Knight’s Party Table Software Architecture Document

Version 1.0

Revision History

Date	Version	Description
11/6/2019	1.0	First Draft of the Software Architecture Document

Table of Contents

1. Introduction

1.1 Purpose

This document provides an architectural overview of the Knight's Party Gaming Table at Vermont Technical College. This document is intended to capture and convey the significant architectural decisions which have been made in designing and building the system.

1.2 Scope

The scope of this document is the implementation of the Knight's Party Table.

1.4 Overview

This document consists of six sections, described by the following:

- Section 1 is an introduction to the software architecture of the Knight's Party Table
- Section 2 is an overview of the constraints and goals of the system architecture
- Section 3 contains the architectural representation of the system
- Section 4 contains the decomposition of the system's architectural views
- Section 5 is an overview of other system considerations
- Section 6 describes quality issues in regards to the system

2. Architectural Goals and Constraints

The Knight's Party Table architecture has been designed with the following objectives in mind:

1. To successfully allow the players to play up to five unique games
2. To function, as designed, at least 99.99% of the time

The major design and implementation constraints for the system are:

1. Simplicity
2. Reliability
3. Debugability

A full list of system requirements can be found in the Knight's Party Table Requirements Document.

3. Architectural Representations

3.1 Architectural Views

Use Case View: the purpose of this view is to outline the major system requirements to be implemented.

Module View: this view shows the general modules of the system and how they will interact with each other.

Component View: the purpose of this view is to outline the different components that the system is comprised of.

Deployment View: this view shows exclusively the physical nodes that will be present when deploying the system.

3.2 Architectural Design Patterns

The architectural design pattern used to create the Knight's Party Table is a Layered Architecture. This design was chosen because while it is not a universal design pattern, we found it would best suit the Knight's Party Table. By establishing separate layers in the system's architecture, we can effectively ensure simplicity and reliability in the Knight's Party Table.

3.3 Architectural Style

The architectural style for the Knight's Party Table architecture:

Presentation Tier
Interaction Tier
Data Tier

Presentation Tier: Used to format and present information to the user

Interaction Tier: Used to interact with the data and game files.

Data Tier: In charge of storing data for the various games.

3.4 Architectural Process

The Knight's Party Table will follow the Scrum method of software development. Scrum utilizes set events to create regularity during development. The Scrum events include the Sprint, Sprint Planning, Daily (or in our case, Weekly), Sprint Review, and Sprint Retrospective.

During the Sprint, code will be written for each task. Sprints will last up to two weeks, and during that time a task will be completed. After that task is completed, it will be debugged and reviewed before another task is chosen and worked on, during the Sprint Review. After the project is complete, we will hold our Sprint Retrospective.

The following chart is the project Gantt Chart.

