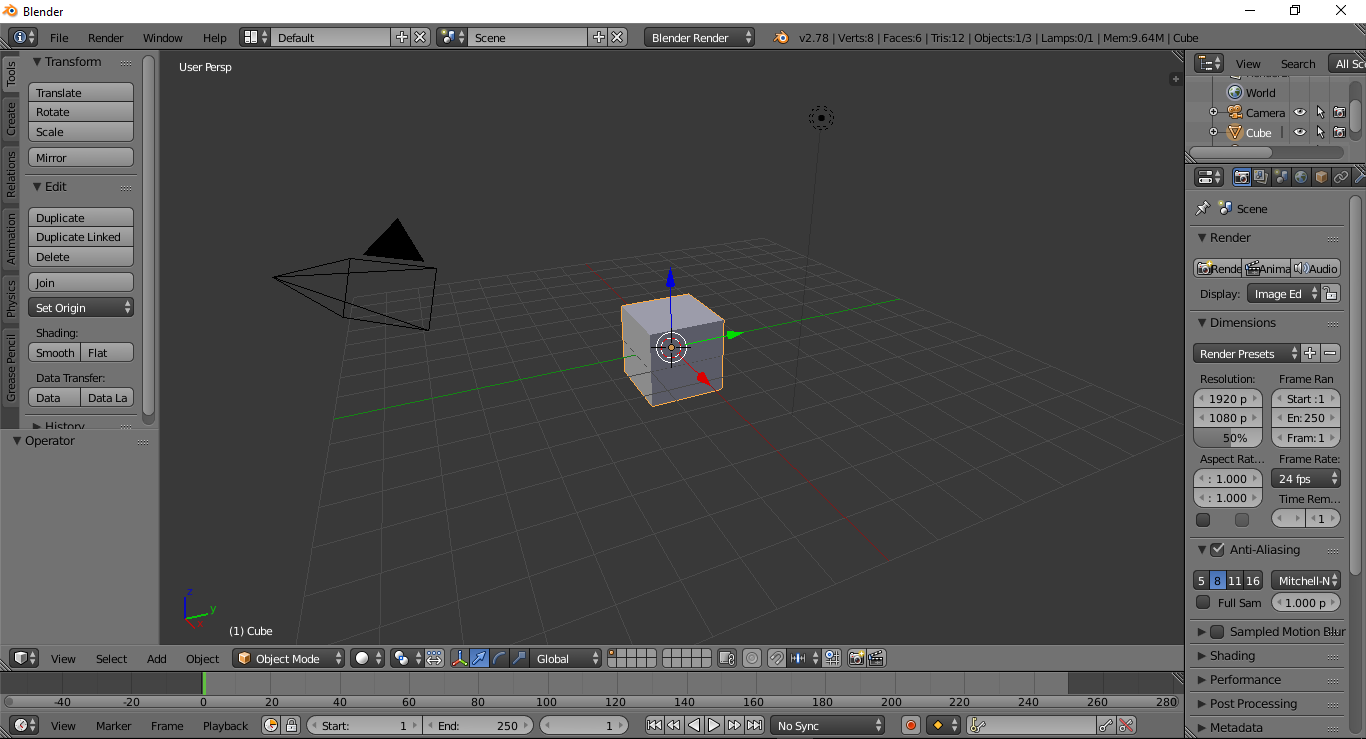


Figure Starting Interface of Blender

In our case, blender was used to create the mainland and the characters from scratch. In the same environment we did the rigging (imposing armature to the mesh in order to manipulate it) and create the walking animations.

## Libraries

Three.js:

Is a lightweight cross-browser JavaScript library/API used to create and display animated 3D computer graphics on a Web browser. Three.js scripts may be used in conjunction with the HTML5 canvas element, SVG or WebGL. Three.js allows the creation of GPU-accelerated 3D animations using the JavaScript language as part of a website without relying on proprietary browser plugins. This is possible thanks to the advent of WebGL. The library can render using Canvas, SVG and WebGL.

# Technical aspects of the Project

## Collision Detection

## The translation and rotation mechanism

In order for the main character to navigate through the terrain he is translated when W for forward and S for backwards are pressed but also rotated around his longitudinal axis. The relevant code is shown below.

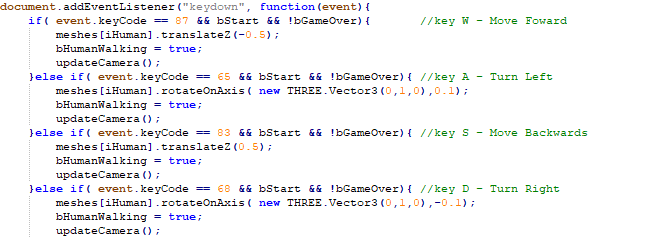


Figure The translation rotation mechanism

**Note!** The camera always follows the main character from behind as a first person shooter game thus its updated along with the movements.

## The follow mechanism

The follow mechanism, exists in the eventListener.js file and works in a very comprehendible way. When one of the z,x,c keyboard keys are pressed the “calculateDistance” function is called which measures the distance between the main character and the character that corresponds to every key. If the distance is smaller than a certain threshold then the selected character is assigned an exact distance from the main character and follows him whenever he goes.

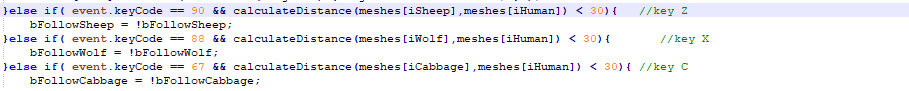


Figure The follow mechanism activator – deactivator

**Note!** The calculated distance is the Euclidean distance between 2 points in 3D space given by the equation:

## Background Environment

In order to visualize the mainland into space we used a skybox as it’s called, which is nothing more than a box containing the mainland whose 6 faces are assigned desired texture. It is a very efficient and realist way to realize a 3D vast environment. The skybox textures were picked from (Custom Map Makers, n.d.).

# Implemented interactions

The possible interactions that the user can apply is through the keyboard as follows:

|  |  |
| --- | --- |
| **Main Character Controls** | |
| **Movement Controls** | |
|  | **W:** Forward,  **L:** left, **S:** Backwards, **R:** Right |
| **Game Commands** | |
|  | **Z,X,C :** Activation keys for follow and unfollow (applicable only at a minimum range away of the object) |

Table Interaction Controls

Also, the user can click on the top left icon as shown in Figure 1 which pops up an information window that explains the purpose of the game as follows.

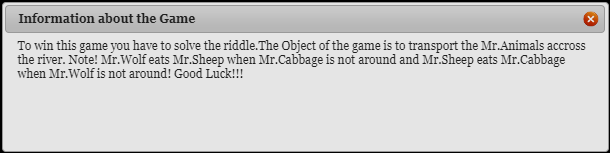


Figure Pop up box

# Bibliography

(n.d.). Retrieved from Blender Foundation: https://www.blender.org/

(n.d.). Retrieved from Custom Map Makers: http://www.custommapmakers.org/skyboxes.php

(n.d.). Retrieved from LionStudiosTM: https://www.youtube.com/user/xVxMARKxVx/featured

Stemkoski, L. (n.d.). Retrieved from stemkoski: http://stemkoski.github.io/Three.js/index.html