



INFR 2310: Computer Animation Algorithms & Techniques

Course GDW Requirements

Report Due: Monday, December 7 at 11:59 PM

Demo: Week of December 7 (TBD)

Worth: 15% of your final grade (10% demo, 5% report)

Overview

For this component of your grade, you will be marked on your integration of concepts related to this course in your GDW demo (the **demo score** worth 10% of your final grade). You will also be marked on a brief report explaining your implementation (the **report** worth 5% of your final grade).

Demo Score

You will be **REQUIRED** to implement the following four features in your framework and use them in your demo:

1. Linear interpolation (LERP)
2. Spline (curve) interpolation with speed control (Catmull-Rom and/or Bezier)
3. Locomotion control system
 - If you are not making a first- or third-person game, then some basic form of pathfinding/steering for characters
4. Morph targets for FK animations created in Blender/other 3D software
 - If you choose to implement skinning, this requirement is waived

If any of these components do not make sense for your game, you MUST email me as soon as possible and we will make alternate arrangements to suit the design of your project.

In addition to the required components, you will need to choose at least one of the following optional features to implement. You can choose from either the Intermediate or Advanced list. If you implement one advanced feature only (no intermediate), you can earn up to 10% bonus on the demo score. If you implement at least one intermediate and one advanced feature or two advanced features, you can earn up to 20% bonus on the demo score.

Intermediate:

- Robust particle system (e.g., many customizable attributes)
- Robust scene graph (e.g., basic visual hierarchy editor)
- Path/keyframe animation editor
- In-engine animation blending

Advanced:

- Advanced particle system editor (e.g., lifetime curves)
- Advanced animation/cutscene editor (e.g., interactive timeline, save/load, etc.)
- FBX import (you may use an existing library) & skinning
- Visual FSM editor for animations
- IK implementation
- ...something else? (email me to see if your idea would qualify)

Demo Grading Scheme

Checklist /10

- Checklist of 4 required + 1 selected optional features (is it there?)

Quality Assessment /50

- Rubric on next page used to assess the quality of each of the 5 features

TOTAL /60

BONUS of up to 12 extra marks out of 60 will be assessed based on your inclusion of any advanced features (see above).

See next page for rubric used to assess demo quality.

Tips

- Prototype early! As topics are covered in class, transfer what you learn from programming in lectures and tutorials into your GDW project. Prototype how you will use this functionality in your game as early as possible.
- You can be creative with how you use different features. LERP doesn't mean you have to have something moving back and forth in a straight line on screen all the time - maybe you use it to fade between your title screen and your menu? Maybe you use it to change the colour of particles over time?

Demo Feature Quality Rubric

	Levels of Achievement			
Criteria	Unacceptable	Novice	Competent	Proficient
Technical Quality Points Possible: 5	0 to 2 Points Implementation is missing core necessary functionality Code is virtually unreadable	2 to 3 Points Implementation barely works and may have errors/bugs Code is poorly written, hard to understand, or badly structured	3 to 4 Points Implementation works with rare or very minor issues Code meets expectations for second year	4 to 5 Points Implementation works with no issues and may have additional functionality Code is very well-written, easy to understand, and sensibly structured within the project
Use and Polish Points Possible: 5	0 to 2 Points Implementation is not actually used in the game Related features of the demo (if there are any) are incredibly rushed	2 to 3 Points Use is contrived or “last-minute” Related features of the demo feel rushed	3 to 4 Points Use in the game is suitable Related features of the demo hang together well	4 to 5 Points Use in the game is suitable and may display some creativity Related features of the demo are well-polished given the development timeline

Report Score

You will need to submit a report explaining your implementation and use of each feature. Your report should probably be about 6-10 pages, though there are no strict length limits.

There is no set required format for this report. What I am looking for from your report is a brief, salient description of the reasoning behind your technical design and chosen use cases for each of the features showcased in your demo. You should also include references to the code files in your project that I should be looking at to assess your implementation.

You might want to structure your report something like this (just a suggestion):

- Title page
- 1 page per feature:
 - One-sentence explanation of what the feature does in your game specifically
 - Short paragraph describing your technical implementation, any details you feel are necessary about your specific code, some challenge you had to overcome, etc.
 - Reference to files containing core implementation details
 - Short paragraph explaining where the feature is used (if you applied the feature creatively, here's the place to brag!)
 - Screenshot showing one example of where the feature is used

Report Grading Scheme

Formatting /2

- Does the report look professional? Does it look like a good technical design document?

Writing Quality /3

- Are there any spelling/grammatical errors? Is the writing style appropriate?

Content /10

- 2 marks given per feature
- Is the information presented for each feature complete? Does it make sense?

Creativity /5

- 1 mark per feature
- Does the explanation for how each feature is designed and used in the game demonstrate some creativity?

TOTAL /20

Tips

- Treat this like a "technical" GDD of sorts - make it professional, but it doesn't have to be exceptionally formal
- I don't expect you to write a novel for this—you don't have to agonize over writing more than 300-400 words or so per feature. I expect that it should take you less than 8-10 hours in total (not per group member) to write an A+ report (probably around 1500-2000 words total) after you have your features implemented.