Namespace Trellis.Core

Classes

Item

Represents an immutable item with an identifier, name, and description.

Passage

Represents a named sequence of steps and associated links within a passage-based workflow or narrative.

PassageLink

Represents a hyperlink to a passage, including its display name, link text, and an indicator of whether the link is broken.

PassageStep

Represents a single step in a passage, including its type and associated value.

TrellisMacro

Represents a macro within the Trellis interactive storytelling framework, including its type and associated arguments.

Enums

PassageStepType

Specifies the type of step within a passage, such as displaying content or executing a macro.

Class Item

Namespace: Trellis.Core

Assembly: Trellis.dll

Represents an immutable item with an identifier, name, and description.

```
public sealed record Item : IEquatable<Item>
```

Inheritance

object ← Item

Implements

IEquatable<Item>

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

Item(string, string, string)

Represents an immutable item with an identifier, name, and description.

```
public Item(string Id, string Name, string Description)
```

Parameters

`Id` string

The unique identifier for the item. Cannot be null.

`Name` string

The display name of the item. Cannot be null.

`Description` string

A textual description of the item. Cannot be null.

Properties

Description

A textual description of the item. Cannot be null.

```
public string Description { get; init; }
```

Property Value

string

Id

The unique identifier for the item. Cannot be null.

```
public string Id { get; init; }
```

Property Value

string

Name

The display name of the item. Cannot be null.

```
public string Name { get; init; }
```

Property Value

string

Class Passage

Namespace: Trellis.Core

Assembly: Trellis.dll

Represents a named sequence of steps and associated links within a passage-based workflow or narrative.

```
public sealed record Passage : IEquatable<Passage>
```

Inheritance

object ← Passage

Implements

IEquatable<Passage>

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.ReferenceEquals(object, object) , object.ToString()

Remarks

Use the Passage record to model a discrete section within a larger interactive flow, such as in interactive fiction, tutorials, or guided processes. Each passage contains a set of steps and may provide links to other passages, enabling branching or navigation.

Constructors

Passage(string, IReadOnlyList<PassageStep>, IReadOnlyList<PassageLink>)

Represents a named sequence of steps and associated links within a passage-based workflow or narrative.

```
public Passage(string Name, IReadOnlyList<PassageStep> Steps, IReadOnlyList<PassageLink>
Links)
```

Parameters

`Name` string

The unique name that identifies the passage.

`Steps` IReadOnlyList<PassageStep>

The ordered collection of steps that define the content or actions within the passage.

`Links` IReadOnlyList<PassageLink>

The collection of links that connect this passage to other passages or destinations.

Remarks

Use the Passage record to model a discrete section within a larger interactive flow, such as in interactive fiction, tutorials, or guided processes. Each passage contains a set of steps and may provide links to other passages, enabling branching or navigation.

Properties

Links

The collection of links that connect this passage to other passages or destinations.

```
public IReadOnlyList<PassageLink> Links { get; init; }
```

Property Value

IReadOnlyList<PassageLink>

Name

The unique name that identifies the passage.

```
public string Name { get; init; }
```

Property Value

string

Steps

The ordered collection of steps that define the content or actions within the passage.

```
public IReadOnlyList<PassageStep> Steps { get; init; }
```

Property Value

IReadOnlyList<PassageStep>

Class PassageLink

Namespace: Trellis.Core

Assembly: Trellis.dll

Represents a hyperlink to a passage, including its display name, link text, and an indicator of whether the link is broken.

```
public sealed record PassageLink : IEquatable<PassageLink>
```

Inheritance

object ← PassageLink

Implements

IEquatable<PassageLink>

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

PassageLink(string, string, bool)

Represents a hyperlink to a passage, including its display name, link text, and an indicator of whether the link is broken.

```
public PassageLink(string Name, string Text, bool Broken = false)
```

Parameters

`Name` string

The unique name or identifier of the target passage.

`Text` string

The text to display for the link.

`Broken` bool

Indicates whether the link is considered broken. Set to true if the link does not resolve to a valid passage; otherwise, false.

