Project Proposal and Planning

Binary Bards: Unleashed

## Project Title & Description

**Title:** Binary Bards: Unleashed

**Description:**

This project is a 2D turn-based role-playing game (RPG) developed in C# using sprite-based graphics and a custom graphical user interface (GUI). Players control a party of characters, navigating through various levels and engaging in strategic, turn-based combat. Combat actions—such as attacking, defending, skills, and using items—are selected through the GUI and executed with corresponding sprite animations. The game incorporates health, skills, and magic that must be used to overcome enemies and climb the ***Dungeon Tower***. This project demonstrates the use of object-oriented programming principles, game loops, turn handling, and sprite animation within the C# environment using *Windows Forms App (.NET Framework)* through Visual Studio or *Unity*.

## Problem Identification & Research

### Existing Similar Games

#### Pokémon Emerald

* **Strengths:** Diverse character selection and movesets. Types are effective against some characters and not effective against others, which promotes strategy.
* **Weaknesses:** The type chart is very complex and therefore might confuse players and take away from the fun
* We will tone down on the amount of gameplay options the player has, and offer a more streamline gameplay experience
* We will also incorporate randoms so that no 2 runs are the same

#### Persona 3: Reloaded

* **Strengths:** Persona 3 has a system called “Personas”, which are like modular weapons. They are characters that are bound to your actual human characters, and have different skills, abilities, etc.
* **Weaknesses:** System feels too complex at times, and the gameplay is so complicated that it takes away from the fun
* Similarly, we will connect abilities/skills to a single character type, but the abilities will have the same impact but with less diversity
* We will also incorporate stats for the actual characters themselves as opposed to a secondary space like a weapon, etc.

#### Pokémon Mystery Dungeon

* **Strengths:** In addition to diverse characters and movesets, it also has a randomly generated dungeon with different floor layouts every time. It is also similar to a roguelike
* **Weaknesses:** Top-down dungeon crawler, so the sprites and assets aren’t super high quality
* We will have a similar procedurally generated enemy layout and map
* Our game is not a dungeon crawler; therefore, we will not have a procedurally generated map *layout*, just the map itself

#### Darkest Dungeons

* **Strengths:** It is light on asset animations, uses a simple system to generate complex dungeons, and has good emergent gameplay loops.
* **Weaknesses:** The game can feel too uninvolved as there is no hard-written storyline, and there are very harsh punishments that could discourage players from continuing to play.
* **Improvements:** Binary Bards: Unleashed will feature a more simplistic style of gameplay that takes inspiration from the light animation styles and will avoid giving players intense debuffs and status effects to encourage replay value.

#### Hearthstone

* **Strengths:** It is a card-based game strategy game that features a wide variety of cards, abilities, and deck customization. The balanced turn-taking makes it easily accessible to the majority of players and makes it rewarding to play.
* **Weaknesses:** The UI and card features can quickly become frustrating for new players to learn. Frequent updates and expansions can also create a pay-to-win environment which can deter players from continuing.
* **Improvements:** Binary Bards: Unleashed will feature a simple UI system that is built to be intuitive and will feature a fair but challenging progression system throughout the Dungeon Tower.

## Expected Features & Functionality

* Turn-Based Gameplay
* 4 Player Party made of different classes with different abilities
  + Cleric
    - **Weapon:** Harp
    - **Skill:** Healing
  + Sorcerer
    - **Weapon:** Lute
    - **Skill:** Area of Affect Spell
  + Troubadour (Rogue)
    - **Weapon:** 2 Cello Bows
    - **Skill:** Stealth (Smoke Screen)
  + Chellist (Tank)
    - **Weapon:** A cello without a bow
    - **Skill:** Defend (Can block all damage to a teammate)
* Group inventory of Consumables and Passive Items that help the whole party
* Scaling difficulty
  + One Boss every ten floors
  + After Boss is Defeated, prompted a question “Continue or Leave the Dungeon?”
* 5 Fight options
  + Attack (Basic attack, uses attack value of the character with no multiplier)
  + Skill (One class-based skill per character turn)
  + Item (Allows the use of consumables during battle)
  + Guard (Reduces damage taken by a multiplier of defense)
  + Stats (Able to see characters' health points, attack, defense, and buffs/debuffs. This option doesn’t progress the fight)
* Changing backgrounds and enemies
* Leveling system that increases stats based on experience gained from enemies

## Potential Features:

* Status Effects
  + Burn
  + Bleed
  + Frightened

## Technologies to be Used

|  |  |
| --- | --- |
| Types | Technologies Used |
| IDEs: | * Visual Studio Code * Visual Studio |
| Libraries: | * .NET Framework * Unity |
| Version Controller: | * GitHub |
| Team Communication: | * Discord * Microsoft Teams |
| File Sharing | * GitHub * OneDrive * Unity |
| Assets | * Unity Asset Store * Itch.io |

## Team Member Roles & Responsibilities

|  |  |  |
| --- | --- | --- |
| Role | Member | Responsibilities |
| Project Manager | Shabhan George | Project Timeline, deadlines, communication, and project definition. |
| Lead QA Tester | Michael Gremse | Test plans/cases, ensure functionality of features, and documentation of bug testing. |
| UI/UX Designer | Shabhan George | App presentation, visual designs, program intuitiveness, user-friendly features, and the main user loop. |
| Documentation Specialist | Jason Edgington | Documenting program features, recording the design process, ensuring accurate version control logs, and submitting documents alongside the project. |
| Lead Developer | Zachary A. Carmichael | Programming of the main skeleton, functionality of complex features, program optimization, and overseeing contributions to the code base. |

## Project Timeline

