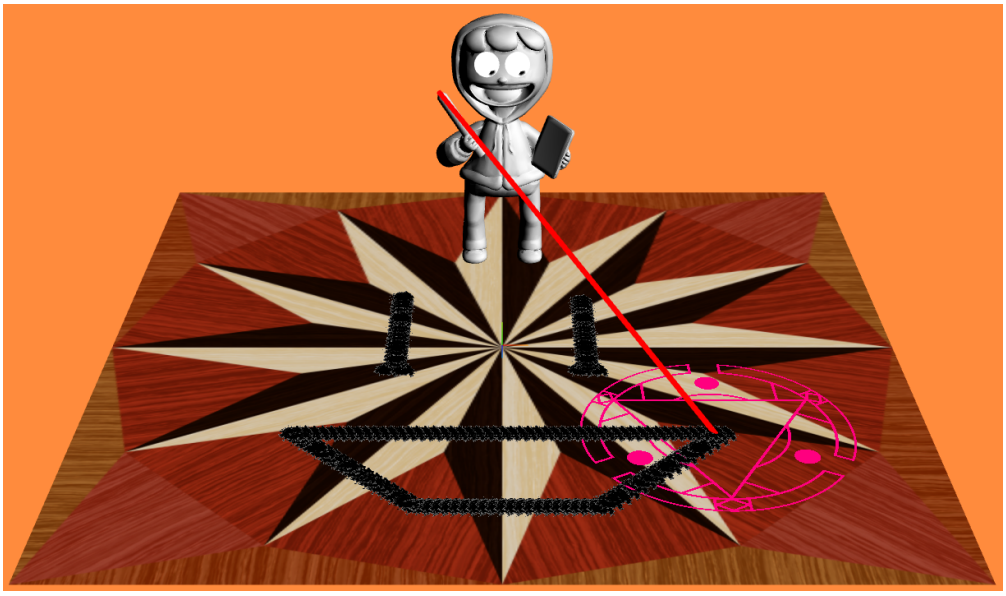
CPSC 314
Assignment 2, Part 1: Magical Transformations

Due 11:59PM, February 7th, 2020



1 Introduction

1.1 Getting the Code

Assignment code is hosted on the UBC Students Github. To retrieve it onto your local machine navigate to the folder on your machine where you intend to keep your assignment code, and run the following command from the terminal or command line:

```
git clone https://github.students.cs.ubc.ca/cpsc314-2019w-t2/a2-part1-release.git
```

1.2 Template

- The file `A2-P1.html` is the launcher of the assignment. Open it in your preferred browser to run the assigment, to get started.

- The file `A2-P1.js` contains the JavaScript code used to set up the scene and the rendering environment. You will need to make small changes in it to answer the questions.
- The folder `glsl` contains the vertex and fragment shaders for the wizard, eyes, scorch mark geometry, and magic circle geometry.
- The folder `js` contains the required JavaScript libraries. You do not need to change anything here.
- The folder `obj` contains the geometric models loaded in the scene.
- The folder `images` contains the texture images used.

2 Work to be done (40 points)

In this Assignment you will utilize your knowledge of transformations to make things move. Please note that this assignment is split into two parts, this is the first part, the second will be released shortly. Your grade for A2 will be determined by the sum of both parts. First, ensure that you can run the template code in your browser. See instructions in Assignment 1. Study the template to get a sense of how it works.

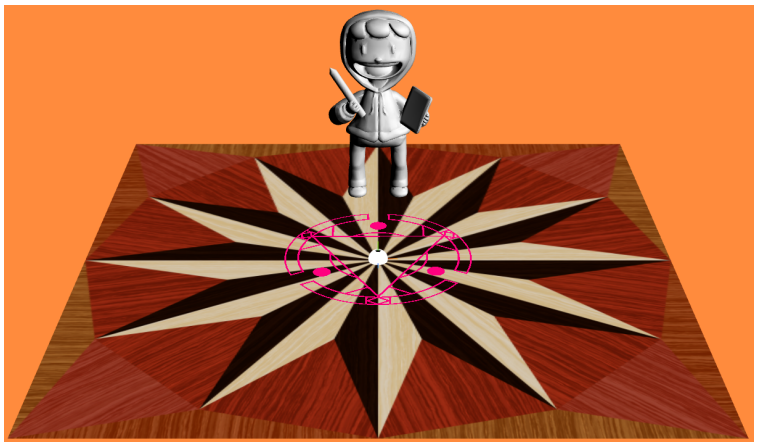


Figure 1: The initial scene.

Required Features

1. **(15 points)** Curious Wizard
   The wizard has become curious about the movements of magic circle, and has decided to watch it closely. You task for this question is to position and rotate the eyes of the wizard. Once they are correctly positioned, make sure that the wizard's eyes always 'look at' the magic circle. Your tasks are:

(a) Attach the provided eyes to the wizard. Two offsets have been provided for you that will position the eyes accordingly, you will need to pass these offsets to the shader `eye.vs.glsl`.