Properties

Broken

Indicates whether the link is considered broken. Set to true if the link does not resolve to a valid passage; otherwise, false.

```
public bool Broken { get; init; }
```

Property Value

bool

Name

The unique name or identifier of the target passage.

```
public string Name { get; init; }
```

Property Value

string

Text

The text to display for the link.

```
public string Text { get; init; }
```

Property Value

string

Class PassageStep

Namespace: Trellis.Core

Assembly: Trellis.dll

Represents a single step in a passage, including its type and associated value.

```
public sealed record PassageStep : IEquatable<PassageStep>
```

Inheritance

object ← PassageStep

Implements

IEquatable<PassageStep>

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

PassageStep(PassageStepType, object?)

Represents a single step in a passage, including its type and associated value.

```
public PassageStep(PassageStepType StepType, object? Value)
```

Parameters

`StepType` PassageStepType

The type of the passage step, indicating the action or category of this step.

`Value` object

The value associated with the step. The expected type and meaning depend on the specified step type. May be null if the step does not require an associated value.

Properties

StepType

The type of the passage step, indicating the action or category of this step.

```
public PassageStepType StepType { get; init; }
```

Property Value

PassageStepType

Value

The value associated with the step. The expected type and meaning depend on the specified step type. May be null if the step does not require an associated value.

```
public object? Value { get; init; }
```

Property Value

object

Enum PassageStepType

Namespace: Trellis.Core

Assembly: Trellis.dll

Specifies the type of step within a passage, such as displaying content or executing a macro.

```
public enum PassageStepType
```

Fields

`display = 0`

`macro = 1`

Class TrellisMacro

Namespace: Trellis.Core

Assembly: Trellis.dll

Represents a macro within the Trellis interactive storytelling framework, including its type and associated arguments.

```
public sealed record TrellisMacro : IEquatable<TrellisMacro>
```

Inheritance

object ← TrellisMacro

Implements

IEquatable<TrellisMacro>

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

TrellisMacro(string, string[])

Represents a macro within the Trellis interactive storytelling framework, including its type and associated arguments.

```
public TrellisMacro(string macroType, string[] macroArgs)
```

Parameters

`macroType` string

Macro type

`macroArgs` string[]

Macro arguments

Properties

macroArgs

Macro arguments

```
public string[] macroArgs { get; init; }
```

Property Value

string[]

macroType

Macro type

```
public string macroType { get; init; }
```

Property Value

string

Namespace Trellis.Engine

Classes

TrellisEngine

`TrellisEngine` is the main class responsible for managing the state and progression of an interactive story.

Class TrellisEngine

Namespace: Trellis.Engine

Assembly: Trellis.dll

`TrellisEngine` is the main class responsible for managing the state and progression of an interactive story.

```
public class TrellisEngine
```

Inheritance

object ← TrellisEngine

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

TrellisEngine(string, string)

Initializes a new instance of the TrellisEngine class using the specified story data path and story type.

```
public TrellisEngine(string storyDataPath, string storyType)
```

Parameters

`storyDataPath` string

The file path to the story data to be loaded. This should point to a valid story data file compatible with the specified story type.

`storyType` string

The type of story data to load. This determines how the story data is interpreted and processed.

Exceptions

ArgumentException

Thrown if the story data cannot be loaded for the specified story type, or if the provided story data path or type is invalid.

Methods

GetChoiceCount()

Gets the number of available navigation choices.

```
public int GetChoiceCount()
```

Returns

int

The total number of navigation choices currently available. Returns 0 if no choices are present.

GetCurrentPassage()

Gets the current passage in the navigation sequence.

```
public Passage GetCurrentPassage()
```

Returns

Passage

The current Passage instance representing the active passage. Returns null if there is no current passage.

GetCurrentStep()

Gets the current step in the navigation sequence.

```
public PassageStep GetCurrentStep()
```

Returns

PassageStep

The current PassageStep representing the user's position in the navigation. Returns null if there is no current step.

GetPassageLinks()

Retrieves a list of links associated with the current passage.

```
public List<PassageLink> GetPassageLinks()
```

Returns

List<PassageLink>

A list of PassageLink objects representing the available links in the current passage. The list will be empty if the passage contains no links.

