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

・責務
　　・ゲーム全体の初期化と終了処理の統括
　　・GamePhase(ステージ選択、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(タイミングは出撃フェイズ、タイトル)
　　・ゲーム全体にわたるイベント発生の仲介、画面遷移指示
・主要なプロパティ
　　・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
　　/InitializeGame();ゲーム全体の初期化
　　/PhaseExit();現在のフェイズの終了処理
　　/PhaseEnter();次フェイズの開始処理
　　/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
　　/StageSelect;ステージ選択
　　/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

1.3:ゲームの遷移フロー

A【ゲーム起動】ーー＞B【タイトル画面】;

B－－新規ゲーム開始ーー＞C【ステージ選択】;
B－－ロードゲームーー＞D【出撃フェイズ】;

C－－＞D【出撃フェイズ】;

D－－ユニット選択/配置ーー＞出撃決定ーー＞E【戦闘フェイズ】;
D－－セーブーー＞D
D－－C【ステージ選択】に戻るーー＞C;

E－－勝利条件達成ーー＞F【ステージクリア画面】ーー＞B【タイトル画面】;

E――敗北条件達成――＞G【ゲームオーバー画面】――＞B【タイトル画面】;

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

・責務
　　・戦闘の初期化と終了処理
　　・ターン開始/終了処理
　　・プレイヤーターン、敵ターン、不確定要素フェイズなどのターン進行管理
　　・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
　　・ダイクストラ法によるユニットの移動命令の中継
　　・敵ターンの開始時と終了時にAIManagerを呼び出す
・主要なプロパティ
　　/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>)
　　/ProcessTurn();ターン処理の分岐において、プレイヤーターン後に
　　AIManager,ProcessEnemyTurn()を呼び出す
　　/ExecuteUnitAction(Unit unit, ActionData action);敵AIからの行動命令(AIAction)を受け取り、
　　対応するアクション(移動、攻撃)を実行する
・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DijkstraPathfinder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

・責務
　　・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
・主要なプロパティ
　　/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（ダイクストラ法計算で参照される）(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)
/_isPlayerUnit;プレイヤーユニットか敵ユニットかを識別する(bool)
/_aiDate;敵ユニットの場合、AI固有の設定データ(攻撃優先順位、行動パターンなど)を保持する構造体やクラスへの参照

・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
/RemoveBuff();バフ効果を元に戻す
/ApplyDebuff(DebuffType type, int value)デバフ効果を適用する
/RemoveDebff();デバフ効果を元に戻す
/ResetAction();ターン開始時に行動済み状態をリセットする
/GetPossibleAction();現在の状況で敵ユニットが取り得る行動のリストを返す(AIが行動を評価する際に利用)

2.3:UnitType Enum
ユニットの特性を定義する列挙型

・Infantry;歩兵(基準となるユニット)
・Aquatic;水棲(水場、水害地形で影響受けない)
・Flying;飛行(地形の影響を無視するが、弓兵に弱い)
・Cavalry;騎馬(基本移動能力が高いが、地形の影響を強く受ける)
・Heavy;重兵(基本移動能力は低い、防御力が高い)
・Archer;弓兵(遠距離攻撃が可能、近距離攻撃は不可)
・Mountain;山賊(山地形で影響を受けない)

2.4:Weaponクラス
武器の情報を管理するクラス
・責務
・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
・主要なプロパティ
/WeaponId;武器のユニークID(string)
/WeaponName;武器名(string)
/WeaponType;武器の種類(enum WeaponType)
/AttackPower;武器威力(int)
/HitRate;武器の命中率(int)
/Range;攻撃範囲/射程(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、規定値に達した場合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）

