Brent Knopp

Champion Document

**1.  Brief Introduction:**

My feature is titled Combat/Overworld Event/Logic. It will handle every aspect of the unit’s action event after an individual unit is moved. Each player will have multiple units in a game, where one individual unit will be moved on the game board per each turn. After the unit is moved this feature will be invoked. The scope of the feature will consist of deciding and handling a player’s action event for a unit, which will include selecting the type of action event and calculating and updating damage occurring on each action event. The base game will consist of two action events: battle and do-nothing. The battle event is an attack on an enemy, while the do-nothing event is used as a defensive maneuver or to close distance before an attack can be performed. As the game expands in development, more action events can be added to the action event superclass. For example, we might include the option to buy or trade game items used in battle to increase a unit’s weapon attribute for battle.

The battle action event will mainly handle the unit’s attack with an emphasis on awarding the correct damage to each unit, per attack. This will include all animation graphics involved in the attack. The unit will first suffer damage from the type of terrain that the unit has moved to. Then it will handle the player targeting an enemy unit to attack with a varying attack form and check for the correct distance requirements to proceed. Then, the attack will commence, and each unit will be allowed to do one damage sequence followed by an animated battle scene that will be displayed for the battle. The battle event will calculate the damage of each attack based on the unit’s experience attributes and weapon attributes used, followed by an update of the overall health of each unit involved in the attack. There will be experience points awarded from a successful battle attack that will make the unit’s attributes stronger for future attacks. Health will then be checked to decide if any of the involved units are dead and will then be immediately eliminated from the game with an animated death scene.

The end of this feature will coincide with the end of the player's turn. The do-nothing event is a basic event that will only check and update damage from terrain movement. There will be no battle in this type of action, but the unit must adhere to terrain damage with each move. The do-nothing event will check the health of the unit to see if it has died, followed by a death animation graphic. The form of action will be used as a defensive strategy or when a unit needs to cover a large amount of terrain before it can attack.

**2. Use Case Diagram with Scenario**

### Scenario

### Name: Event Action

**Summary:** The event action will handle the sequence and the type of action on a unit’s turn. This event will begin after the unit moves to a new position on the board and ceases when the unit’s turn is over. This event will consist of selecting an offensive/defensive attack and dealing the correct damage to both unit’s involved in the event on a player’s turn. The player’s turn will include one event action per turn. The event will include all graphics scenes related to the event.

**Actor:** Player Unit

**Secondary Actor:** Enemy Unit

**Preconditions:** Player’s turn and has moved a unit to new space on map.

**Basic sequence:**

**Step 1:** Deals Damage to the current terrain moved to by the unit.

**Step 2:** Check health and update units health and death.

**Step 3:** Selects the type of battle action and where on the grid for the battle action.

**Step 4:** Selects the type of weapon to be used in the battle action.

**Step 5:** Determine if the range is correct for the battle event.

**Step 6:** If invalid event, go back to Step 3.

**Step 7:** Initiate Event and Calculate Battle damage for both units.

**Step 6:** Play graphics of the battle event.

**Step 7:** Display results

**Step 8:** Check health and update units health and death.

**Exceptions:**

**Step 1:** Invalid battle event selected, go back to step 2.

**Step 2:** Invalid control buttons for selecting event, do nothing and wait for correct button input.

**Post conditions:** Update both units’ health and death, Return control of game’s turn to the enemy player.

**Priority:** Overall this feature is a priority \*1 and must be included in the game, it is the backbone of how the game is played. However, we will only need 1 type of battle event and 1 type of weapon to get a basic game running. Including different types of battle events and weapons are priority 3\*. Increasing the number of battle events and weapons would make the game more interesting and increase user’s attention. The graphic of the battle scene is not needed and are of priority 3\*. The results of each battle event need to be displayed to the user and are priority 2\*.

**ID:** C01

\*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

**4. Acceptance Test**

The inputs for this feature are the players game play event decision, unit position on terrain tile and the corresponding units’ attribute sets that are involved in the game play decision. A typical battle decision will involve both the player’s unit attributes, and the enemy’s unit attributes. However, when no battle occurs it will only involve the player’s unit since the enemy is not involved. The output of this feature is an updated units’ attribute set that will be changed directly by an event algorithm that is dependent on these four inputs: event decision, terrain tile, player’s attributes, and enemy’s attributes. This algorithm will update the health, weapon and experience attributes set based on a randomized battle algorithm giving preference to the unit with higher attributes. Thus, it will be responsible for determining when units are eliminated and destroyed from the game. The acceptance test for this feature will be a test that confirms that the proper units’ attribute calculations were done currently, and the correct values were updated in the game resulting in the correct game piece being destroyed.

A base-line acceptance test would consist of creating randomized test vectors, which will include one player’s attributes, 1 enemy’s unit attributes, 1 type of event selection, and 1 type of terrain tile. The test will need to check all the damage calculations, check all experience calculations, and determine all deaths correctly happen to both units within each event. The tests will need to be done with all variations of health and check invalid instances with negative health or attack abilities. These tests should also include every attack spot and confirm that each battle event cannot be performed out of range or on non-unit map position. These invalid attacks need to be handled correctly. Every battle conclusion needs to be confirmed that the correct animation has occurred, control has transferred to the enemy and only one battle event can be performed per turn. The tests will need to confirm that the range of the unit attacking can only fight with the correct weapon range. The do-nothing action will need to be tested to confirm that no enemy attributes are affected with this selection and the terrain damage is included in the event.

**5. Timeline**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Work Items** | |  |  |  |
|  |  |  |  |  |
|  | **Task** | **Duration (PWKs)** | **Predecessor Task(s)** |  |
|  | 1. Feature Requirements | 5 | --- |  |
|  | 2. Create Event Selection User Interface | 8 | 1 |  |
|  | 3. Design attribute structure | 5 | 2 |  |
|  | 4. Logic for Battles | 7 | 2 |  |
|  | 5. Update Results of Event | 7 | 3, 4 |  |
|  | 6. Game Graphics of Battle | 10 | 1 |  |
|  | 7. Feature Testing | 7 | 5, 6 |  |
|  | 8. Installation of feature to Game | 4 | 7 |  |

**Pert Chart**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | 13 | 5 | 18 |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | 3 | | |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | 15 | 2 | 20 |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 5 | 8 | 13 |  |  |  |  |  | 20 | 7 | 27 |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 2 | | |  |  |  |  |  | 5 | | |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 5 | 0 | 13 |  |  |  |  |  | 20 | 0 | 27 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | 13 | 7 | 20 |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | 4 | | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0 | 5 | 5 |  |  |  |  |  | 13 | 0 | 20 |  |  |  |  |  | 27 | 7 | 34 |  | 34 | 4 | 38 |  |
| 1 | | |  |  |  |  |  |  |  |  |  |  |  |  |  | 7 | | |  | 8 | | |  |
| 0 | 0 | 5 |  |  |  |  |  |  |  |  |  |  |  |  |  | 27 | 0 | 34 |  | 34 | 0 | 38 |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 5 | 10 | 15 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 6 | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 17 | 12 | 27 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Gantt Chart**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TASK** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 |
| **1** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **2** |  |  |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **3** |  |  |  |  |  |  |  |  |  |  |  |  |  | 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **4** |  |  |  |  |  |  |  |  |  |  |  |  |  | 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **5** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 3,4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **6** |  |  |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **7** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 5,6 |  |  |  |  |  |  |  |  |  |  |
| **8** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 7 |  |  |  |