

Final Project Proposal
Fire Emblem Remastered

Our project is a turn-based tactical role-playing battle simulator based on the game Fire Emblem: Awakening. We would have a split display such that the terrain would be displayed on the top part of the screen while the current unit's information is displayed on the bottom part of the screen. Before each wave, the player gets to select a limited number of characters from his list of characters to use in the battle. During the battle, the player has to control each unit manually. The movement of each character is dictated by a tile-based grid-like system and during your turn you will have to move your characters to a specific tile. During combat, the enemy units will be controlled by a form of artificial intelligence where the enemy will position its units and/or attack your units. During battle, each character gains experience points by performing actions, such as attacking an enemy, killing an enemy or healing an ally.

During battles, once a certain experience is reached, the character levels up and new skill points are randomly awarded to a character's attributes such as maximum health, speed, strength. The more you use a character in battle, the more experience that character gains. A character's class affects both their attack capabilities and their mobility on the battlefield, for example, mages and archers can attack at greater ranges than melee units but mounted units have more powerful attacks and have greater mobility. Another element of the game is the "weapon triangle" which works like rock, paper, scissors. This means that lances are stronger than swords, swords are stronger than axes and axes are stronger than lances. The game will also use a weapon durability system which will be governed by a heap so that when your weapon breaks in battle your next equipped weapon will be determined by the next highest ranked weapon in the heap.

The order of which enemy will attack will be determined by a priority queue which will assign the order of the enemy units turns based on attributes such as level, class, health, etc. The enemy's turn will be governed by an algorithm that will utilize a counter to check the shortest distance to the closest ally unit utilizing a distance function and quadrant movement to minimize system runtime. In order to beat the game, the player has to reach the final round of the game and defeat the boss. Once this stage of the game is reached, the player is given the option to restart the game or continue and level up the rest of his characters. During any stage of the game the player may revisit one of the waves in order to level up his characters or obtain currency/items. There will be an ingame shop so that the player can buy weapons when his break or to buy potions to use during the next battle. We might incorporate CSV game data storage depending on how much time we have after completing the game so that the player can continue where he/she left off after ending off the game. We will also be using the perlin noise processing function to generate terrain based off of gradients to make it visually appealing.

CHARACTER FEATURES

- In the beginning of the game, the user will be instructed to play a tutorial
- Afterwards, the user will be asked to select 5 characters
- Depending on the progression of the user, the user can select n characters to bring with them to fight the enemies
- Selection may include depending on unlocked characters:
 - Lord, Tactician, Cavalier, Knight, Myrmidon, Thief, Fighter, Mercenary, Archers, Mage, Healer, Troubadour, Pegasus Knight, Wyvern Knight
- Character Statistics
 - LVL
 - Level of your character
 - EXP
 - How much experience character has until it levels up
 - HP
 - Hit points of character
 - Atk
 - How much damage character can do
 - Hit
 - Chance of hitting a target
 - Crit
 - Chance of getting a critical hit
 - Avo
 - Chance of avoiding an attack
 - Str
 - Adds to physical damage
 - Mag
 - Adds to magic damage
 - Skill
 - Adds to critical hit and accuracy
 - Spd
 - Helps you attack twice, also raises your evasion to avoid attacks as well.
 - Lck
 - Higher chance of getting a critical hit, and it also lowers the opponent's chances of getting a critical hit.
 - Def
 - Adds to defense
 - Res
 - Ability to guard better against magical attacks

- Mv
 - Amount of steps character can take in one turn
- Select weapons
 - Weapons will be used in a particular order using a modifiable queue
 - Once a weapon breaks, the character will automatically be given the next weapon in the queue
- Select items
- Amount of currency

ENEMY FEATURES

- All enemies under one class
 - As the user progresses through the stage, he or she will experience more enemies at higher stats compared to the previous level
 - Enemy can only move and attack, cannot use items
 - Enemy can have items in their inventory that will be dropped upon death
 - Specified spawn areas, and enemies move out from these spawn points and attack the player
 - Random enemies will have certain amount of currency on them that will be dropped upon death, and the currency then can be used to buy characters in the shop

MAP FEATURES

- Map generator using perlin noise function found in processing
- Terrain will have certain areas that increase stats

BATTLE FEATURES

- [Calculations](#)
- Priority queue for which enemy attacks first
 - Will be based on lvl, class, health, etc.
- Weapon triangle
 - Sword beats Axe
 - Axe beats Lance
 - Lance beats Sword
- Combat forecasts
 - A chart of the outcome of an attack will be visible before each attack

- Battle properties are determined by different formulas
 - $\text{Attack} = \text{Strength or Magic} + \text{Weapon's Might} + \text{Weapon Rank bonus}$
 - $\text{Hit rate} = \text{Weapon's Hit rate} + [(\text{Skill} \times 3 + \text{Luck}) / 2] + \text{Weapon Rank bonus}$
 - $\text{Critical} = \text{Weapon's Critical} + (\text{Skill} / 2)$
 - $\text{Avoid} = (\text{Speed} \times 3 + \text{Luck}) / 2$
 - $\text{Rating} = \text{Strength} + \text{Magic} + \text{Skill} + \text{Speed} + \text{Luck} + \text{Defence} + \text{Resistance}$

INCORPORATED TERM 2 CONCEPTS

- Heap for weapons
 - Weapon durability system so that when your weapon breaks in battle your next equipped weapon will be determined by the next highest ranked weapon in heap
 - (1)sword <- (2)axe <- (3)sword2 <- potion <- (4)axe2
 - 1 -> 4 weapon usage
- Priority queue for which enemy attacks first
 - The order of which enemy will attack will be determined by a priority queue which will assign the order of the enemy units turns based on attributes such as level, class, health, etc.
- Doubly Linked list for the waves
 - Credit to Mansour for the suggestion
 - Waves are organized inside doubly linked lists so that the current node is the current wave and if you wish to access a previous wave then you can go back a couple of nodes to access the wave and replay it.

EXTRA (POTENTIAL) CONCEPTS

- Save the player's data onto a CSV file found in the directory so that the player can start wherever he left off
 - Would need a login system so that a player cannot access another's data
- Ability to pair up units so to allow a chance to dual guard and dual strike
- Include a battle scene animation

MINIMUM VIABLE PRODUCT (MVP)

- Game is capable of loading start menu and selecting the 1st wave
- Game is able to play 1st wave and generate 1 map using perlin noise, spawn 1 character and 1 enemy

- Both character and enemy are able to attack each other and once either dies the game ends