3：マップシステム
ゲームマップの生成、管理、地形効果などを扱うシステム。ダイクストラ法で使用されるマップデータを提供する

3.1:MapManagerクラス
ゲームマップの生成、管理、地形効果の適用などを統括するシングルトンクラス

・責務
　　・ステージIDに基づいてマップデータをロードし、マップを生成する
　　・各グリッドの地形タイプ、ユニットの配置状況を管理
　　・地形による移動コスト、回避率、防御力ボーナスなどを提供
　　・不確定要素による地形変化を適用する
　　・DijkstraPathfinderが必要とするマップ情報(タイルごとの移動コスト、通行可能性)を提供
・主要なプロパティ
　　/GridSize;マップのグリッドサイズ(Vector2Int)
　　/TileGrid;各グリッドの情報を格納する2次元配列(Tile[,])
・メソッド
　　/LoadMap(int stageId);指定されたステージIDのマップデータをロードし、グリッドを生成
　　/GetTileAt(Vector2Int position);指定座標のタイル情報を取得する
　　/GetMovementCost(Vector2Int position, UnitType unitType);指定されたグリッドとユニットタイプに応じた移動コストを返す。DijkstraPathfinderがこのメソッドを利用する
　　/IsValidGridPosition(Vector2Int position);指定された座標がマップ範囲内か判定する
　　/ApplyTerrainEffect(Vector2Int position);指定座標の地形効果（防御/回避ボーナスなど）を適用する
　　/ChangeTerrain(Vector2Int position, TerrainType newType)特定のグリッドの地形タイプを変更する

3.2:Tileクラス
マップの各グリッドの情報を管理するクラス
・責務
　　・グリッドの座標、地形タイプ、上に存在するユニットの参照を保持
　　・地形ごとの防御力ボーナス、回避ボーナスを提供する

・主要なプロパティ

/GridPosition;グリッドの座標(Vector2Int)
/TerrainType;地形の種類(enum TerrainType)
/OccupyingUnit;そのグリッドに存在するユニット(Unitクラスインスタンス、ない場合は null)

・メソッド

/GetDegenseBonus();防御力ボーナスを返す
/GetEvadeBonus();回避ボーナスを返す
/SetOccupyingUnit(Unit unit);指定グリッドに存在するユニットを設定する

3.3:TerrainType Enum
地形の種類を定義する列挙型

/Plain;平原
/Forest;森
/Mountain;山
/Desert;砂漠
/Water;水場
/River;川
/Snow;積雪
/Flooded;水害
/Landslide;土砂崩れ
/Paved;舗装

3.4:DijkstraPathfinderクラス
ダイクストラ法を用いて、開始地点から到達可能な全てのマスへの最短コストを計算するクラス

・責務

・MapManagerからマップ情報を取得し、ダイクストラ法アルゴリズムを実行する
・指定された移動力内で到達可能な全てのマスのリストとその最小移動コストを返す
・計算された結果から、任意の目標地点への最短経路を再構築して提供

・主要なプロパティ

・なし（メソッド中心のユーティリティクラス）

・メソッド

/FindReachableTiles(Vector2Int startPos, Unit unit);ダイクストラ法アルゴリズムを実行し、指定された移動力(unit.CurrentMovementPoints)内で到達可能な全てのマス(Dictionary<Vector2Int,int>)とその最小移動コストを返す。UIでの移動範囲表示や実際の移動パス計算に利用
/GetPathToTarget(Vector2Int startPos, Vector2Int targetPos, Unit unit);
FindReachableTilesの結果から、指定された目標地点への最短経路(List<Vector2Int>)を再構築して返す

3.5PathNodeクラス(DijkstraPathfinder内部用)
DijkstraPathfinderが経路計算に用いるノードオブジェクト

・責務

・グリッド座標、累積コスト、親ノードの情報を保持

・主要なプロパティ

/GridPosition;グリッド座標(Vector2Int)
/CostFromStart;スタート地点からの実コスト(int)
/Parent;経路再構築のための親ノード(PathNode)
/IsWalkable;そのマスが通行可能かの判定(bool)

4：不確定要素システム
ターン経過によって天候や地形を変化させるシステム

4.1:UncertaintyManagerクラス
不確定要素の発生、適用、兆候の管理を統括するシングルトンクラス

・責務
・特定のターン経過またはイベントによって不確定要素を発生させる
・発生する不確定要素の種類をランダムまたは設定されたロジックで決定
・不確定要素によるマップ（地形）やユニット（ステータス）への影響を適用する
・不確定要素の変化の兆候を生成し、UIManagerを通じて表示
・主要なプロパティ
/CurrentWeather;現在の天候タイプ(enum WeatherType)
/NextUncertaintyEventTurn;次の不確定要素が発生するターン(int)
・メソッド
/GenerateUncertaintyEvent();不確定要素イベントを生成する
/ApplyWeatherEffect(WeatherType weather);天候効果をマップ全体または指定範囲に適用する
/PredictUncertainty();不確定要素の変化の兆候を生成し、プレイヤーに提示する
/GetAffectedUnits(WeatherType weather);特定の天候によって影響を受けるユニットのリストを返す
/ApplyTerrainChange(Vector2Int position, TerrainType newType);不確定要素による地形の変化を適用する
/ApplyUnitStatusEffect(Unit unit, StatusEffect effect);不確定要素によるユニットへの影響を適用する

