Spring ‘16 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 all problems use row vectors and a right-hand coordinate system (just like the textbook).

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).

410 students may assume that they have access the following functions (510 students may not):

```csharp
float dot(Vector3 u, Vector3 v);
Vector3 cross(Vector3 u, Vector3 v);
Vector3 reflect(Vector3 u, Vector3 v); // reflects u across v
```

Topics:

- Game loops and game objects
- Single and double buffering, tearing
- Sprites, sprite sheets, and the painter’s algorithm
- Vector-vector addition and subtraction, vector-scalar multiplication
- Dot and cross products, vector reflection, scalar projection
- Linear interpolation
- 1x4 vectors, 4x4 matrices, vector-matrix multiplication, matrix-matrix multiplication
- Rotation, translation, and scaling transformations, coordinate spaces
- Orthographic and perspective projections
- Ambient, diffuse, and specular lighting equations, z buffering

1. [10] Consider the following function:

```csharp
float angle(Vector3 u, Vector3 v);
```

Implement the above function so that it returns the angle between u and v in radians. You may assume that the vectors are normalized.

2. [10] Consider the following function:

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

Implement the above function so that it returns true if a, b, and c all lie on a single 3D line, false otherwise:

3. [10] Briefly describe the difference between single and double buffering and explain how double buffering avoids tearing: