Project Plan and Preliminary Schedule

For CEG-4120, Spring 2021, Group 4

By

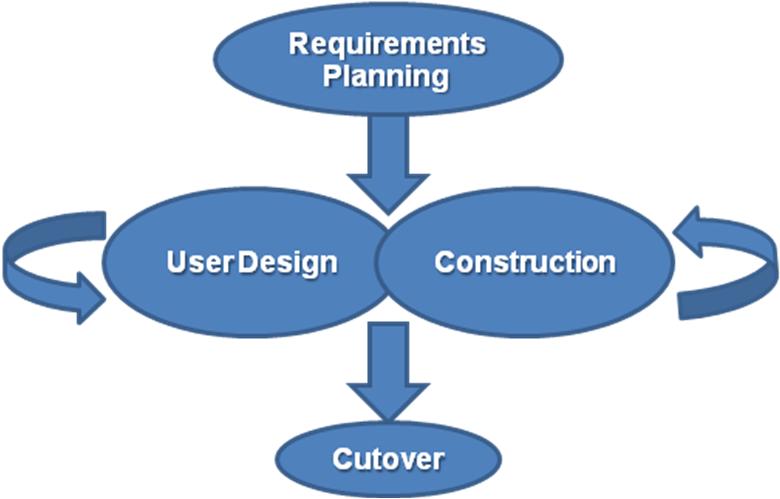
Katie Miller

David Wilson

Reid Wynkoop

# Project Plan

For the course project in CEG 4120, “Managing the Software Development Process,” we will build a program containing fully functional versions of the classic board games *Scrabble* and *Monopoly.* We plan to accomplish this by roughly following the “James Martin” Rapid Application Development (RAD) software lifecycle model (SLM) depicted in Figure 1.



**Figure 1: Depiction of the James Martin RAD model from** [**Wikipedia**](https://en.wikipedia.org/wiki/Rapid_application_development)**.**

The process for this SLM begins with the “Requirements Planning” phase, which actually encompasses the full range of activities usually associated with the Requirements, Analysis, and Design phases of the traditional waterfall method. From there, we enter a cyclical process of “User Design” and “Construction” phases, which incorporate designing, implementing, and basic testing. This cycle will see us first develop a foundational core for our program and then iterate upward to a feature-complete product. The model concludes with an exit to the “Cutover” phase, which includes final acceptance testing and submission of the course product.

At the time of this writing, we plan to implement the game with Unity, a free game engine that will allow us to focus on the user interface objects and script behaviors with the C# language. We plan to use preconstructed assets and images for all game pieces, cards, or tiles as needed. We selected these tools and resources to expedite the development of the project in the short window provided by the course.

# Preliminary Schedule

We have divided the overall course project into the six milestones presented in Table 1. The first two represent the Requirements, Analysis, and Design activities that go into the “Requirements Planning” phase of our SLM. The third through fifth milestones represent the cycle of iterative “User Design” and “Construction” phases. The sixth milestone represents the “Cutoff” phase.

**Table 1: Division of Project Milestones and Accompanying Schedule.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Milestone** | **Duration** | **Start Date** | **End Date** |
| 1 | Initial Requirements | 1 Week | January 18th | January 24th |
| 2 | Analysis and Design | 1 Week | January 25th | January 31st |
| 3 | Game Foundations | 2 Weeks | February 1st | February 14th |
| 4 | Scrabble Completion | 3 Weeks | February 15th | March 7th |
| 5 | Monopoly Completion | 5 Weeks | March 8th | April 11th |
| 6 | Final Testing and Submission | 1 Week | April 12th | April 19th |

The general expectations for each milestone are roughly as follows:

1. Initial Requirements
   1. Select tools and languages to be used for development.
   2. Create individual accounts for each developer to use as needed.
   3. Set up workspace for all group members.
   4. Define specific requirements for each of the remaining milestones.
2. Analysis and Design
   1. Write a list of all user interface object “classes.”
   2. Assign a list of attributes and behaviors to each class as needed.
   3. Design a pair of flow charts describing the normal turn routine in each game.
3. Game Foundations
   1. Create the program’s main menu, escape menu, and basic menu functions.
   2. Create the score card for each game and use it to control the turn order.
   3. Create a basic collection of visual pieces for each game, including the boards, game pieces and dice for Monopoly, and tiles for Scrabble.
   4. Ensure that players can place tiles on the Scrabble board.
   5. Ensure that players can roll the dice and move their piece around the Monopoly board.
   6. Implement elementary local multiplayer and functionality for passing turns and quitting.
   7. Test existing features.
4. Scrabble Completion
   1. Implement complete scoring for both tiles and special word squares.
   2. Implement score tracking and the game completion sequence.
   3. Implement automatic validation based on tile placement and word integrity.
   4. Implement online multiplayer.
   5. Test new features.
5. Monopoly Completion
   1. Implement money, deeds, cards, property improvements, and special squares.
   2. Implement purchasing, selling, renting, mortgaging, and improving mechanics.
   3. Implement special square mechanics such as “Income Tax” and “Jail.”
   4. Implement money tracking, bankruptcy, and the game completion sequence.
   5. Implement online multiplayer.
   6. Test new features.
6. Final Testing and Submission
   1. Perform final acceptance testing.
   2. Clean up remaining issues with visuals or behavior.
   3. Submit the project.