Project4

# Project Overview

This project aims to build a website that enables friends and family members to share video games.

Video games provide a fun and engaging experience for those that play them. Unfortunately, most games have an end. Once all the game’s quests or battles have been completed, there is no more content to experience within the game. However, even after you are finished with the game, you still might want to keep the game because it was so much fun to play. Maybe you will go back to playing it in a few months or years. In between play throughs of your favorite games, why not let someone else borrow them from you?

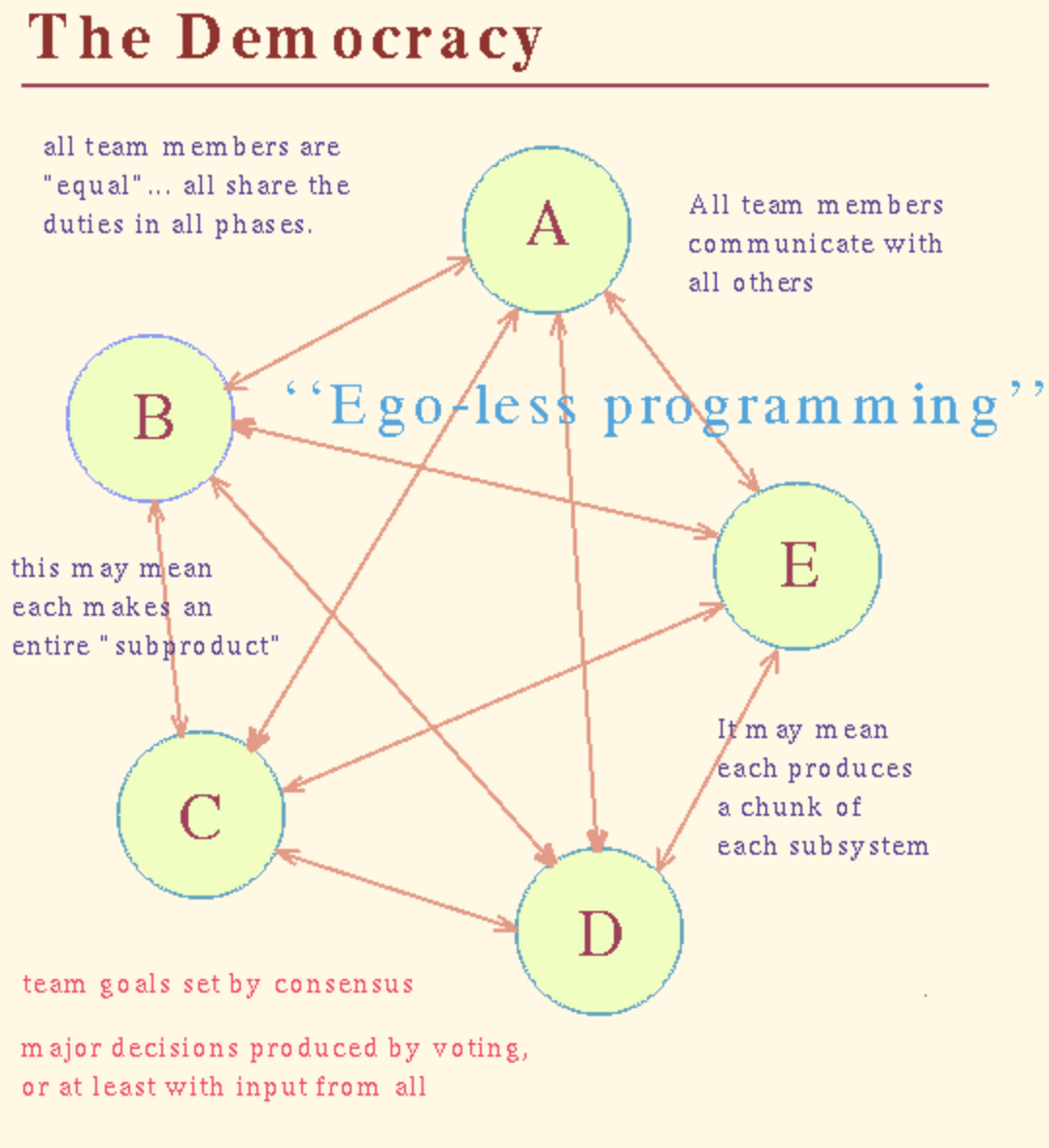
From the website, users will be able to browse the assortment of games, filtered by console, to find the right game. Games will be searchable by categories, so users can find a game with a specific number of players or a game that can be completed quickly. Users will have the ability to upload their own games to the database when they want to lend some out. The user will also be able to remove games if they no longer want to loan it to others. The typical user should also have the ability to send invites to their friends and family via e-mail.

