Turn-based-Template

Generated by Doxygen 1.9.2

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

Namespace Documentation

4.1 TurnBased Namespace Reference

Classes

- class GameManager

  Responsible for controlling data for gameplay and scene change handling.

4.2 TurnBased.Data Namespace Reference

Classes

- class InitialSituation

  A full game initial state contains different data that is used by the game to initialize first turn.
- struct UnitInitialState

  Contains the data needed for a unit in initial situation.
- class UnitSetup

  Controls unit data setup, used by UnitsFactory to initialize units on the grid.

Enumerations

- enum **UnitType** { **Infantry** , **Tank** , **Plane** }

4.3 TurnBased.Gameplay Namespace Reference

Classes

- class Cell

  Controls the color and data for cell in Position.
- class GameplayManager

  Controls Gameplay loop, giving turns and determining who wins.
- class Unit

  Controls Unit runtime data, current health and available moves.
- class **UnitInitializer**

  Initializes and verifies unit data and initial scenarios, to be used by Gameplay.GameplayManager
- class UnitsFactory

Enumerations

- enum **CellHighlightMode** { **NotHighlighted** , **CanChooseIt** , **CanMoveTo** , **CanAttack** }

4.4 TurnBased.Grids Namespace Reference

Classes

- class GridData
- class **GridInitializer**

  Initializes and verifies grid data to be used by Gameplay.GameplayManager
- class GridRenderer

  Responsible for controlling cells and rendering different events.

4.5 TurnBased.Player Namespace Reference

Classes

- class AIPlayer

  Controls AI Player behavior, making decisions and ending turns, this implements a dumb movement AI in which it returns random available position.
- class HumanPlayer

  Gives input control to the human player once he gets the turn.
- class InputManager
- class Playerbase

  Base class for player, controls turn loop once it got the turn.
- class SmartAIPlayer

  A Smarter AI player with deterministic moves instead of random ones.

Enumerations

- enum **PlayerType** { **AI** , **SmartAI** , **Hooman** }

4.6 TurnBased.UI Namespace Reference

Classes

- class EndGamePanel

  The game over panel in which the player can restart with same situation or try a different one.
- class GameplayUIManager

  Responsible for handling gameplay UI Logic and events.
- class HealthBar

  Shows the health for the given unit.
- class HealthBarController

  Controls shown health bars and initializes it.
- class MainMenuUIManager
- class SituationItemUI
- class SituationsLoader
- class SkipButton

4.7 TurnBased.Utils Namespace Reference

Classes

- class **CellUtils**

  Utilities that covers cell related maths.
- class Singleton

  A utility class that supports singleton design pattern, used for cross scene interactions.

Chapter 5

Class Documentation

5.1 TurnBased.Player.AIPlayer Class Reference

Controls AI Player behavior, making decisions and ending turns, this implements a dumb movement AI in which it returns random available position.

Inherits TurnBased.Player.Playerbase.

Inherited by TurnBased.Player.SmartAIPlayer.

Public Member Functions

- **AIPlayer** (List< Unit > units, GameplayUIManager manager, GridData data, GridRenderer gridRenderer)
- override void StartTurn ()

  Starts the player turn, resets its unit energy and gets the UI ready based on its type.

Protected Member Functions

- virtual IEnumerator TurnCoroutine ()

  Controls Turn behavior.
- IEnumerator **AttackIfPossible** (Unit unit)

  Attack if enemy unit found in range.

Protected Attributes

- GridData **m_gridData**

Properties

- override PlayerType **Type** `[get]`

Additional Inherited Members

5.1.1 Detailed Description

Controls AI Player behavior, making decisions and ending turns, this implements a dumb movement AI in which it returns random available position.

5.1.2 Member Function Documentation

5.1.2.1 StartTurn()

```
override void TurnBased.Player.AIPlayer.StartTurn ( )  [virtual]
```

Starts the player turn, resets its unit energy and gets the UI ready based on its type.

