Milestone Two: Software Design and Engineering Enhancement

What Is the Artifact?

The project I chose is a Unity-based craps game that I built during my earlier computer science courses. It’s a digital version of the popular casino dice game. Players can place bets using virtual chips and roll dice to win or lose money. The game includes features like bet tracking, chip values, dice rolling, and balance updates.

I created this project before taking CS 499, and I decided to use it as my artifact because it shows a full working program that uses design and coding skills.

Why I Included This in My ePortfolio

I chose this game for my ePortfolio because it shows what I can do with software design and programming. It includes real gameplay, logic, and a user interface. I also made changes to improve it, which helped make the code easier to understand and work with.

Here’s what makes the project strong:

It handles player actions well – like clicking chips or rolling the dice.

It uses object-oriented programming – I broke the big script into smaller parts (like ChipManager, BetManager, and DiceManager) to make the code cleaner.

It follows better design rules – like separating game logic from the visuals (user interface).

It’s easier to update now – new features can be added more easily because of the improved structure.

Before the update, the code was mostly all in one place and hard to manage. Now, the different parts have their own roles. The functions are smaller and named better, and the code is more organized overall.

Course Outcome I Focused On

The enhancement I made matches this learning goal from CS 499:

“Use proper tools and methods in computing to build software that solves real problems and adds value.”

By improving how the game is organized and how the different parts work together, I showed that I understand how to design and build software in a professional way. I didn’t change my learning goal for this milestone because I feel like the work I did still fits that outcome well.

What I Learned From Making the Enhancements

While improving the code, I learned a lot about why good design matters. At first, my game worked fine, but it was hard to change or fix because the code wasn’t well organized. After breaking the code into smaller pieces with clear roles, it became much easier to manage.

Here are some things I learned:

Bad structure causes problems later – keeping all the logic in one place made it hard to test or change.

Using separate scripts for different jobs (like handling bets or rolling dice) helped me understand object-oriented design better.

Fixing the design now saves time later – future updates will be faster and easier to make.

Some of the challenges I ran into were:

Figuring out how to separate the code without breaking the game.

Making sure all the different parts could still talk to each other properly.

Learning more about Unity’s way of handling scripts and events.

In the end, this project helped me practice real software development skills. I feel more confident now in designing and improving code. I’ll include this in my ePortfolio to show future employers what I can do.