IsFirstStep()

Determines whether the current navigation position is at the first step.

```
public bool IsFirstStep()
```

Returns

bool

true if the current step is the first in the navigation sequence; otherwise, false.

IsLastStep()

Determines whether the current step is the last step in the navigation sequence.

```
public bool IsLastStep()
```

Returns

bool

true if the current step is the last step; otherwise, false.

NavigateTo(string)

Attempts to navigate to the passage with the specified name.

```
public bool NavigateTo(string passageName)
```

Parameters

`passageName` string

The name of the passage to navigate to. Cannot be null or empty.

Returns

bool

true if navigation to the specified passage was successful; otherwise, false.

Reset()

Resets the state of the object to its initial conditions.

```
public void Reset()
```

Remarks

Call this method to clear any navigation history and prepare the object for reuse. This is typically used to reinitialize the object without creating a new instance.

Step()

Advances the navigation to the next step in the sequence.

```
public bool Step()
```

Returns

bool

true if the navigation successfully advanced to the next step; otherwise, false.

Namespace Trellis.Loader

Classes

TwineLinkDto

TwinePassageDto

TwineStoryDto

Class TwineLinkDto

Namespace: Trellis.Loader

Assembly: Trellis.dll

```
public sealed record TwineLinkDto : IEquatable<TwineLinkDto>
```

Inheritance

object ← TwineLinkDto

Implements

IEquatable<TwineLinkDto>

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

TwineLinkDto(string, string, bool)

```
public TwineLinkDto(string name, string text, bool broken = false)
```

Parameters

`name` string

`text` string

`broken` bool

Properties

broken

```
public bool broken { get; init; }
```

Property Value

bool

name

```
public string name { get; init; }
```

Property Value

string

text

```
public string text { get; init; }
```

Property Value

string

Class TwinePassageDto

Namespace: Trellis.Loader

Assembly: Trellis.dll

```
public sealed record TwinePassageDto : IEquatable<TwinePassageDto>
```

Inheritance

object ← TwinePassageDto

Implements

IEquatable<TwinePassageDto>

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

TwinePassageDto(string, string, List<TwineLinkDto>, string, Dictionary<string, string>)

```
public TwinePassageDto(string name, string text, List<TwineLinkDto> links, string pid,
Dictionary<string, string> position)
```

Parameters

`name` string

`text` string

`links` List<TwineLinkDto>

`pid` string

`position` Dictionary<string, string>

Properties

links

```
public List<TwineLinkDto> links { get; init; }
```

Property Value

List<TwineLinkDto>

name

```
public string name { get; init; }
```

Property Value

string

pid

```
public string pid { get; init; }
```

Property Value

string

position

```
public Dictionary<string, string> position { get; init; }
```

Property Value

Dictionary<string, string>

text

```
public string text { get; init; }
```

Property Value

string

Class TwineStoryDto

Namespace: Trellis.Loader

Assembly: Trellis.dll

```
public sealed record TwineStoryDto : IEquatable<TwineStoryDto>
```

Inheritance

object ← TwineStoryDto

Implements

IEquatable<TwineStoryDto>

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

TwineStoryDto(List<TwinePassageDto>)

```
public TwineStoryDto(List<TwinePassageDto> passages)
```

Parameters

`passages` List<TwinePassageDto>

Properties

passages

```
public List<TwinePassageDto> passages { get; init; }
```

Property Value

List<TwinePassageDto>

Namespace Trellis.Test

Classes

FileTest

Class FileTest

Namespace: Trellis.Test

Assembly: Trellis.dll

```
public class FileTest
```

Inheritance

object ← FileTest

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Namespace Trellis.Text

Classes

Tokenizer

Class Tokenizer

Namespace: Trellis.Text

Assembly: Trellis.dll

```
public static class Tokenizer
```

Inheritance

object ← Tokenizer

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Methods

Tokenize(string)

```
public static string[] Tokenize(string text)
```

Parameters

`text` string

Returns

string[]