4.2:WeatherType Enum
天候の種類を定義する列挙型
/Clear;晴れ
/HeavyRain;豪雨
/Blizzard;吹雪
/Sandstorm;砂嵐
/Drought;日照り
/Tornado;竜巻

5：敵AIシステム
敵ユニットがプレイヤーの行動、不確定要素による変化に対応して、戦略的な判断に基づいた行動(移動、攻撃など)を実行するシステム

5.1:AIManagerクラス
敵ユニットのAI行動を統括・管理するシングルトンクラス

・責務
・敵ターンの開始時に、アクティブな敵ユニットの行動順序を決定し、行動を命令する
・各敵ユニットのAIロジック(思考ルーチン)を呼び出す
・行動評価(ターゲット選定、行動決定)に必要な情報をBattleManagerやMapManagerから取得し、AIルーチンに渡す
・複数の敵ユニットがいる場合、全体の状況を考慮した連携行動の判断をサポートする(優先度付けなど)

・AIの難易度設定に応じて、思考の複雑さや判断の精度を調整する

・主要なプロパティ

/_activeEnemyUnits;現在行動可能な敵ユニットのリスト(List<Unit>)

/_difficultyLevel;現在のAIの難易度(enum AIDifficultyLevel)

/_currentUnitIndex;現在行動中の敵ユニットのインデックス(int)

・メソッド

/ProcessEnemyTurn();敵ターンのメイン処理を開始し、各敵ユニットの行動を順序決定する

/SelectBestAction(Unit enemyUnit);特定の敵ユニットにとって最適な行動(移動、攻撃)を評価し決定する

・入力：Unit(行動中の敵ユニット)

・出力：AIAction(決定された行動タイプとターゲット情報を含む構造体)

/EvaluateMove(Unit enemyUnit);敵ユニットの移動範囲内で、最も有利な位置を評価する

/EvaluateAttack(Unit enemyUnit);攻撃範囲内で、最適な攻撃対象と使用する武器を評価する

/FindVulnerablePlayerUnit(Unit enemyUnit);敵ユニットが攻撃可能な範囲内で、最も脆弱なプレイヤーユニット（HPが低いなど）を特定する

/FindOptimalPosition(Unit enemyUnit);移動と攻撃を組み合わせ、最適な位置を決定する

/ExecuteAIAction(AIAction action);決定されたAI行動を実行する(BattleManagerのメソッドを呼び出す)

5.2:AIAction構造体

AIManagerが決定した敵の行動を表す構造体

・主要なプロパティ

・ActionType;行動の種類(enum AIActionType)

・TargetUnit;行動の対象となるユニット(Unit)

・TargetPosition;移動先となるマップ座標(Vector2Int)

・WeaponId;使用する武器ID(int,攻撃の場合)

5.3:AIActionType Enum

敵の行動の種類を定義する列挙型

/Move;移動

/Attack;攻撃

/Wait;待機(行動不能または適切な行動がない場合)

5.4:AIDifficultyLevel Enum

AIの難易度レベルを定義する列挙型。AIの思考ルーチンの深さや判断ロジックの複雑さに影響を与える

・/Easy;簡易的な判断(最も近い敵を攻撃、HPが低い敵を優先など)

・/Normal;基本的な戦略的判断(有利な地形の利用、地形変化の考慮など)

・/Hard;高度な戦略的判断(連携行動、予測に基づく行動、危険回避など)

6：UI/UXシステム

プレイヤーとのインタラクション、情報表示を担うシステム

6.1:UIManagerクラス

ゲーム内のUI要素の表示、更新、プレイヤーからの入力処理を統括するシングルトンクラス

・責務

・HUD（HPバー、時間表示、ターン数表示など）の管理

・メニュー画面（ユニットのステータス、インベントリ、オプションなど）の表示と制御

・ダイクストラ法で計算された移動可能範囲の表示

・ダイクストラ法で計算された経路の視覚的な表示
・ユニットの選択、移動範囲表示、攻撃対象選択などのインタラクション処理
・不確定要素の兆候、メッセージなどの表示
・プレイヤーの入力を受け取り、BattleManagerにイベントを通知
・敵AIの行動中に、AIの思考中であることや、敵ユニットの行動(移動先や攻撃対象)を視覚的にフィードバックするためのUI表示を管理

