**AGP Part 2 – Project Overview**

**B00241075, B00234101, B00236297**

Our AGP Part 2 application is a combined effort from all three of us and is an extension from our individual projects on lighting, particles and geometry shaders respectively. Our task was to take all of these tutorials, extend them and then combine them together along with adding in a game element and a new feature. We have created an application that takes a 3D model with various lights involved in the scene, all of the objects in the scene will have shadows due to the lighting and shadow mapping, the objects in the scene will be bump mapped using geometry shaders. There will also be rain and snow particles in the scene that will collide with the ground and the rain will bounce off the ground whilst the snow will lie on the ground and form a layer of existing snow.

**User Instructions**

* W – Move forward
* A – Move left
* S – Move right
* D – Move backwards
* R – Move upwards
* F - Move Downwards
* < - Rotate left
* > - Rotate right
* Z - Turn Z-fail on/off
* T – Turn Shadow Volume drawing on/off
* Y – Turn Silhouettes on/off
* P - Turn point lights on/off
* I - Turn ambient light on/off
* 5 – Toggle rain on
* 6 – Toggle snow on

We have all added a multitude of features from all of our individual part 1 projects and then combined it into our part 2 applications:

In part 1 we had multiple lighting, including different kinds of light, such as point lights and spot lights. We added to this by implementing robust shadow volumes to display shadows for meshes. There are two different algorithms we can use to generate the shadow volumes, namely the z-pass and the z-fail algorithms. Z-pass is more efficient but do not work in all situations; z-fail works in most situations but is less efficient. Both have been implemented in ours and can be switched between, however the z-fail algorithm, does not work correctly at the moment. In order to realize the algorithm correctly we also needed to add ambient lighting. This light illuminates the entire scene uniformly. In order to generate the shadow volumes we needed to determine the silhouette of the object in relation to a light source. To do this some features had to be implemented including generating adjacency indexing for the meshes, and silhouette detection and extrusion implemented in a geometry shader.

We did Normal Mapping. This gives our objects and textures a more realistic feeling by giving them an effect of depth and more textured.

In the particles section, from part 1 to part 2, we transformed the way they were displayed completely. They were originally in 2D space effectively because they were tied to the camera, now they are in full 3D space and can be maneuvered around and rotated around to view them from different angles. In order to make all of the changes required to add all of the additional features for part 2, we had to separate the snow and rain classes completely, this took a while to do successfully as we had to remove the whole particles class and piece together the snow and rain classes from that. For the first manipulation of pat 2, we had both the snow and rain doing the same thing as they fell to the ground. On impact, they would create another 10 particles that would be used for a splashing effect (like when rain hits the ground) and would accelerate and move in a parabola until the drops hit the ground level again, then they would disappear. However, we decided that this wasn’t very realistic for snow so this was manipulated for the second phase. The snow was altered so that when it hit the ground, it no longer bounced and splatted like the rain, it now lay on the ground to make it more realistic. Though it wasn’t realistic for 10 snow particles to be generated when one landed, so only the particle that lands, lies on the ground and another particle is created in the sky for it to fall again to create a blanket of snow covering the ground. This means that the snow lies on the ground and the rain bounces and splashes off the ground to create more realistic particle weather effects.