(b) Second, you need to make the eyes orient to look forward, you can use a simple rotation matrix for this.

(c) Lastly, you need to make the eyes orient to follow the magic circle, as shown below. You will need to modify `eye.vs.glsl`. One way is to create your own **modelMatrix** in the shader in file `eye.vs.glsl`.

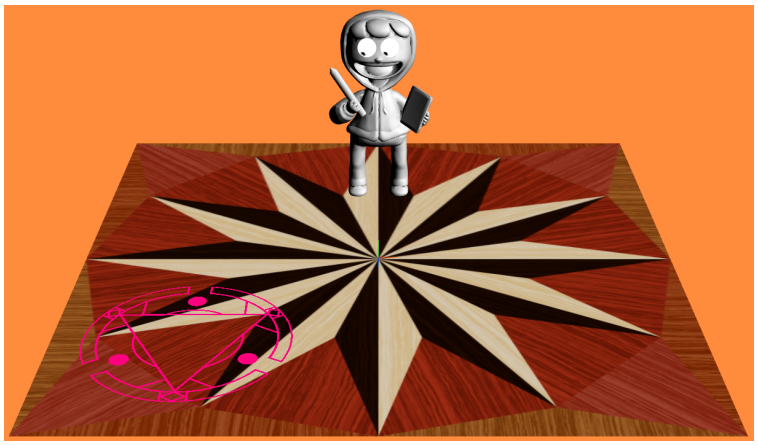


Figure 2: Orienting the eyes of the wizard.

Hint 1: The scaling matrix is already given, you only need to create your own translation matrix and rotation matrices.
Hint 2: The order of matrix multiplication is very important here.
Hint 3: Try to relate the question with the lookAt matrix.
Note: To get full credit for this question, you must build these matrices yourself.

(25 points) Magic Laser!
In this question, you will make the wizard cast a laser beam spell from their wand, to the center of the magic circle. Your tasks are:

(a) Create one laser from the tip of the magic wand that the wizard is holding to the magic circle when `SPACEBAR` is pressed. When `SPACEBAR` is released the laser should disappear, and when it is held down the laser should follow the magic circle. You will need to create the shaders yourself, however both the laser geometry and offset to the wand are provided. See code for `laserGeometry` and `offsetWand` in `A2-P1.js`. Once you have this working, you can comment in the scorch mark code for a neat effect in the function `checkKeyboard()`.

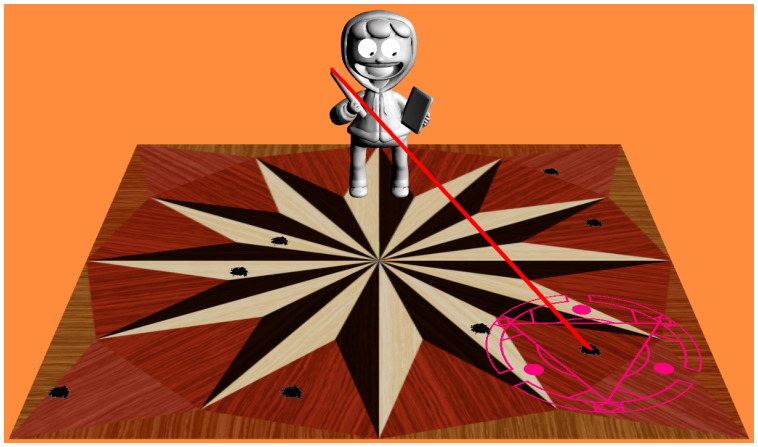


Figure 3: Shooting a laser.

Hint 1: The scale of the laser is the distance between the wand tip and the center of the magic circle.
Hint 2: You may use a similar approach to orienting the eyes for the laser.
Hint 2: The order of matrix multiplication is very important here.
Hint 4: The laser geometry has already been translated

Creative License
This part will be covered in the second part of this assignment.

2.1 Hand-in Instructions

You will be handing in both parts of A2 together. You do not have to hand in any printed code. Create a README.txt file that includes your name, student number, and CWL username, and any information you would like to pass on to the marker. Create a folder called "A2" under your "cs-314" directory. Within this directory have three subdirectories named "part1,""part2", and "creative" and put all the source files, your makefile, and your README.txt file for each part in the respective folder. The assignment should run without any changes directly from your submission folders. The assignment can be handed in on a department computer, which you can SSH into, with the exact command:

```
handin cs-314 A2
```

You may also use `Web-Handin` by following this link `https://my.cs.ubc.ca/docs/hand-in`, logging in with your CWL credentials, and writing cs-314 for the course, A2 for the assignment name, and zipping your assignment folder for submission.

It is always in your best interest to make sure your assignment was successfully handed in. To do this, you may either use the `Check submissions` button in Web-Handin, or using the -c flag on the command line `handin -c cs-314 A2`.