

**Report for Computer Graphics Assignment 1**

**COSC363**

**Kai Koh**

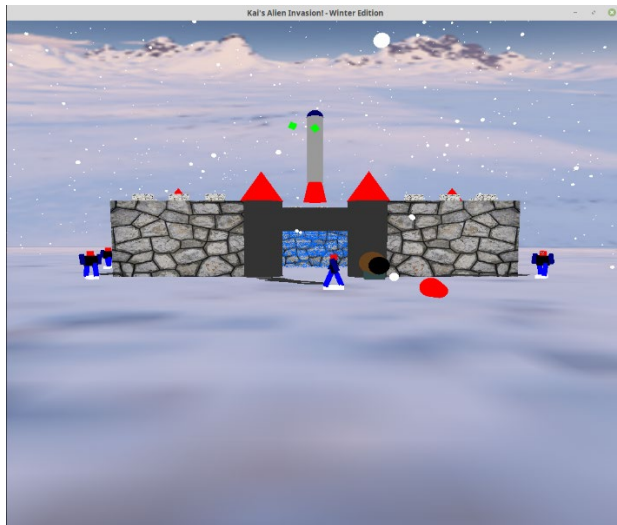
**64276083**

**Hkk18**

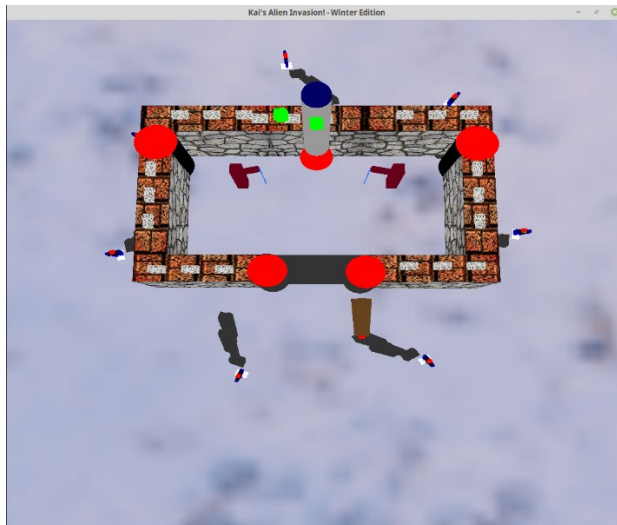
## 1 Scene Description

Winter is coming...and so are the Aliens! On this snowy day, Robots-on-Skis (yes they are skiing!) are on patrol, skiing around the perimeter of the castle. On your command, the Cannon-of-Death will fire at the oncoming onslaught of Aliens, while the Spaceship-of-Destiny will be able to evacuate you should it be necessary. The spaceship contains state-of-the-art revolving sensor to detect any imminent dangers and will avoid them, so you will be totally safe while on the run! The waterfalls are on energy-saving mode until you decide to unleash the waterfall curtains. As the snowstorm brews, are you prepared for the invasion of Snowakanda, ol' mighty Player One?

## 2 Screenshots



This is a screenshot showing the front view of the Particle Systems of Snowfall and Waterfall Curtain, Launching of Spaceship, Firing of Cannon and Robots patrolling the perimeter of the Castle.



This is the alternate view (spaceship view) when HOME button is toggle once showing the top view, as seen from the spaceship, with the Waterfall Curtain and Snowfall toggled off. Also depicts shadows of the patrolling robots.



Screenshot showing the back view as well as light source.

### 3 Extra Features

- a) Implemented planar shadows of Robots by using GL\_Lighting and reusing robot function but changing the color to shadow-grey.
- b) Spotlight is cast on Robots, hence the shadows.
- c) The Waterfall Curtain Particle System when toggled off or by default, is an animated water tap that is spilling out water on both sides of the interior of the castle.
- d) Implemented the two camera modes, as seen from the first and second screenshots, by setting camera height and angles for each toggle. As spaceship lifts off, the second camera mode displays the view of the castle beneath it.
- e) Physic Models
  - Used Gravity and Velocity to calculate and display speed of Snowfall. Equations used are  $\text{snowfall}[\text{loo}].y\_pos += (\text{velocity} / (\text{decrease\_speed} * 1000))$  and  $\text{velocity} += \text{float}(\text{GRAVITY})$ .
  - Used Gravity to calculate and display displacement of water droplets. Equation used is  $\text{sprinkler}[\text{i}].y\_change += \text{float}(\text{GRAVITY}) * \text{sprinkler}[\text{i}].\text{mass}$ .
  - Gave mass to water droplets. Equation used is  $\text{sprinkler}[\text{i}].\text{mass} = \text{float}(0.5 + 0.5 * \text{randomize}())$ , where  $\text{randomize}()$  is a function to randomly generate numbers.
- f) I did not manage to generate any sweep surfaces.
- g) A surface shape generated could be the head of the spaceship where I used  $\text{glClipPlane}$  to form the head.
- h) I did not manage to successfully implement collision detection.
- i) I have used the AntarcticDawn-themed skybox TGA images from the lab to implement the Skybox texture.
- j) I have implemented 2 Particle Systems, Snowfall and Waterfall Curtain as seen from the screenshots above.

### 4 Challenges

Snowfall may sometimes leak outside of predefined skybox area. I have added an IF-ELSE condition to check if the displacement of the snow will land outside the area to solve this bug.

There is also a noticeable reduce in speed when snowfall is toggled on, which I was unable to find a direct fix for. I have tried reduced the TOTAL\_SNOW from 10000 to 1000 which reduces the lag, which means it could be a mere processing capacity of the computer instead of an actual bug in the code.

Initially, at any point of time, only one object can animate; if spaceship is moving, and I pressed the button to fire a bullet, the spaceship will pause while the bullet moves, and vice versa. I have since fixed it by giving two separate timer functions for glutTimerFunc to recognize respectively, bullet\_timer and spaceship\_timer.

I had to add float() to values with decimal places in order to suppress the implicit conversion warnings. Also, I was unable to suppress the warnings of 'no previous extern declaration for non-static variable' for certain variable declarations.

## **5 Controls**

### **Camera Controls:**

HOME: Toggles between default view and spaceship view.

UP: Zooms camera forward in the current direction.

DOWN: Zooms camera backward in the current direction.

LEFT: Pans camera slightly left.

RIGHT: Pans camera slightly right.

X: Angles camera slightly upwards.

Z: Angles camera slightly downwards.

### **Interactive Controls:**

C: Fires bullet from cannon.

L: Launches spaceship.

S: Toggle Snowfall.

R: Restarts launch of spaceship. (In case User somehow missed the animation)

W: Toggles creation of waterfall curtain.

ESC: Closes the application window.

*Note: Please check README.md for the usage guide of compiling and running the project as well.*

## **6 References and Resources**

- Google and Stackoverflow have been the greatest help for addressing certain doubts like the type of parameters each gl/glut function takes in, as well as learning how to develop a Particle System.
- COSC363 labs for TGA images, skybox textures and for how to code in C++/OpenGL in general.