

# CSCI 191T Final Exam Study Guide

## Physics

Know how to use the various numerical integration schemes to compute the updated position/velocity, given the current position/velocity/acceleration and the time step  $dt$ .

Describe how the size of the time step  $dt$  affects the stability and accuracy of the simulation, and how these effects can be mitigated.

Understand the concept of “order” as it applies to numerical integration schemes.

Understand the concept of an integration scheme being “energy conserving”, why this is a useful property, and which schemes possess it.

## Graphics

Be able to explain the difference between an array-of-pointers representation and a row-major matrix representation for 2D images, and why one or the other representation is preferred.

Know how to index a pixel  $x,y$  in a row-major image of a given size.

Understand how matrix multiplication is used to simplify the transformation of 2- and 3-D points. Remember that we use column vectors!

Be able to recognize the four types of primitive transformation matrices.

Be able to apply a transformation matrix to a column vector by matrix multiplication.

Be able to compose transformations by multiplying their matrices, and understand the difference between pre-multiplication and post-multiplication.

Understand the purpose of the fourth component  $w$  in homogeneous coordinates, and be able to apply the perspective transform to a homogeneous coordinate vector.

## Implementation Issues

Understand the tradeoffs implied in allocating memory off the heap, using the various garbage collection schemes, etc.

Describe the garbage collection schemes we discussed (stop-and-copy, mark-and-sweep, reference counting, and generation), in particular, their various advantages and deficiencies when compared to each other, and to manual memory management.

Given a simple class structure, transform it into a data-centric design for better cache locality.

Understand the basics of the entity-component-system architecture, and when it might be preferable to a traditional object-oriented scheme.

Understand the importance of precise timing information for smooth animation.

## Networking and Multiplayer

Describe the two peer-to-peer and one client-server network topologies we discussed, and understand the advantages and disadvantages to each.

Describe how we could determine the latency (time delay) between two networked hosts, even though we do not have a single global clock to use as a reference.

Understand how an unreliable protocol like UDP affects the kinds of updates we can send over the network.

Understand the ways in which UDP is less reliable than TCP.

## Advanced Topics

Understand the kind of trivially-parallel problems to which GPU computation can be applied, and be able to give examples of problems that both are, and are not, suitable for GPU parallelization.

Understand the basics of Perlin noise: how it is built up from regularly distributed, pseudorandom values.

Understand the concept of *local* vs. *global* procedural generation methods, and be able to give examples of each.

Understand the basics of the A\* pathfinding algorithm: what kind of graph search it performs, and how the heuristic  $h()$  affects its accuracy and performance.

Understand the basics of *planning-based AI*, in particular, how it solves the problem of achieving some target goal.