COSC422 – Assignment 2

James MacKay

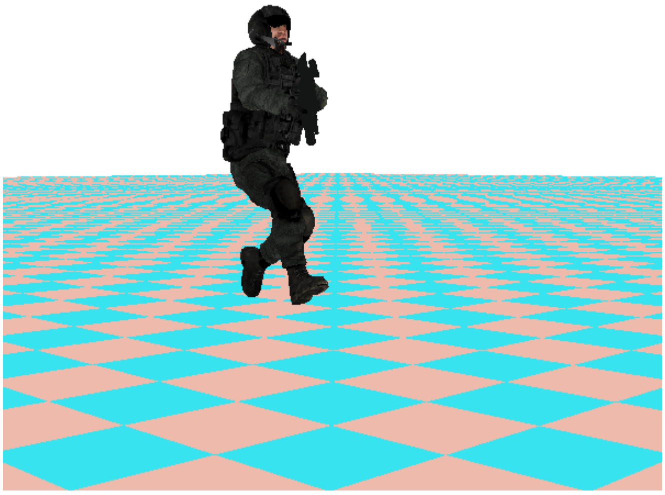

Figure 2: Each character model walks along an "infinite" plane at various speeds

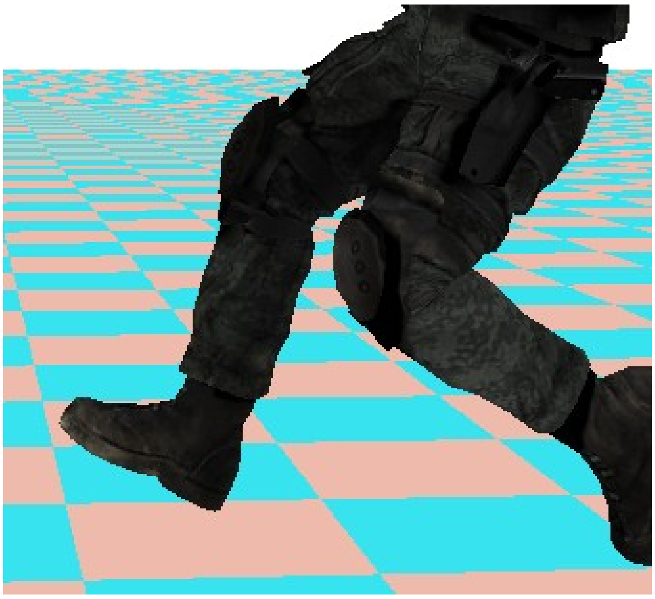

Figure 1: Blended bone weights showing smooth transitions between meshes

The assignment is provided as a single C++ source file and can be compiled with the following:

```
g++ -Wall -o Assignment Assignment.cpp -lGL -lGLU -lglut -lGLEW -lassimp -lIL
```

And the program run by executing the generated file: `./Assignment.` Alternatively, the entire program can be built and executed by running the `build_and_run.sh` file provided.

The three models (Army Pilot, Mannequin and Dwarf) can be cycled through with the use of the spacebar. The camera is moved forwards and backwards with the use of the up and down arrow keys respectively and rotated around the figures with the left and right arrows.

While viewing a character model in a walking motion, the floor plane underneath them moves in the opposite direction (like a treadmill) to give the illusion of movement. Both the camera and the models remain stationary.

The first model presented to the user is the army pilot. This model is provided in the `ArmyPilot.x` file and includes a built in walking sequence displayed. Additionally, the model is textured with the provided images. His gun however remains a solid colour.

The file also provides a weight included for each bone in the model. In order to maximise realism, the influence of multiple bones are blended