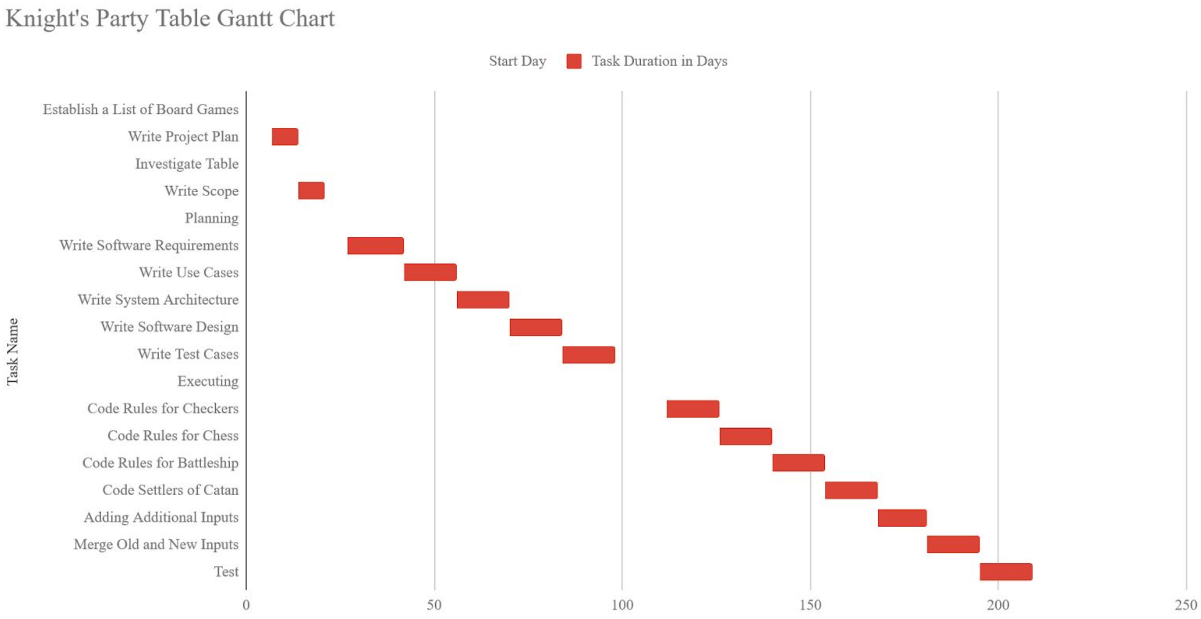

Figure I: The Gantt Chart

4. Architectural View Decomposition

4.1 Deployment View

The deployment view of the system shows the physical nodes that will be interacting with each other in the on-location system. The only physical aspect of the system is the Knight's Party Table itself, which is made up of the central PC and two Controller Units that the user utilizes to interact with the system.

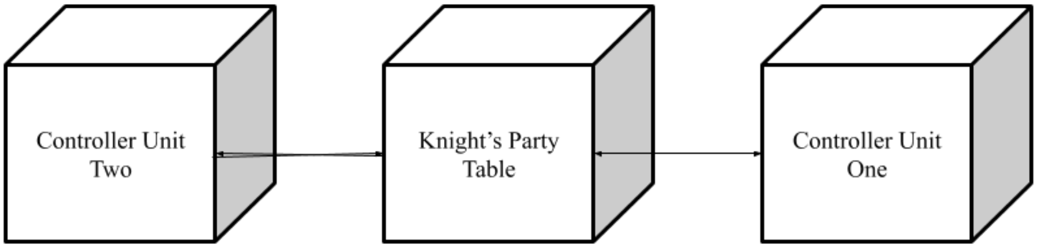

Figure II: The Deployment View

4.2 Use-Case View

The general functionality of the Knight's Party Table can be seen in the following diagram:

Start/Select a Game

The user will be able to select a game from the list presented to them and have that game properly boot, allowing them to play.

Play a Game

The user will play the game they selected.

Take a Turn

The user will take their turn in the game they are playing. This varies from game to game. For example, a turn in both Chess and Checkers involves selecting and moving a piece, but in Settlers of Catan it involves building settlements.

Finish a Game

The user will successfully finish a game. This varies from game to game. For example, in Settlers of Catan, it requires a player to collect ten Victory Points. In Chess, a player must place the other player in checkmate, in Checkers all of the other player's pieces must be captured, and in Battleship all of the other player's ships must be sunk.

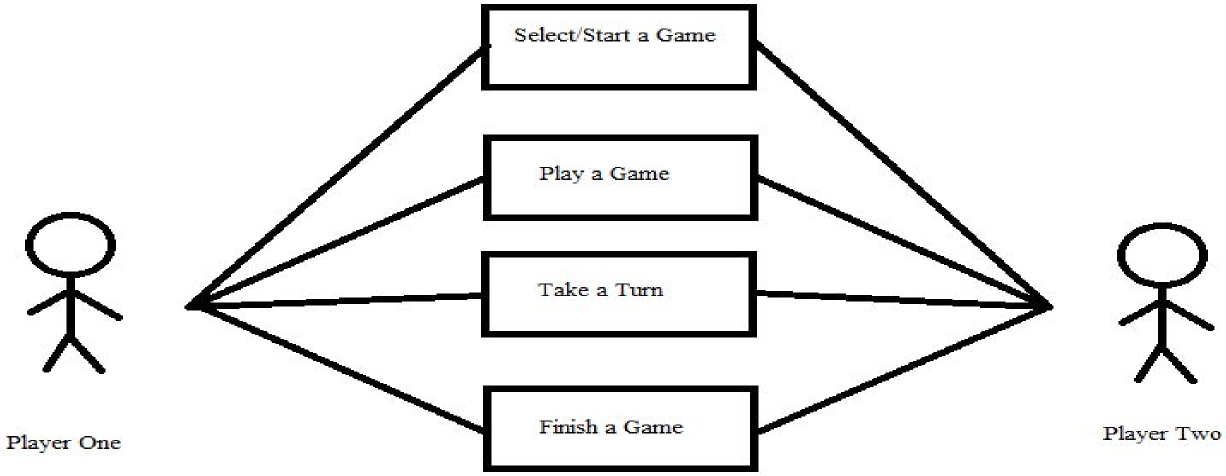

Figure III: The Use-Case View

4.3 Module View

The module view is meant to show the different modules the system is comprised of through the different layers of interaction. The Knight's Party Table is comprised of two different modules - the User Interface and the Games Library. The User Interface is how the user interacts with the system. The Games Library is where all of the games are stored and launched, by using the User Interface.

