# CS 271 – Development Part 2

## Core Gameplay Features

* **Base Damage Calculations:**
  + Code for calculating base damage for both the player and enemies is implemented.
* **Player Healing:**
  + Functionality for player healing during gameplay is complete.

## Damage Multipliers

* Development (WIP) of damage multipliers specific to player classes:
  + **DPS Multiplier:** Focused on per second damage output.
  + **AOE Multiplier:** Damage for a multiple enemy encounter based on positioning.
  + **Critical Hit (Randomized Damage):** A function for calculating critical hits to a random enemy
* Further clarification and refinement are ongoing.

## Turn-Based Logic

* **Asynchronous Turn Functions:**
  + Logic to manage turn-by-turn interactions is implemented, allowing smooth transitions between player and enemy actions.

## Status Effects

* **Stun Mechanic:**
  + A function is in development to skip an enemy's turn as part of a strategic combat element.

## UI Functionality and Animation

* **Button Selection:**
  + Design for buttons for button-based inputs to trigger various player actions (e.g., attacking, healing, or using items) is still in development.
* **Animation:**
  + Combat and secondary animation are still undergoing refinement

## Text Feedback

* **General Text Output:**
  + Functions for displaying relevant information (e.g., damage dealt, healing amount, or enemy actions) in the UI are in progress.
* **Unity Project Sharing:**
  + Additional animations and UI work are ongoing and will be integrated after team members receive the shared project files.

## Current Codebase:

* **General Architecture**
  + Player and enemy instances inherit from a general Unit class and core functions are called in the game’s main script
  + BattleHUD script is separate for display functions.
  + Enumeration is used instead of a switch-case for main game states during the battle [turn system]

**UNIT CLASS:**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Unit : MonoBehaviour

{

public string unitName;

public int unitLevel;

public int baseDmg;

public int addDmg=0;

public int maxHP;

public int currentHP;

public void Heal(int amount)

{

currentHP += amount;

if (currentHP > maxHP)

currentHP = maxHP;

}

public void Selection(int skill)

{

if (skill == 1)

{

Debug.Log("Selected skill 1");

//damage multiplier

}

else if (skill == 2)

{

Debug.Log("Selected skill 2");

//damage multiplier

}

else if (skill == 3)

{

Debug.Log("Selected skill 3");

//damage multiplier

}

else if (skill == 4)

{

Debug.Log("Selected skill 4");

}

addDmg \*= skill;

}

public bool TakeDamage(int baseDmg)

{

int totalDmg = baseDmg + addDmg;

currentHP -= totalDmg;

if (currentHP <= 0)

return true;

else

return false;

}

}

**COMBAT MANAGER [MAIN LOOP]:**

System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public enum BattleState { START, PLAYERTURN, ENEMYTURN, WON, LOST }

public class BattleSystem : MonoBehaviour

{

public GameObject playerPrefab;

public GameObject enemyPrefab;

public Transform playerBattleStation;

public Transform enemyBattleStation;

Unit playerUnit;

Unit enemyUnit;

public Text dialogueText;

public BattleHUD playerHUD;

public BattleHUD enemyHUD;

public BattleState state;

// called before the first frame update

void Start()

{

state = BattleState.START;

StartCoroutine(SetupBattle());

}

IEnumerator SetupBattle() //for each enemy in room, spawn prefab

{

//for [enemy] in room

GameObject playerGO = Instantiate(playerPrefab, playerBattleStation);

playerUnit = playerGO.GetComponent<Unit>();

GameObject enemyGO = Instantiate(enemyPrefab, enemyBattleStation);

enemyUnit = enemyGO.GetComponent<Unit>();

dialogueText.text = "You face a " + enemyUnit.unitName + "...";

playerHUD.SetHUD(playerUnit);

enemyHUD.SetHUD(enemyUnit);

yield return new WaitForSeconds(2f);

state = BattleState.PLAYERTURN;

PlayerTurn();

}

IEnumerator PlayerAttack()

{

bool isDead = enemyUnit.TakeDamage(playerUnit.baseDmg + playerUnit.addDmg);

enemyHUD.SetHP(enemyUnit.currentHP);

dialogueText.text = "The attack is successful!";

yield return new WaitForSeconds(2f);

if(isDead)

{

state = BattleState.WON;

EndBattle();

} else

{

state = BattleState.ENEMYTURN;

StartCoroutine(EnemyTurn());

}

}

IEnumerator EnemyTurn() //enemy coroutine

{

dialogueText.text = enemyUnit.unitName + " attacks!";

yield return new WaitForSeconds(1f);

bool isDead = playerUnit.TakeDamage(enemyUnit.baseDmg);

playerHUD.SetHP(playerUnit.currentHP);

yield return new WaitForSeconds(1f);

if(isDead)

{

state = BattleState.LOST;

EndBattle();

} else

{

state = BattleState.PLAYERTURN;

PlayerTurn();

}

}

void EndBattle()

{

if(state == BattleState.WON)

{

dialogueText.text = "You won the battle!";

} else if (state == BattleState.LOST)

{

dialogueText.text = "You were defeated.";

}

}

void PlayerTurn()

{

dialogueText.text = "Choose an action:";

}

IEnumerator PlayerHeal()

{

playerUnit.Heal(5);

playerHUD.SetHP(playerUnit.currentHP);

dialogueText.text = "You healed for 5 HP!";

yield return new WaitForSeconds(2f);

state = BattleState.ENEMYTURN;

StartCoroutine(EnemyTurn());

}

public void OnAttackButton()

{

if (state != BattleState.PLAYERTURN)

return;

StartCoroutine(PlayerAttack());

}

public void OnHealButton()

{

if (state != BattleState.PLAYERTURN)

return;

StartCoroutine(PlayerHeal());

}

public void OnSkillButton( Button button)

{

string btnName = button.name;

switch (btnName)

{

case "Skill1":

playerUnit.Selection(1);

dialogueText.text = "You used a Lute!";

break;

case "Skill2":

playerUnit.Selection(2);

dialogueText.text = "You used a Drum!";

break;

case "Skill3":

playerUnit.Selection(3);

dialogueText.text = "You used a Trumpet!";

break;

case "Skill4":

playerUnit.Selection(4);

dialogueText.text = "You used a bagpipes!";

break;

default:

Debug.LogError("Button not found: ");

break;

}

}

}

**BATTLE HUD:**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class BattleHUD : MonoBehaviour

{

public Text nameText;

public Text statsText;

public Slider hpSlider;

public void SetHUD(Unit unit)

{

nameText.text = unit.unitName;

statsText.text = "Atk:" + unit.baseDmg + " Def:" + unit.maxHP;

hpSlider.maxValue = unit.maxHP;

hpSlider.value = unit.currentHP;

}

public void SetHP(int hp)

{

hpSlider.value = hp;

}

}