・主要なプロパティ

マップ画面時
/MapHPBarUI;マップ画面用のHPバーのUIコンポーネント
/UnitInfoPanel;ユニット詳細情報パネル
/TurnCounterText;ターン数表示テキスト
/UncertaintyForecastText;不確定要素の予測表示テキスト
/MoveHighlightPrefab;移動可能マスをハイライトするPrefab
/PathLineRenderer;経路を描画するためのLineRendererコンポーネント
/TimeDisplayUI;時間表示のUIコンポーネント
/PhaseDisplayUI;フェイズ表示のUiコンポーネント
/CursorInfoPanel;カーソル位置のユニット情報と地形情報を表示するパネル
/UnitStatusPanel;選択中のユニットの現在のステータスを表示するパネル
/TerrainInfoPanel;カーソル地点の地形の詳細情報を表示するパネル
/MenuIconUI;各種メニューへのアクセスを示すアイコンのUIコンポーネント
戦闘画面時
/AttackerStatusUI;戦闘時の攻撃側のステータスを表示するUIコンポーネント
/DefenderStatusUI;戦闘時の防御側のステータスを表示するUIコンポーネント
/BattlePredictionUI;戦闘結果の予測（ダメージ、命中率）を表示するUIコンポーネント
/EquippedWeaponInfoUI;戦闘時の装備中の武器情報を表示するUIコンポーネント

・メソッド

/UpdateHUD();マップ画面のHUD(HPバー、フェイズ表示など)情報を更新する
/ShowUnitInfo(Unit unit);指定ユニットの詳細情報を表示する
/DisplayMovementRange(Dictionary<Vector2Int,int> reachableTiles);移動可能な範囲をマップ上にハイライト表示する
/DrawPath(List<Vector2Int> path);計算された経路をマップ上に描画する
/ClearHighlightAndPath();ハイライトと経路表示をクリアする
/DisplayAttackRange(List<Vector2Int> attackableTiles);攻撃可能な範囲をマップ上にハイライト表示する
/ShowForecast(string forecastMessage);不確定要素の予測メッセージを表示する
/OnTileClicked(Vector2Int clickedPosition);マップ上のタイルがクリックされた際の処理
/OnUnitSelected(Unit selectedUnit);ユニットが選択された際の処理
/UpdateMapScreemUI();マップ画面時のUI（HPバー、時間、フェイズ、カーソル情報、ユニットステータス、地形情報、メニューアイコン）を更新する
/ShowBattleScreenUI(Unit attacker, Unit defender, BattlePredictionData prediction, Weapon attackerWeapon);戦闘画面時のUI（両者のステータス、戦闘予測、装備中の武器情報）を表示する
/HideBattleScreenUI();戦闘画面UIを非表示にする
/UpdateCursorInfo(Unit unitOnCursor, Tile terrainInfo);カーソル地点のユニット情報と地形情報を更新する
/UpdateUnitStatusPanel(Unit unit);選択中のユニットのステータスパネルを更新する
/ShowBattlePrediction(Unit attacker, Unit defender, BattlePredictionData prediction);戦闘予測UIを表示し、データを設定する
/ShowAIThinkingIndicataor();AIが思考中であることを示すUIを表示
/ShowEnemyActionPreview(AIAction action);敵ユニットが実行する行動のプレビュー(移動経路、攻撃対象のハイライトなど)を表示
/ShowErrorMessage(string message);エラーメッセージの表示

7：データ管理システム
ゲーム内の各種データを管理し、セーブ/ロードを行うシステム

7.1:DataManagerクラス
ユニット、武器、マップなどのマスターデータを管理し、セーブデータとの連携を行うシングルトンクラス

・責務
・CSVまたはXML形式のマスターデータをロードし、メモリ上に保持
・セーブデータの読み書き（PlayerPrefsまたはファイルI/Oを使用）
・データの一貫性と整合性の保証
・AI固有のマスターデータ(AI行動パターン、難易度ごとの設定など)のロードと管理
・主要なプロパティ
/UnitMasterData:ユニットのマスターデータ(Dictionary<string, UnitData>)
/WeaponMasterData;武器のマスターデータ(Dictionary<string, WeaponData>)
/MapMasterData;マップのマスターデータ(Dictionary<int , MapData>)
/_aiPatternDate;AIの行動パターンに関するデータ(Dictionary<int , AIData>)

