INFO6015 – Animation 1: Final Exam

Thursday, December 15th, 2011, 10:00 AM to 10:00 PM

Instructor: Michael Feeney

The exam format:

- The questions are ***not*** of equal weight. There are three (3) pages with seven (7) questions with a total of 180 marks (if my math is correct)

- The answers may be one or a combination of the following:

  - Short answer (in your own words)
  - Snippets of code
  - Complete running solutions

- CLEARLY indicate which answer goes to which question. My suggestion is that you place each answer in its own folder, named "Question_01", "Question_02" and so on (or something equally clear). Another option is to create a Visual Studio solution and add a number of projects – one per question – to it.

- Place any written answers into a Word, RTF, or text file. Again, *clearly* indicate which question you are answering.

- If you are combining answers (which is possible), please indicate this with a "readme" file or some note (not buried in the source code somewhere) indicating this.

- For applications: If it doesn't build and run; it's like you didn't answer it. I'll correct trivial, obvious problems, but you need to be sure that it compiles and/or runs.

- You have until 10:00 PM on Thursday, December 15th to submit all your files to the appropriate drop box on Fanshawe Online; note that it is not written to take twelve (12) hours, it's just that you have that long to complete it.

- This is an "open book" style exam, but keep in mind that there is a difference between "collaboration and learning" and "plagiarism," so:

  - There may be some slight customization on each test. In other words, they will be the same "type" of questions – i.e. I'm looking that you can demonstrate a particular thing – but the specifics are different
  - I will be very suspicious of code that is virtually identical.

- You can reach me:
  - In my "cubicle of destiny" most afternoons this week (I'm invigilating in the mornings)
  - On the office phone: (519) 452-4430 x4798
  - My cell 519-494-7569
  - Or through e-mail (mfeeney@fanshawec.ca), of course.

Questions:

1. (10 marks) Is there a technical reason why you couldn't implement your main character in a modern game as an MD2 or MD3 model?

2. (10 marks) What are some of the technical limitations to MD2 and MD3 models (in other words, why aren't they still used all the time for modern games)?

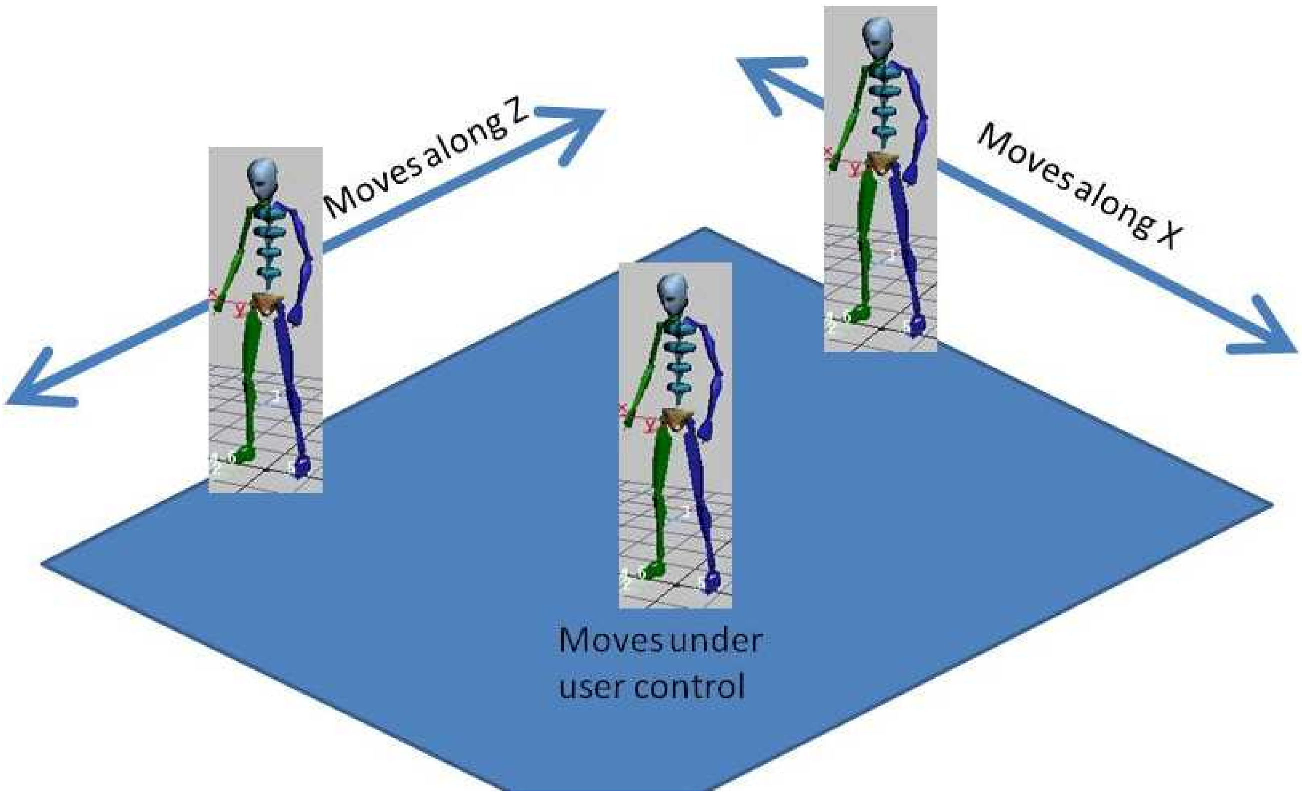
3. (50 marks) Implement a scene with three biped models that are animated using BHV files. One should be automatically moving back and forth along the x axis, another moving back and forth along the z, and the remaining one should be under user control. They should not be able to move along the y axis (up and down).

