天地鳴動：プロトタイプ主要システム設計書

1：ゲームサイクルとフローシステム

1.1:GameManagerクラス
ゲームの進行状態、フェイズ管理、シーン遷移などを統括するシングルトンクラス

・処理
・ゲーム全体の初期化と終了処理
・GamePhase（出撃、戦闘、ステージクリア、ゲームオーバー）の管理と遷移
・シーン管理（タイトル、マップ選択、戦闘、リザルト）
・各フェイズにおけるUIの表示/非表示制御
・セーブ/ロードシステムの呼び出し
・クラス
・CurrentPahse：現在のゲームフェイズ(enum GamePhase)
・CurrentStageId：現在のステージID(int)
・メソッド
/InitializeGane();ゲーム全体の初期化
/ChangePhase(GamePhase newPhase);フェイズを切り替える
/LoadScene(String SceneName);シーンをロード
/SaveGame();ゲームをセーブする
/LoadGame();ゲームをロードする

1.2:GamePhase Enum
ゲームの各フェイズを定義する列挙型
/Title;タイトル画面
/Deployment;出撃フェイズ
/Battle;戦闘フェイズ
/StageClear;ステージクリア
/GameOver;ゲームオーバー
/Loading;ロード中

2:バトルシステム
ターン制シミュレーションRPGの核となるシステム群

2.1:BattleManagerクラス
戦闘全体の進行、ターン管理、ユニットの行動順序などを統括するシングルトンクラス

・処理
・戦闘の初期化と終了処理
・ターン開始/終了処理
・ユニットの行動順序管理
・勝利/敗北条件の判定とゲームオーバー処理
・不確定要素システムとの連携
・A*アルゴリズムによるユニットの移動命令の中継
・クラス
/CurrentTurn;現在のターン数(int)
/ActiveUnit;現在行動中のユニット(Unitクラスインスタンス)
/AllUnits;現在マップ上に存在する全ユニットのリスト(List<Unit>)
・メソッド
/StartBattle();戦闘を開始する
/EndTurn();ターンを終了し、次のターンの処理を開始する

/CheckWinCondition();勝利条件を判定する
/CheckLoseCondition();敗北条件を判定する
/ApplyUncertaintyEffect();不確定要素の効果を適用する
/MoveUnit(Unit unit, Vector2Int targetPosition)指定ユニットをターゲット位置へ移動させる(内部でA*を呼び出す)
/PerformAttack(Unit attacker, Unit defender)ユニット間の攻撃処理実行する

2.2:Unitクラス

ゲーム内の各ユニット(プレイヤー、敵)の情報を管理するクラス

・処理
- ・ユニットの基本ステータス管理
- ・現在のHP、移動力、経験値の管理
- ・装備品、特性、現在の行動状態(行動済みか否か)の管理
- ・レベルアップ、経験値獲得処理

・クラス
- /UnitId;ユニットのユニークID(string)
- /UnitName;ユニット名(string)
- /CurrentHP;現在のHP(int)
- /MaxHP;最大HP(int)
- /CurrentMovementPoints;現在の移動力(A*計算で参照される)(int)
- /AttackPower;攻撃力(int)
- /Skill;技(int)
- /Speed;速さ(int)
- /Defense;防御力(int)
- /CurrentEXP;現在の経験値(int)
- /CurrentLevel;現在のレベル(int)
- /UnitType;(enum UnitType)ユニットの特性(enum UnitType)
- /EquippedWeapon;(Weapon)装備中の武器(Weaponクラスインスタンス)
- /CurrentGridPosition;(Vector2Int)マップ上の現在のグリッド座標(Vector2Int)
- /HasActedThisTurn;今ターン行動済みか(bool)

・メソッド
- /UpdatePosition(Vector2Int newPosition);ユニットのグリッド座標を更新する
- /ConsumeMovementPoints(int cost);移動ポイントを消費する
- /TakeDamage(int damage);ダメージを受ける
- /GainExperience(int exp)経験値を獲得し、レベルアップ処理を行う
- /ApplyBuff(BuffType type, int value)バフ効果を適用する
- /ApplyDebuff(DebuffType type, int value)デバフ効果を適用する
- /ResetAction();ターン開始時に行動済み状態をリセットする

2.3:UnitType Enum

ユニットの特性を定義する列挙型

- ・Infantry;歩兵
- ・Aquatic;水棲
- ・Flying;飛行
- ・Cavairy;騎馬
- ・Heavy;重兵
- ・Archer;弓兵

・Mountain;山賊

2.4:Weaponクラス
武器の情報を管理するクラス
・処理
　・武器の基本ステータス管理
　・武器経験値、レベルアップ管理
・クラス
　/WeaponId;武器のユニークID(string)
　/WeaponName;武器名(string)
　/WeaponType;武器の種類(enum WeaponType)
　/AttackPower;攻撃力(int)
　/HitRate;命中率(int)
　/Range;攻撃範囲(int)
　/CurrentWeaponEXP;現在の武器経験値(int)
　/CurrentWeaponLevel;現在の武器レベル(int)
・メソッド
　/GainWeaponExperience(int exp);武器経験値を獲得し、レベルアップ処理を行う

2.5:WeaponType Enum
武器の種類を定義する列挙型

・Sword;剣
・Axe;斧
・Lance;槍
・Bow;弓

2.6:戦闘計算式

・武器相性：剣＞斧＞槍の三すくみ
　・有利な場合：命中+15%
　・不利な場合：命中-15%
・命中率計算：（攻撃側の技*2 + 武器の命中）-（防御側の速さ*2 + 地形回避率）
・回避率計算：攻撃側の命中率に対する、防御側の実質的な回避率
・追撃：攻撃側の速さ - 防御側の速さ≧4の場合、追撃が発生2回攻撃
・ダメージ計算：（攻撃力 + 武器攻撃力）-（防御力 + 地形防御力ボーナス）