Reimplemented from TurnBased.Player.Playerbase.

5.1.2.2 TurnCoroutine()

```
virtual IEnumerator TurnBased.Player.AIPlayer.TurnCoroutine ( )  [protected], [virtual]
```

Controls Turn behavior.

Reimplemented in TurnBased.Player.SmartAIPlayer.

The documentation for this class was generated from the following file:

- Project/Assets/Scripts/Player/AIPlayer.cs

5.2 TurnBased.Gameplay.Cell Class Reference

Controls the color and data for cell in Position.

Inherits MonoBehaviour.

Public Member Functions

- void **ChangeHighlight** (CellHighlightMode cellHighlightMode)

  Changes the highlight mode to given mode.
- void **OnCellPressed** ()

  Serialized event listener, invokes the event for the cell.

Properties

- CellHighlightMode **CurrentHighlightMode** `[getset]`

  Current Highlight mode of the cell in grid renderer.
- Vector2Int **Position** `[getset]`

  Position of the cell in the grid.
- Unit **OccupyingUnit** `[getset]`

  Current occupying unit of the cell.

Events

- Action< Cell > **CellSelected**

  Invoked when the cell is selected via input.

5.2.1 Detailed Description

Controls the color and data for cell in Position.

The documentation for this class was generated from the following file:

- Project/Assets/Scripts/Gameplay/Cell.cs

5.3 TurnBased.UI.EndGamePanel Class Reference

The game over panel in which the player can restart with same situation or try a different one.

Inherits MonoBehaviour.

Public Member Functions

- void **ShowEndGamePanel** (int winningPlayerId)
- void **OnRestartButtonPressed** ()
- void **OnMenuButtonPressed** ()

5.3.1 Detailed Description

The game over panel in which the player can restart with same situation or try a different one.

The documentation for this class was generated from the following file:

- Project/Assets/Scripts/UI/EndGamePanel.cs

5.4 TurnBased.GameManager Class Reference

Responsible for controlling data for gameplay and scene change handling.

Inherits TurnBased.Utils.Singleton< GameManager >.

Public Member Functions

- void **GoToGameplayScene** ()
- void **GoToMainMenu** ()

Properties

- InitialSituation **ChosenInitalSituation** `[getset]`

  The current chosen situation.

5.4.1 Detailed Description

Responsible for controlling data for gameplay and scene change handling.

The documentation for this class was generated from the following file:

- Project/Assets/Scripts/GameManager.cs

5.5 TurnBased.Gameplay.GameplayManager Class Reference

Controls Gameplay loop, giving turns and determining who wins.

Inherits MonoBehaviour.

5.5.1 Detailed Description

Controls Gameplay loop, giving turns and determining who wins.

The documentation for this class was generated from the following file:

- Project/Assets/Scripts/Gameplay/GameplayManager.cs

5.6 TurnBased.UI.GameplayUIManager Class Reference

Responsible for handling gameplay UI Logic and events.

Inherits MonoBehaviour.

Public Member Functions

- void **InitializeHealthbars** (List< Unit > units, float gridScale)

  Initialize the healthbars for the given units and scale it based on the given gridScale
- void **OnSkipButtonPressed** ()

  Invokes Skip Button Pressed event, used as serialized action in unity.
- void **ReadyToEndTurn** ()

  Switchs UI state to ready to end.
- void **ChangeSkipButtonState** (bool state)

  Change the interactability of the skip button.
- void **ShowEndGamePanel** (int playerId)

  Shows end game panel with the given winning player index.
- void **TurnStarted** (PlayerType type)

  Switch UI state to be be at the start of a turn.

Events

- Action **SkipButtonPressed**

  Invoked when the skip button is pressed.

5.6.1 Detailed Description

Responsible for handling gameplay UI Logic and events.

The documentation for this class was generated from the following file:

- Project/Assets/Scripts/UI/GameplayUIManager.cs

5.7 TurnBased.Grids.GridData Class Reference

Properties

- int **RowsCount** `[getset]`
- int **ColumnsCount** `[getset]`
- float **CellSize** `[getset]`
- List< Unit > **CurrentUnits** `[getset]`

The documentation for this class was generated from the following file:

- Project/Assets/Scripts/Grids/GridData.cs

5.8 TurnBased.Grids.GridRenderer Class Reference

Responsible for controlling cells and rendering different events.

Inherits MonoBehaviour.

Public Member Functions

- void **InitializeGridCells** (GridData data)

  Initializes the cells based on the given grid data.
- void **ShowCellsToChooseForUnits** (IEnumerable< Unit > units, bool resetAll=true)

  Highlight the cells that have the given units.
- void **HighlightCellWithUnit** (Unit unit)
- void **ResetAll** ()

  Resets all cells.
- void **ShowActionsForUnit** (Unit unit)

  Shows available action for the given unit.
- Cell **GetCellAtPos** (Vector2Int pos)

  Gets cell at the given position.
- void **ShowExplosion** (Vector2Int position, Action onComplete)

  Shows explosion at the given cell position.
- void **ShowExplosion** (Cell cell, Action onComplete)

  Shows explosion at the given cell.
- bool **IsOutOfBounds** (Vector2Int position)

  Checks if the position is out of grid bounds.
- bool **IsEmptyAtPos** (Vector2Int position)

  Checks if the cell at the given position is empty.

Events

- Action< Cell > **CellSelected**

  Invoked when a cell is selected.

5.8.1 Detailed Description

Responsible for controlling cells and rendering different events.

The documentation for this class was generated from the following file:

- Project/Assets/Scripts/Grids/GridRenderer.cs

5.9 TurnBased.UI.HealthBar Class Reference

Shows the health for the given unit.

Inherits MonoBehaviour.

Public Member Functions

- void **Initialize** (Unit unit, float gridScale)

  Initializes a bar for the given unit and grid scale.

5.9.1 Detailed Description

Shows the health for the given unit.

The documentation for this class was generated from the following file:

- Project/Assets/Scripts/UI/HealthBar.cs

5.10 TurnBased.UI.HealthBarController Class Reference

Controls shown health bars and initializes it.

Inherits MonoBehaviour.

Public Member Functions

- void **InitializeHealthBarsForUnits** (List< Unit > units, float size)
  Initialize health bars for the given unit.

5.10.1 Detailed Description

Controls shown health bars and initializes it.

The documentation for this class was generated from the following file:

- Project/Assets/Scripts/UI/HealthBarController.cs

5.11 TurnBased.Player.HumanPlayer Class Reference

Gives input control to the human player once he gets the turn.

Inherits TurnBased.Player.Playerbase.

Public Member Functions

- **HumanPlayer** (GridRenderer gridRenderer, GameplayUIManager manager, List< Unit > units)
- override void StartTurn ()
  Show units that can be chosen and wait for cell selected event.
- override void EndTurn ()
  Clear Cell Selected Event and propagate end turn to the gameplay manager.

Properties

- override PlayerType **Type** `[get]`

Additional Inherited Members

5.11.1 Detailed Description

Gives input control to the human player once he gets the turn.

5.11.2 Member Function Documentation

5.11.2.1 EndTurn()

```
override void TurnBased.Player.HumanPlayer.EndTurn ( )  [virtual]
```

Clear Cell Selected Event and propagate end turn to the gameplay manager.

Reimplemented from TurnBased.Player.Playerbase.

5.11.2.2 StartTurn()

```
override void TurnBased.Player.HumanPlayer.StartTurn ( )  [virtual]
```

Show units that can be chosen and wait for cell selected event.

Reimplemented from TurnBased.Player.Playerbase.

The documentation for this class was generated from the following file:

- Project/Assets/Scripts/Player/HumanPlayer.cs

5.12 TurnBased.Data.InitialSituation Class Reference

A full game initial state contains different data that is used by the game to initialize first turn.

Inherits ScriptableObject.

Public Attributes

- Vector2Int **GridSize**

  The size of the grid in cells count per axis.
- float **CellSize**

  The Cell scale modifiers, this is to insure a static position in front of static camera.
- PlayerType[ ] **Players**

  The players available at the start of the situation.
- UnitInitialState[ ] **InitialUnits**

  Units, what player owns it and where it is.

5.12.1 Detailed Description

A full game initial state contains different data that is used by the game to initialize first turn.

The documentation for this class was generated from the following file:

- Project/Assets/Scripts/Data/InitialSituation.cs

5.13 TurnBased.Player.InputManager Class Reference

Inherits MonoBehaviour.

Properties

- HumanPlayer **CurrentPlayer** `[getset]`

The documentation for this class was generated from the following file:

- Project/Assets/Scripts/Player/InputManager.cs

5.14 TurnBased.UI.MainMenuUIManager Class Reference

Inherits MonoBehaviour.

Public Member Functions

- void **ChosenSituationChanged** (InitialSituation initialSituation)
- void **OnStartGameButtonPressed** ()

The documentation for this class was generated from the following file:

- Project/Assets/Scripts/UI/MainMenuUIManager.cs

5.15 TurnBased.Player.Playerbase Class Reference

Base class for player, controls turn loop once it got the turn.

Inherited by TurnBased.Player.AIPlayer, and TurnBased.Player.HumanPlayer.

Public Member Functions

- **Playerbase** (List< Unit > units, GameplayUIManager manager, GridRenderer gridRenderer)

  Initializes the player and adds the listener so the player knows when a unit is dead from its side.
- virtual void StartTurn ()

  Starts the player turn, resets its unit energy and gets the UI ready based on its type.
- virtual void EndTurn ()

  Ends Player turn and gives control to the GameplayManager to give control to the next player.

Protected Attributes

- List< Unit > **m_myUnits**
- GameplayUIManager **m_gameplayUIManager**
- GridRenderer **m_gridRenderer**

Properties

- int **Id** `[getset]`

  The index of the player in GameplayManager.
- abstract PlayerType **Type** `[get]`
- IEnumerable< Unit > **UnitsWithEnergy** `[get]`

Events

- Action **TurnEnded**

  Invoked when the player has ended the turn.

5.15.1 Detailed Description

Base class for player, controls turn loop once it got the turn.

5.15.2 Member Function Documentation

5.15.2.1 EndTurn()

```
virtual void TurnBased.Player.Playerbase.EndTurn ( )  [virtual]
```

Ends Player turn and gives control to the GameplayManager to give control to the next player.

Reimplemented in TurnBased.Player.HumanPlayer.

5.15.2.2 StartTurn()

```
virtual void TurnBased.Player.Playerbase.StartTurn ( )  [virtual]
```

Starts the player turn, resets its unit energy and gets the UI ready based on its type.

Reimplemented in TurnBased.Player.AIPlayer, and TurnBased.Player.HumanPlayer.

The documentation for this class was generated from the following file:

- Project/Assets/Scripts/Player/Playerbase.cs

5.16 TurnBased.Utils.Singleton< T > Class Template Reference

A utility class that supports singleton design pattern, used for cross scene interactions.

Inherits MonoBehaviour.

Public Member Functions

- virtual void **Awake** ()

Properties

- static T **Instance** `[get]`
- static T **SafeInstance** `[get]`

5.16.1 Detailed Description

A utility class that supports singleton design pattern, used for cross scene interactions.

Type Constraints

T* : *Component

The documentation for this class was generated from the following file:

- Project/Assets/Scripts/Utils/Singleton.cs

5.17 TurnBased.UI.SituationItemUI Class Reference

Inherits MonoBehaviour.

Public Member Functions

- void **Initialize** (InitialSituation situation, bool initState)

The documentation for this class was generated from the following file:

- Project/Assets/Scripts/UI/SituationItemUI.cs

5.18 TurnBased.UI.SituationsLoader Class Reference

Inherits MonoBehaviour.