   The two bipeds that are moving automatically should be facing the "correct" way for whatever animation they are running.

   The alignment, speed of motion, etc. should look reasonably correct for the animations you are running. For example, you shouldn't be using a "climbing a wall" or "getting punched out and falling down" for your animations.

   (10 marks) They are all using *different* animations.

   Something like this:



4. (**Bonus:** 15 marks) Alter your answer to question three (3) so that the user controlled biped is turning to face the direction of motion. In other words, instead of being able to "slide" it in the XZ plane, you would have a "turn left" and "turn right" keys along with "forwards" and "backwards" keys. Note that this isn't a "turn 90 degrees," but rotates a little bit while the key is being pressed. Also note: part of the bonus is to determine an appropriate way of handling going "forwards" then immediately "backwards."

5. (50 marks) You want to create some class that assists in the smooth transition (LERP) between the *last* frame of one BVH file animation and the *first* frame of the next BVH file.

   You have been given the following requirements for your class:

   i) It has a method that takes two BVH file names.

   ii) It has a method that takes two strings, indicating the joint name and channel name, along with a float that takes a number between 0.0 and 1.0. This will return a float indicating the linearly interpolated value between the *last* frame of the *first* BVH and the *first* frame of the *second* BVH for the corresponding joint and channel.

   For instance, passing a value of 0.0 will return the angle from the last frame of the first BVH file, where passing 1.0 will pass the angle from the first frame of the second BVH file.

   Passing a value of 0.5 will return the 50% linearly interpolated (LERP) joint & channel value between these two frames, and so on.

   iii) You may use an existing BVH loader if you wish.

   iv) You should assure that both BVH files have the same joint & channel names.

   Note: <u>You need to show that these values are being calculated correctly.</u> You can either load a complete 3D scene that LERPs slowly between the two BVH files, or you can create a console application that outputs some example data.

6. (25 marks) Alter the answer in question 5 (five) by making a third method that takes a single float (with a value between 0.0 and 1.0) and returns a vector of string-string-float triplets (i.e. a class consisting only of two strings and a float). In intention is that this vector would contain the entire list of joint + channel names in the BVH file, along with the interpolated values, when being passed a single LERP value representing the transition between frames (in other words, instead of passing each joint & channel name for method ii) in question 5, you would send a *single* float and receive *all* the data at once).

7. (10 mark) Why it is generally a Good Idea (or why is your life as a programmer easier) when you omit scaling in a scene graph?

That's it.