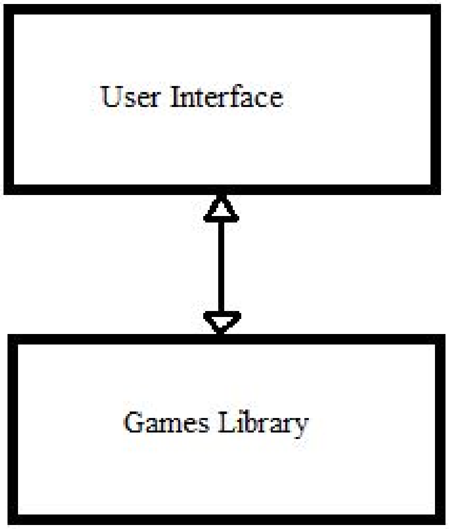

Figure IV: The Module View

4.4 Component View

The component view describes the different systems and subsystems of the Knight's Party Table. The user interacts with the Knight's Party Table via the Controller Units, making selections in the User Interface and playing various games. The games are accessed through the Games Library via the User Interface. The games are played via input from the User Interface directly to the games.

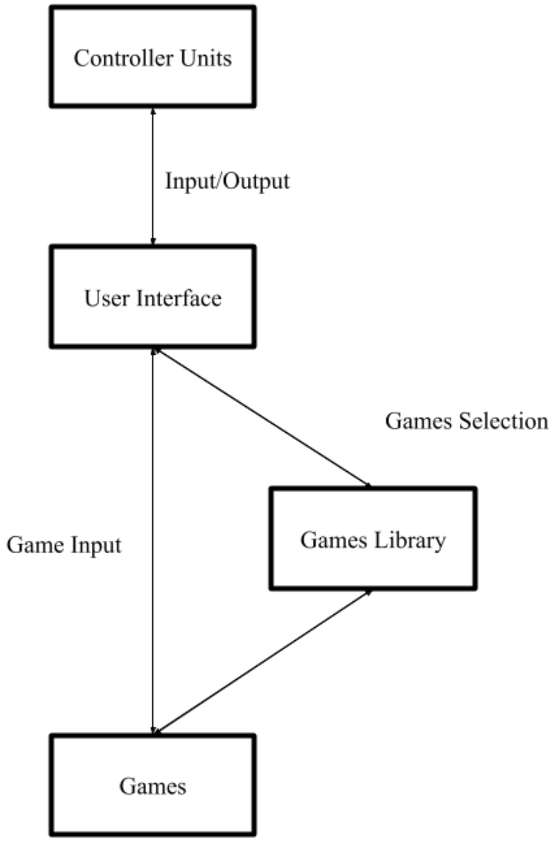

Figure V: The Component-and-Connector View

5. Size and Performance

The Knight's Party Table requires the following:

- Number of Use Cases to be implemented: 14
- Number of Servers Needed: None
- Number of Players Accessible: Two
- Number of Games Playable: Five

The Knight's Party Table will allow up to two players to play a game simultaneously, and display public information on a third monitor to the public.

The Knight's Party Table will not store high score data, unless the project is completed ahead of schedule.

The Knight's Party Table will take less than one minute to load a game when a player launches it.

6. Quality

This section will define important quality standards the system will follow:

1. The system application will meet the requirements specified.
2. The system will be 99.999% reliable.
3. The system will be accessible, as defined by the following:
    a. Any user with common understanding of the board game will be able to successfully play a game with little outside instruction.
    b. Any user with a basic understanding of how to utilize a personal computer will be able to successfully navigate the system's menus.
4. The system will be maintainable, and any group that further "improves" our work will be able to do so with little strife.
    a. The system's code will be well documented to ensure a fellow programmer can understand the functions of each module.