Spring ’19 CIS 410/510 Midterm 1 Review

You may bring one page 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’re working with a vector-math library that supports vector addition, vector subtraction, vector-scalar multiplication (e.g., `Vector3 u = 2.0f * v + w`), a dot-product function `float dot(Vector3 a, Vector3 b)`, and a square-root function `float sqrt(float f)`:

Topics:
- Game loops and types of game objects (drawn, updated, both)
- Vector-vector addition and subtraction, vector-scalar multiplication, vector length
- Dot and cross products – definition (510 only for cross product), uses
- Vector normalization, linear interpolation
- Game mechanics, experience duration, ancillary rewards, practical rewards, difficulty, harnessed pacing & intensity, playtesting
- Commit discipline, 6 principals of Kanban
- Command, Flyweight, Observer, State, Subclass Sandbox, Spatial Partition patterns

1. [10] Consider the following function:

   ```csharp
   bool closerTo(Vector3 a, Vector3 b, Vector3 c);
   ```

   Implement the above function so that it efficiently (e.g., without using square roots) returns true if `a` is closer to `b` than `c`, false otherwise:

2. [10] Briefly describe 3 reasons why game difficulty should ideally scale superlinearly with player progression:

3. [10] Briefly describe the 6 practices of Kanban: