**CMP203 – Submission**

**Fraser Barker - (1600196)**

* Controls.
  + **1** – **4** to switch between various texture filtering modes.
    - **1** – Point Sampling
    - **2** – Bilinear
    - **3** – Trilinear
    - **4** – Point on near/Trilinear on far
  + **7** – **9** to switch between various cameras within the scene.
    - **7** – Default camera “Free Camera”.
      * Can move via **W**,**A**,**S**,**D** keys, up with **Spacebar**, down with **C**.
      * Can rotate via the mouse.
    - **8** – Tram camera
      * Can rotate via the mouse.
    - **9** – Door camera
      * No movement/rotation allowed.
  + **W, A, S, D, Spacebar, C** are used to move the camera around the scene.
    - **W** – Moves the Free Camera forwards.
    - **S** – Moves the Free Camera backwards.
    - **A** – Moves the Free Camera left.
    - **D** – Moves the Free Camera right.
    - **Spacebar** – Moves the Free Camera up the Y-axis, rotation independent.
    - **C** – Moves the Free Camera down the Y-axis, rotation independent.
  + **I** and **K** to move the tram backwards and forwards within the scene.
    - **I** – Moves the tram in the negative X direction (left).
    - **K** – Moves the tram in the positive X direction (right).
  + **Q** and **E** to open close the door in the scene.
    - **Q –** Opens the door once the door locks are retracted.
    - **E** – Closes the door.
  + **Mouse** movement is used to rotate a camera if it’s designed to rotate.
  + **F** allows the user to turn wireframe mode on/off.
  + **R** resets every aspect of the scene.
* How I met each aspect of the coursework brief.
  + (A good coursework should include all of the following features.)
  + The scene must show lighting from multiple lights of different types, colours and some animated.
    - I have **7** lights within my scene. 2 are above the door, one on each side of the tram, one in each tram dock and another to give the scene a little light.
    - The two lights above the door are **spot** lights, which rotate and become enabled in tandem with the door opening (i.e. when the user presses and holds **E** or **Q**). They are both **red** in colour.
    - The lights on either side of the tram are also **spot** lights and are enabled when the user moves the tram (via pressing **I** or **K**). They are both **white** in colour.
    - The lights within each tram dock are **point** lights. They are always enabled. They are both **yellow** in colour.
    - The scene light is a **point** light. It is always enabled. It is **white** in colour.
  + The scene must show use of texturing. Additionally, demonstrating texture filtering.
    - I have **5** (technically 7) **different textures** within my scene.
      * **Hazard Texture** – Is the texture I apply to the tram rail.
      * **Wall Texture** – Is the texture I apply to each wall.
      * **Grate Texture** – Is the texture I apply to both parts of the walkway and also where I utilise some transparency/alpha blending effects.
      * **Door Top Texture** – Is the texture I apply to the top of the door. (There is also a flipped version of this I apply to the other side of the door.)
      * **Door Bottom Texture** – Is the texture I apply to the bottom of the door. (There is also a flipped version of this I apply to the other side of the door.)
    - I allow the user to switch between various texture filtering modes via the keys **1** – **4**. It’s not very noticeable on the textures I have chosen however if you look closely at any one of the walls you will notice a slight difference between each selected texture filtering mode.
      * **1** – Point Sampling
      * **2** – Bilinear
      * **3** – Trilinear
      * **4** – Point on near/Trilinear on far
  + A working camera. The user must be able to manipulate the view through using the mouse and keyboard to control the camera. Additionally, you should provide multiple cameras each with a different focus such as limited controls, fixed views, procedurally controlled views or different camera types.
    - I have **3** different cameras within my scene.
      * 1 – **Free Camera**. This is the default camera selected when the scene starts. The user can fly around the scene using the WASD, Spacebar and C keys. Rotation is controlled via the mouse.
      * 2 – **Tram Camera**. This is a camera which position is bound to the location of the tram (i.e. It can only be moved via **I** or **K** being pressed). It can be rotated via the mouse.
      * 3 – **Door Camera**. This camera can neither move nor rotate. It is purely a static camera which is focussed on looking into the reflective plane behind the door.
  + A clear example of using the matrix stack for Hierarchical modelling and animation through hierarchical means.
    - I utilise the matrix stack in various places throughout my scene. One example is where I use it within my light setup so that when the door lights are rotating they do not affect the other lights within the scene.
    - [ CODE HERE]
  + Use of transparency effects / Alpha blending.
    - I use a transparency effect/alpha blending on my walkway within the scene.
    - [ PICTURE HERE ]
    - [ CODE HERE ]
  + Use vertex arrays (not including model loading).
    - I utilise vector<Vector3> to store my procedurally generated shapes within the scene.
  + Models loaded in from an external file.
    - I load two models in from an external file
      * 1 – The tram within the scene.
      * 2 – The crowbar in the door room.
  + Examples of Procedurally generated shapes.
    - I procedurally generate the door locks which are in front of the door. They are composed of a cylinder and a disc.
    - [ CODE HERE ]
    - [ PICTURE HERE ]
  + User interaction (controlling objects in the scene other than the camera).
    - The user can control the tram (**I** or **K**).
    - The user can open/close the door (**Q** or **E**).
  + A wireframe mode.
    - Pressing the **F** button allows the user to turn wireframe mode on/off.
  + Advance features such as shadows and use of the stencil buffer.
    - I incorporate both shadows and reflection utilising the stencil buffer within my scene.
      * The tram and railway have a shadow.
        + [ PICTURE HERE ]
      * The reflective plane in the door room behind the door reflects **7** objects.
        + [ PICTURE HERE ]
  + The application should be carefully designed and constructed showing appropriate use of classes and well commented.
    - I comment and utilise tidying functions to make navigation around my program much easier.
* References:
  + Helped me to utilise the stencil buffer to cull the shadow once it had gone over the plane it was projected onto boundaries: <http://artis.imag.fr/Recherche/RealTimeShadows/pdf/stencil.pdf>
  + For the crowbar model: <https://www.turbosquid.com/FullPreview/Index.cfm/ID/622279>
  + Gave me some insight into loading .mtl files although I have yet to have it fully implemented: <https://xiangchen.wordpress.com/2010/05/04/loading-a-obj-file-in-opengl/> & <https://www.youtube.com/watch?v=yfsVBh2AaA8>
  + For the tram model: <https://3dwarehouse.sketchup.com/model/fce48e6cd0f81c0c873433210e278bfe/Half-Life-Tram>
  + HL Textures: <http://25.media.tumblr.com/240d2e8c8ecc6e97807ffbfe06be1b28/tumblr_mwmqccBQJ81s1rufio1_1280.jpg>
  + Grate Texture: <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcR10ZPSaY8cjLKK_Ek2QGGTfswKkg1YJOK-gnZ6edek_-Q_8QqC>
  + Wall Texture: <http://media.moddb.com/images/articles/1/103/102706/auto/dt_brick_cr.jpg>