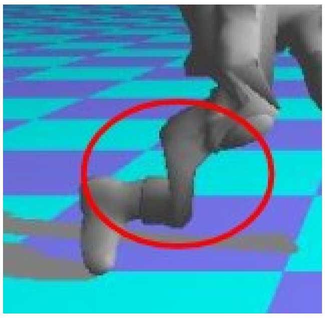

Figure 3: Unblended vertices

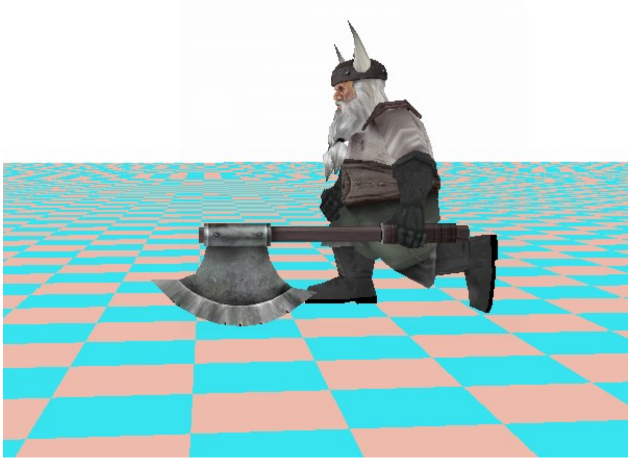

Figure 4: Dwarf displaying the kneeling action

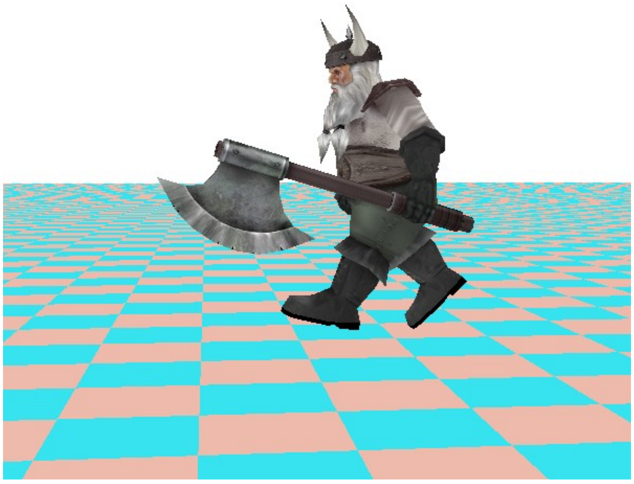

Figure 5: Dwarf displaying the walking animation

(based on these weights), avoiding problems which arise from using only the influence of a single bone (compare Figure 2 (unblended) with Figure 1 (blended)).

Pressing the spacebar reveals the second model, a running mannequin. Untextured, this model is also a solid colour. The animation for this model was provided in a separate file to the model itself, but as they share identical join hierarchies, bones are easily matched to meshes in order to animate the figure.

A further press of the spacebar reveals the final model, the dwarf. Like the army pilot, this model has a built-in 'kneeling' animation. Notice that the floor plane does not move while the dwarf is stationary. Pressing the '2' key triggers the second character animation, the walking animation. This was provided in a separate file (`avatar_walk.bvh`) with a different joint hierarchy than the model itself. In order to correctly present the walking animation onto the dwarf, a 'mapping' was done between nodes in the model's joint hierarchy and those in the animation itself. This mapping was performed based on the indices of each node. Those nodes which were not mapped to (namely, the top half of the dwarf) remained playing out the original kneeling motion. To revert back to the kneeling animation, simply press the '1' key.

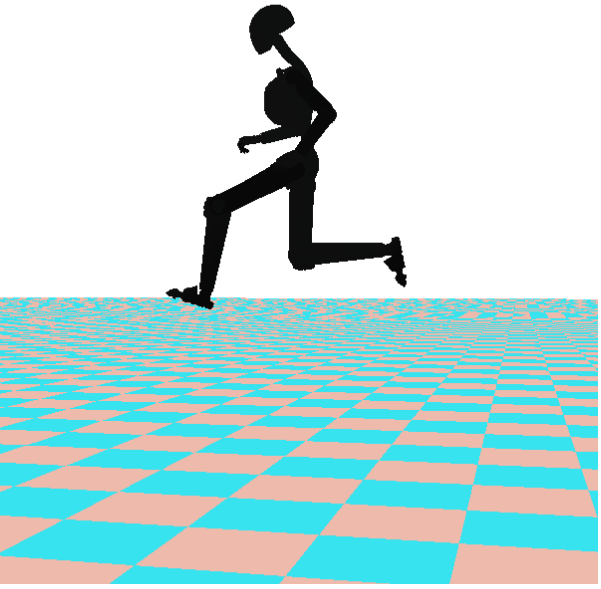

Figure 6: The running mannequin figure has some issues with lighting

This part (mapping the animation to the model) was particularly difficult. Not least because each animation sequence lasted for a different number of 'ticks'.

A second difficult (and quite possibly misguided) decision regarding the inclusion of each model in a single executable proved to be particularly vexing. Keeping track of multiple tick values for a number of models and animation sequences became quite complex. The resulting code is somewhat obfuscated in its intention and function.