When a game has been lent out, the owner of the game will update its status on the website. The owner could even have the option to enable a waiting list for his/her game. This would help owners keep track of the people waiting to borrow their games. The website would have features that allow the user to check games in and out quickly, as we all know how precious time is.

As we want the database for this site to continually grow among friends and relatives, we will have a feature that enables users to request new or updated features.

# Team Organization

Our group will be organized democratically. This form of organization is also known as the ego-less team. Below is a diagram representing the organization that will be used.



With the formation of the Democratic Team we will be able to work on the project together. The communication in our team will be transparent and allow everybody to work on task together. The majority of the work will be assigned to individuals, but will be completed as a team effort. This way everybody will be able to work on the task they have the most experience and assist others where needed.

# Software Development Process

The development process will be broken up into five phases. Each phase will resemble and take features from Sprints in the Agile method and iterations in the Spiral process. Specifically, each phase will be like a Sprint in that work to be done will be organized into small tasks, placed into a “backlog”, and prioritized. Then, using on time-box scheduling, the team will decide which tasks the phase (Sprint) will address. The team will use a Scrum Board to keep track of which tasks are in the backlog, added to the current Sprint, in progress, and done.

Each phase will also be a little like an iteration in a Spiral process, in that each phase will include some risk analysis and that any development activity (requirements capture, analysis, design, implementation, etc.) can be done during any phase. Early phases will focus on understanding (requirements capture and analysis) and subsequent phases will focus on design and implementation. Each phase will include a retrospective.

|  |  |
| --- | --- |
| **Phase** | **Iteration** |
| 1. | Phase 1 - Requirements Capture |
| 2. | Phase 2 - Analysis |
| 3 | Phase 3 - Architectural, UI, and DB Design |
| 4 | Phase 4 - Detailed Design, Implementation, and Unit Testing |
| 5 | Phase 5 - More Implementation and Testing |

We will use Unified Modeling Language (UML) to document user goals, structural concepts, component interactions, and behaviors.

# Communication policies, procedures, and tools

The main communication method among members will take place in the group’s Discord server. The server can be expanded to accommodate one-on-one communication as well as group discussions. Private messages can also be sent over Discord in order to keep the general channel for the project free of off-topic clutter.

Procedures

* Sprint planning following the scrum framework
* Ego-less workflow where task oversight is done by everyone
* Documentation Stored in Overview > Sub-App directories

Tools

* UML: [draw.io](http://draw.io/)
* Database: PostgreSQL
* Server: Pi based storage.
* Programming Languages / Frameworks: Node.js, Jquery, HTML, CSS, PSQL, pgAdmin, PLSQL, Vue.js

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk Analysis** | | | | |
|  |  |  |  |  |
| Identify your risks by area and then estimate their probability and impact | | | | |
|  |  |  |  |  |
| **Risk / Mitigation Activity** |  | **Probability** | **Impact** | **Estimate (additional hours if risks occurs)** |
|  |  |  |  |  |
| **R1** | **Co-worker illness** | **Medium/High** | **Medium/High** |  |
| 1.1 | Blood clots / Allow recovery | High | High | 20 |
| 1.2 | Common Cold / Allow recovery | Medium | Medium | 5 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| **R2** | **System error** | **Medium/High** | **Medium/High** |  |
| 1.1 | File Corruption / More commits | Medium | Medium/High | 10 |
| 1.2 | Computer Crash / Buy new computer | Medium | Medium | 2 |
|  |  |  |  |  |
|  |  |  |  |  |
|  | (add more areas as need) |  |  |  |
|  |  |  |  |  |
| **Total** |  |  |  | **37** |

# Configuration Management

See the README.md in the Git repository.