You may bring two pages of notes, front and back.

Questions will be in short-answer format with partial credit for partial answers.

You will be asked to write pseudocode resembling Mono (the C# variant used by Unity).

You may assume that you have access to a vector math library that supports vector addition, vector subtraction, vector-scalar multiplication, and basic scalar functions (e.g., sqrt, pow, sin, cos, arccos, cot).

Topics:
- Vector-vector addition and subtraction, vector-scalar multiplication
- Dot product – definition, uses
- 3D sound - listeners, emitters, Doppler effect, occlusion/obstruction
- Digital Signal Processing (DSP) - uses for reverb, pitch shifting, compression, low-pass filtering
- Rays and line segments, uses for ray casting in games
- Sphere-sphere, AABB-AABB collision implementations, AABB from points, quadtrees
- Semi-Implicit Euler Integration equations (from forces to linear velocities to positions)
- State pattern – implementation, advantages
- Path nodes and navigation meshes, Euclidian, Manhattan distances
- Greedy best-first and A* pathfinding algorithms (not implementations)

1. [10] Describe three features in games that could be implemented using ray casting:

2. [10] Briefly describe a situation where it would be appropriate to use each of the following Digital Signal Processing (DSP) effects in a game: reverb, pitch shifting (other than Doppler), compression, low-pass filtering:

3. [15] Consider the following function:
void aabb(Vector3[] points, out Vector3 min, out Vector3 max);
Implement the above function so that it configures the min and max vertices defining an Axis Aligned Bounding Box (AABB) exactly encompassing the specified points:

4. [10] Is Manhattan distance more computationally efficient to compute than Euclidian distance? Why or why not?