# Computer Graphics - Assignment 2

CS 4830 — Dr. Mihail
Department of Mathematics Computer Science
Valdosta State University

February 4, 2019

## 1 Introduction

In this assignment, you will implement naïve matrix multiplication in an HTML document and practice your Javascript skills. The learning goals of this assignment are as follows:

1. Implement matrix-matrix, matrix-vector multiplication
2. Work with input/output in HTML/Javascript

You are given an incomplete HTML document, which you are to use to implement the matrix multiplication. This assignment consists of 2 parts: 1) matrix-matrix multiplication and 2) matrix-vector multiplication. The goal is to use incremental development, and write a function **multiply(A, B)** that returns the product of A and B, regardless of their shape, as long as they can be multiplied (i.e., they are comformable). You will use multiply() function in future assignments, so it is critical you finish this assignment.

## 2 Javascript

**Starting out** The start-up script for this assignment contains HTML definitions of 6 tables (3 for each part) with text inputs for each cell. The tables correspond to matrices. Each input box is an array element. The HTML inputs can be extracted from Javascript using the following syntax:

```
1 var foo = document.getElementById('tag_id').value;
2 var bar = parseFloat(foo);
```

In the code above, foo is a string data type, and bar is a float. You can begin this assignment by implementing functions *get_part1matrices_from_html()* and *get_part2matrices_from_html()*. They will both populate Javascript matrices $A$ and $B$. When the user clicks the compute buttons on either part 1 or 2, the data from HTML is collected into Js matrices A and B, then multiply() is called. Once multiply finishes, the HTML tables that stand for matrices C and F get populated. It is critical that the multiply() function works for both cases. You may only use if-statements for verifying if the matrices are comformable. The rest of the function doesn't need if statements.

**Javascript arrays** You have to use Javascript arrays for this assignment. The syntax to create an empty 1-dimensional array is as follows:

```
1 var X = [];
```

Notice how there is no pre-allocated size. Arrays are sized dynamically by assignment during runtime. 2-dimensional arrays are created from 1-dimensional arrays, where each element of the 1-d array is another array. The syntax for creating a 2-d array of size 4 by 4 with all zero elements is as follows:

```
var X = [];
for(var row = 0; row<4; row++) {
  X[row] = [];
  for(var col = 0; col<4; col++) X[row][col] = 0;
}
```

# 3 Program Requirements

- It is important that you only have multiply() function. You will receive no credit for this assignment if more than one multiply() function is used for the two parts, or if there are if-statements that trivialize the two cases. This function should work for all matrix multiplications, as long as they are comformable. The two test cases are shown here because they occur frequently in computer graphics.
- You cannot modify any of the function definitions, or create extra functions.

# 4 Hints

- Troubleshooting: most modern browsers have debugging tools built-in. One of the most useful ones is the console, which can be accessed on most browsers using the F12 key. You can output to the console using the following Js command:

```
console.log(variable);
```

  It is also in the console that syntax errors are shown by the Javascript engine. It is helpful to have that open as you develop.

- The length of an array is stored in the property *length*. For example, getting the first dimension (rows) of an array $x$ is done as follows:

```
var rows_x = x.length;
```

- When a variable is first declared, assigned (or declared and assigned symultaneously), it must be preceded by the var keyword.
- Please verify your solution using a software that can perform matrix multiplication, or by hand.

# 5 Due Date

This assignment is due before midnight on Thursday, February 7th.

# 6 Grading Rubric

- The get_part1matrices_from_html() and get_part2matrices_from_html() are correctly implemented using for loops. A series of assignments is not acceptable. This has to be done systematically using (nested) for loops. **50%**
- The multiply() function is correctly implemented and works for at least the both cases presented here. **50%**