・メソッド
/LoadAllMasterData();全てのマスターデータをロードする
/SaveGameData(GameSaveData saveData);ゲームデータをセーブする
/LoadGameData();ゲームデータをロードする
/GetUnitData(string unitId);指定IDのユニットマスターデータを返す
/GetWeaponData(string weaponId);指定IDの武器マスターデータを返す

7.2:データクラス（構造体）
各マスターデータに対応するデータ構造を定義

/UnitData:UnitId,UnitName,BaseHP,BaseAttackPower,..（マスターデータ用）..
/WeaponData;WeaponId,WeaponName,WeaponType,BaseAttackPower,HitRate,Range,...
（マスターデータ用）
/MapData;StageId,MapSize,InitialUnitPlacements,TerrainLayout,..（マスターデータ用）
/GameSaveData;CurrentTurn,PlayerUnitsData(List<UnitSaveData>),
EnemyUnitsData(List<UnitSaveData>),CurrentWeather,....（セーブデータ用）
/UnitSaveData;UnitId,CurrentHP,CurrentMovementPoints,CurrentEXP,EquippedWeaponId,
GridPosition,HasActedThisTurn,...（ユニットのセーブデータ用）

7.2.1;AIデータクラス（構造体）
AIの行動パターンや設定を定義するデータクラスを必要に応じて追加
/AIPatternData;敵の種類や役割に応じたAIの行動ロジックを定義するデータ
AIDifficultySettingData;難易度ごとのAIの判断基準やパラメータを設定するデータ

8：サウンドシステム
ゲーム内でのBGMや効果音の再生を管理するシステム

8.1:AudioManagerクラス
BGM、SE、ボイスの再生、停止、音量調整を統括するシングルトンクラス

・責務
・BGMのループ再生、切り替え

・SEの再生（同時再生制限、優先度設定など）
・音量の設定（マスター、BGM、SE、ボイス）

・主要なプロパティ
/BGMVolume;BGMの音量(float)
/SEVolume;SEの音量(float)
/MasterVolume;全体のマスター音量(float)

・メソッド
/PlayBGM(string bgmClipName);指定されたBGMを再生する
/StopBGM();現在再生中のBGMを停止する
/PlaySE(string seClipName);効果音を再生する
/SetBGMVolume(float volume);BGMの音量を設定する
/SetSEVolume(float volume);SEの音量を設定する
/SetMasterVolume(float volume);マスター音量を設定する

9：視覚化システム
ゲームオブジェクトの見た目の制御、アニメーション、エフェクトを管理するシステム

9.1:TileVisualizerクラス
各タイルGameObjecctにアタッチされ、そのタイルの見た目や状態変化を表現する

・責務
・地形タイプに応じたSpriteの表示
・選択時、移動可能範囲、攻撃可能範囲などのハイライト表示
・不確定要素による地形変化時のアニメーション
・不確定要素による地形変化時や、ユニットがタイル上を移動する際のアニメーション

・主要なプロパティ
/Tile:関連するTileクラスのインスタンス
/SpriteRenderer;タイルのSpriteRendererコンポーネント
/HighlightRenderer;ハイライト表示用のRendererコンポーネント

・メソッド
/UpdateVisual(TerrainType type, bool isHighlighted = false);タイルの見た目を更新し、地形タイプに応じたSprite変更やハイライトのオン/オフを行う
/AnimateTerrainChange(TerrainType newType);地形変化時のアニメーションを再生する

9.2:UnitVisualizerクラス
各ユニットGameObjectにアタッチされ、ユニットの見た目の移動、アニメーション、エフェクトを表現する

・責務
・ユニットのSprite表示、向きの変更
・ユニットのHPバーなどのUI要素の表示
・ダイクストラ法で計算された経路に沿ったユニットの滑らかな移動アニメーション
・戦闘アニメーション、待機アニメーションの再生
・ダメージエフェクト、レベルアップエフェクトの再生

・主要なプロパティ
/Unit;関連するUnitクラスのインスタンス
/Animator;ユニットのアニメーターコンポーネント

・メソッド
/MoveAlongPathVisual(List<Vector2Int> path, float moveSpeed);経路に沿ってユニットを滑らかに移動させるコルーチンを開始
/PlayAttackAnimation();攻撃アニメーションを再生する