The documentation for this class was generated from the following file:

- Project/Assets/Scripts/UI/SituationsLoader.cs

5.19 TurnBased.UI.SkipButton Class Reference

Inherits MonoBehaviour.

Public Member Functions

- void **ChangeText** (string text)
- void **ChangeButtonState** (bool isInteractable)

The documentation for this class was generated from the following file:

- Project/Assets/Scripts/UI/SkipButton.cs

5.20 TurnBased.Player.SmartAIPlayer Class Reference

A Smarter AI player with deterministic moves instead of random ones.

Inherits TurnBased.Player.AIPlayer.

Public Member Functions

- **SmartAIPlayer** (List< Unit > units, GameplayUIManager manager, GridData data, GridRenderer grid↩ Renderer)

Protected Member Functions

- override IEnumerator TurnCoroutine ()

  Controls Turn behavior.

Additional Inherited Members

5.20.1 Detailed Description

A Smarter AI player with deterministic moves instead of random ones.

5.20.2 Member Function Documentation

5.20.2.1 TurnCoroutine()

```
override IEnumerator TurnBased.Player.SmartAIPlayer.TurnCoroutine ( )  [protected], [virtual]
```

Controls Turn behavior.

Reimplemented from TurnBased.Player.AIPlayer.

The documentation for this class was generated from the following file:

- Project/Assets/Scripts/Player/SmartAIPlayer.cs

5.21 TurnBased.Gameplay.Unit Class Reference

Controls Unit runtime data, current health and available moves.

Inherits MonoBehaviour.

Public Member Functions

- void **SetInitialState** (UnitSetup setup)
- void **InitializeUnitAtCell** (Cell cell, float scale, Vector3 rotation)
- void **ResetEnergy** ()
- void **MoveUnitFromTo** (Cell currentCell, Cell targetCell, Action onComplete)
- void **AttackUnit** (Unit targetUnit)
- void **TakeDamage** (int damage)

Properties

- Vector2Int **AttackRange** `[get]`
- Vector2Int **Energy** `[get]`
- float **HealthPercent** `[get]`
- bool **UnitAttacked** `[get]`
- Vector2Int **CurrentCellPosition** `[getset]`
- int **OwningPlayer** `[getset]`

Events

- Action< Action > **HealthChanged**
- Action **UnitDead**

5.21.1 Detailed Description

Controls Unit runtime data, current health and available moves.

The documentation for this class was generated from the following file:

- Project/Assets/Scripts/Gameplay/Unit.cs

5.22 TurnBased.Data.UnitInitialState Struct Reference

Contains the data needed for a unit in initial situation.

Public Attributes

- int **OwningPlayer**
- UnitType **Type**
- Vector2Int **Position**

5.22.1 Detailed Description

Contains the data needed for a unit in initial situation.

The documentation for this struct was generated from the following file:

- Project/Assets/Scripts/Data/InitialSituation.cs

5.23 TurnBased.Data.UnitSetup Class Reference

Controls unit data setup, used by UnitsFactory to initialize units on the grid.

Inherits ScriptableObject.

Public Attributes

- Unit **Prefab**
- UnitType **Type**
- string **Name**
- int **Health**
- int **Damage**
- Vector2Int **AttackRange**
- Vector2Int **MoveRange**

5.23.1 Detailed Description

Controls unit data setup, used by UnitsFactory to initialize units on the grid.

The documentation for this class was generated from the following file:

- Project/Assets/Scripts/Data/UnitSetup.cs

5.24 TurnBased.Gameplay.UnitsFactory Class Reference

Inherits MonoBehaviour.

Public Member Functions

- bool **InitializeFactory** ()
- bool **TryInitailizeUnitWithState** (UnitInitialState state, out Unit unit)

The documentation for this class was generated from the following file:

- Project/Assets/Scripts/Gameplay/UnitsFactory.cs

Index