

# COSC422 Assignment 2 Harry Dobbs

## Task 2.1 – Animation of embedded motion data (6 Marks):

Brief Outline: I have implemented various methods to ensure the animation looks realistic. The main methods used include:

- Position Interpolation – The ArmyPilot.x animation is made up of rotation and position keys. Each of these keys correspond to a point in time. However sometimes the time might be between the time of position keys. This time can be used to work out an interpolated position between the two keys dependant on the time. An interpolated position vector will be generated and then used to generate the position matrix.
- Rotation Interpolation – This is a similar concept to position interpolation except instead of finding a position vector it will find a quaternion which is then used to generate the rotation matrix.
- **Bone Weighting (3 Marks):** An algorithm was implemented that allows for vertex blending. This works by associating the vertices with the neighbouring bone matrices. This prevents weird unrealistic distortion of the bones.
- View Frame Tracking – The animation has been programmed to happen at the origin. The camera can rotate around this point whilst zooming in and out this is done by altering the FOV parameter and performing `glRotatef()` to the scene.
- A moving plane (1 Mark) was implemented to simulate movement of the model.

Problems:

- I had an issue with the original rendering of the model being heavily distorted this was due to incorrect usage of the transpose method. I used the transpose method as a method that returned the transposed matrix rather than one that applied it to the current matrix. This was solved by correctly understanding how the method works.



Figure 2: Front on View of Army Pilot with Vertex Blending



Figure 1: Side on View of Army Pilot with Vertex Blending

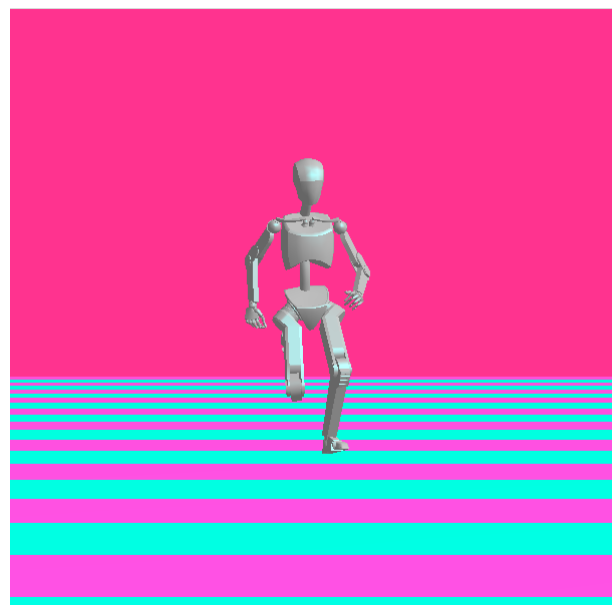
## Task 2.2 – Animation of rigged character model using a separate motion file (6 Marks):

Brief Outline: This part of the assignment used much of the code from the first part of the assignment. The main additional methods included in this part of the assignment were:

- Getting the rotation keys from the animation “run.fbx” file and applying these transformations to the model “Mannequin.fbx”. This involved loading two fbx files and collecting the required data from the “run.fbx” file and applying the transformations to the “Mannequin.fbx” file.
- A moving plane (**1 Mark**) was Implemented to simulate movement of the model. This was done by constantly updating the vertice’s positions to make the plane move in the opposing direction to the runner.



*Figure 3: Side on View of Mannequin*



*Figure 4: Front on View of the Mannequin*

## Task 2.3 – Animation Retargetting (6 Marks):

Brief Outline: This part of the assignment required specific channels to override channels of another animation. This would allow the upper body animation to remain the same whereas the lower body can be selected to either walk or kneel. The main methods included in this part of the assignment were:

- Analysing the channels of each animation sequence and finding the appropriate corresponding channel to map to. I used the print functions to find the channel numbers of the corresponding nodes.
- I would override the inbuilt lower body animation with the external animation
- As each of the animations had a different time period the tick had to be linearly adjusted to find the corresponding time in the animation. I did this by multiplying by a constant. The constant was calculated by dividing the inbuilt animation tick duration by the external animation tick duration.
- The floor plane will stop moving when the kneeling animation is being played.



*Figure 5: Dwarf doing the walking animation*



*Figure 6: Dwarf doing the Kneeling Animation*

Problems: The main problem I had with the Dwarf animation is that it would crash everytime I retargetted the channel. This was because after the rotation transformation of each channel I need to reset the animation back to the scene animation (lines 416 – 420) aswell as reverting the tick value to an applicable one for the scene animation. I solved this by carefully reading through my code.

## Controls

Button	Action
Left Arrow Key	Rotate Left
Right Arrow Key	Rotate Right
Forward Arrow Key	Zoom in
Backwards Arrow Key	Zoom out
1 (Dwarf Only)	Kneeling Animation( Dwarf Only)
2 (Dwarf Only)	Walking Animation (Dwarf Only)

## References:

Dr. R. Mukundan . (2019) . Computer Graphics COSC422 [PowerPoint Slides] [Animation Data]. Retrieved from <https://learn.canterbury.ac.nz/mod/resource/view.php?id=803416>

No additional files used.