Sprint 01 progress report

by Evan Arroyo and Joshua Sims

The goals set for this past sprint were to become more familiar with Unity, C#, and Visual Studio; to create the basic models for Pong; to get the user-controlled movement of the paddles working; and to learn how to implement persistent data storage for the game.

We became more accustomed to Unity’s UI as we followed several tutorials on Unity3D.com. The UI is, like that of many development programs, teeming with options which are unrelated to developing the basics for our game. However, since this is just the beginning of our project and because we are interested in using Unity as a future resource outside of this project, we do not mind the many advanced features scattered around the UI – we want to eventually become familiar with those features as well. With the help of Unity’s tutorials, we created rectangular paddles, a ball, the background (a flat plane), and barriers – the most essential models in Pong. We learned, generally, how to write code for the camera so that the camera angle is not static and can be repositioned. We also learned about the code necessary for the animation of the models and we have written code which allows the paddle to be moved side-to-side by the player.

In writing the code, we became more familiar with Visual Studio. Although, Visual Studio, like Unity, has many features which we do not yet understand. For the sake of writing maintainable, reliable code, we researched the C# coding conventions (as defined by Microsoft) as well as the creation and running of unit tests for C# code (also defined by Microsoft). We have not yet implemented any unit tests, but we plan to do so during the coming sprint and thereafter. We are confident that writing the code for this game will be a consistently productive process without frequent interruptions caused by the lack of familiarity with the language. We believe this to be true because we are familiar with Java which is very similar to C# -- thus, we are already able to write code for many common operations/tasks. Most of our C# research will be conducted for the purpose of learning C# data structures.

The final part of this past sprint was dedicated to researching the implementation of persistent data storage. Unity provides a long, thorough tutorial which explains and demonstrates how to perform data serialization using C# data structures and their respective methods. We have not yet incorporated persistence and we likely will not until the final couple of sprints (as defined by our project timeline). The purpose of researching persistence now was to understand our current and future work in that context and thus ease the work in the final couple of sprints in which we collect and visualize gameplay generated statistics.