/PlayDamageAnimation();ダメージアニメーションを再生する
/UpdateHPBar(int currentHP, int maxHP)HPバーを更新する
/SetDirection(Vector2Int direction);ユニットの向きを更新する

10：入力システム
プレイヤーからの入力を管理し、ゲーム操作に変換するシステム

10.1:InputManagerクラス
キーボード、マウス、ゲームパッド（XBOXコントローラー想定）からの入力を抽象化し、ゲームロジックに渡すシングルトンクラス
・責務
・各入力デバイスからの入力を検知し、適切なゲームアクションにマッピングする
・入力状態（押されている、離された、ホールドされているなど）を管理する
・UI操作とゲーム操作の切り替えを管理する
・主要なプロパティ
/CurrentInputType;現在の入力タイプ(KeyboardMouse,Gamepadなど)(enum InputType)
/CursorGridPosition;マップ上の現在カーソル地点があるグリッド座標
・メソッド
/InitalizeInput();入力システムを初期化する
/UpdateInput();毎フレーム入力状態を更新する
/IsConfimButtonPressed();決定ボタン(Aボタンなど)が押されたか判定する
/IsCancelButtonPressed();キャンセルボタン(Bボタンなど)が押されたか判定する
/GetMoveDirection();カーソル移動の方向(D-Pad,Left Stick)を返す
/IsMenuButtonPressed();メニューボタン(Startボタンなど)が押されたか判定する
/InitializeXboxConrtoller();XBOXコントローラーの入力設定を初期化する
ProcessXboxControllerInput();XBOXコントローラーからの入力を処理し、ゲームアクションに変換する

10.2:InputType Enum
現在のアクティブな入力デバイスの種類を定義する列挙型

/KeyboardMouse;キーボードとマウス
/Gamepad;ゲームパッド(XBOXコントローラーなど)

10.3:XboxControllerBinding
XBOXコントローラーの各ボタンとスティックのゲーム内での役割を定義

・マップ画面時
・左スティック/D-Pad;カーソル移動(CursoGridPositionを更新)
・Aボタン;決定/ユニット選択/移動確定
・Bボタン;キャンセル/選択解除/行動キャンセル
・Startボタン;メニューを開く
・Yボタン;ユニットステータス詳細表示切り替え
・Xボタン;攻撃範囲表示切り替えなど
・LB/RB;ユニット切り替え(次/前のユニットへフォーカス)
・戦闘画面時
・Aボタン;攻撃確定
・Bボタン;攻撃キャンセル
・Xボタン;武器情報詳細表示切り替え
・左スティック/D-Pad;攻撃対象選択(複数存在する場合)

11：技術要件と開発環境

・開発言語：C#
・開発環境：Visual Studio Code
・ゲームエンジン：Unity
・バージョン管理：Git
・データエディタ：CSVまたはXML形式でインポート/エクスポート可能
・AIツール：Gemini

12：主要システム設計書作成について

12.1:AIツールの利用について
本主要システム設計書の作成にあたり、以下の目的でAIツール(Gemini)を利用しました。AIツールは、あくまで補助的な役割として活用し、最終的な内容の決定と責任は作成者である私個人に帰属します。

・設計書作成にあたっての参考資料の作成支援
文書作成初期段階において、本設計書のイメージに近いゲーム作品（例：ファイアーエムブレム封印の剣」など）を参考に、システム構成や機能要素に関する網羅的な参考資料の作品をAIツールに依頼しました。この参考資料を基に、天地鳴動プロジェクトに必要な情報や考慮すべき点を検討し、本設計書の骨子を策定しました。

・作成資料のフィードバック及び誤字脱字の添削
作成した各項目や説明文について、内容の網羅性、具体性、論理的一貫性、説明の不足点などをAIツールに複数回フィードバックを求めました。これにより、設計書の品質向上を図るとともに、誤字脱字や表記ゆれの確認にも利用し、文書としての精度を高めました。

・知識面や技術理解度の補填
ゲーム作品全体のシステム設計は多岐にわたるため、自身の知識や技術だけでは全体像を見据えた網羅的かつ詳細な設計書の制作が困難な側面がありました。AIツールを特に不足している知識や技術理解を深めるための強力な補助として活用し、特定のシステム（例：ダイクストラ法を用いた経路探索、不確定要素のロジック構築、敵AIシステムなど）に関する情報収集や、技術的な実装方法の概念を理解する上で大いに役立てました。

12.2