Getting Started

Adding the NuGet pacakge to your project

You need to pull BP.AdventureFramework into your project. The easiest way to do this is to add the NuGet package. The latest package and installation instructions are available here (https://github.com/benpollarduk/BP.AdventureFramework/pkgs/nuget/BP.AdventureFramework).

First Game

Once the package has been installed it's time to jump in and start building your first game.

```
// create the player. this is the character the user plays as
var player = new PlayableCharacter("Dave", "A young boy on a quest to find the meaning of life.");

/// create region maker. the region maker simplifies creating in game regions. a region contains a series of rooms
var regionMaker = new RegionMaker("Mountain", "An imposing volcano just East of town.")
{
    // add a room to the region at position x 0, y 0, z 0
    [0, 0, 0] = new Room("Cavern", "A dark cavern set in to the base of the mountain.")
};

// create overworld maker. the overworld maker simplifies creating in game overworlds. an overworld contains a series or regions
var overworldMaker = new OverworldMaker("Daves World", "An ancient kingdom.", regionMaker);

// create the callback for generating new instances of the game
// - the title of the game
// - an introduction to the game, displayed at the start
// - a description, displayed on the about screen
// - a callback that provides a new instance of the games overworld
// - a callback that provides a new instance of the player
// - a callback that determines if the game is complete, checked every cycle of the game
// - a callback that determines if it's game over, checked every cycle of the game
var gameCreator = Game.Create(
    "Dave",
    "Dave awakes to find himself in a cavern...",
    "A very low budget adventure.",
    x => overworldMaker.Make(),
    () => player,
    x => EndCheckResult.NotEnded,
    x => EndCheckResult.NotEnded);

// begin the execution of the game
Game.Execute(gameCreator);
```

Namespace BP.AdventureFramework.Assets

Classes

ConditionalDescription (BP.AdventureFramework.Assets.ConditionalDescription.html)

Represents a conditional description of an object.

Description (BP.AdventureFramework.Assets.Description.html)

Represents a description of an object.

ExaminableObject (BP.AdventureFramework.Assets.ExaminableObject.html)

Represents an object that can be examined.

ExaminationResult (BP.AdventureFramework.Assets.ExaminationResult.html)

Represents the result of an examination.

Identifier (BP.AdventureFramework.Assets.Identifier.html)

Provides a class that can be used as an identifier.

Item (BP.AdventureFramework.Assets.Item.html)

Represents an item that can be used within the game.

Structs

Size (BP.AdventureFramework.Assets.Size.html)

Represents a size.

Interfaces

IExaminable (BP.AdventureFramework.Assets.IExaminable.html)

Represents any object that is examinable.

IPlayerVisible (BP.AdventureFramework.Assets.IPlayerVisible.html)

Represents any object that is visible to a player.

Delegates

ExaminationCallback (BP.AdventureFramework.Assets.ExaminationCallback.html)

Represents the callback for examinations.

- **BP.AdventureFramework.
  Assets
  (BP.AdventureFramework.Assets**

  ConditionalDescription
  (BP.AdventureFramework.Assets.C
  Description
  (BP.AdventureFramework.Assets.D
  ExaminableObject
  (BP.AdventureFramework.Assets.E
  ExaminationCallback
  (BP.AdventureFramework.Assets.E
  ExaminationResult
  (BP.AdventureFramework.Assets.E
  IExaminable
  (BP.AdventureFramework.Assets.IE
  IPlayerVisible
  (BP.AdventureFramework.Assets.IF
  Identifier
  (BP.AdventureFramework.Assets.Id
  Item
  (BP.AdventureFramework.Assets.It
  Size
  (BP.AdventureFramework.Assets.S

+ **BP.AdventureFramework.
  Assets.Characters
  (BP.AdventureFramework.Asse**

+ **BP.AdventureFramework.
  Assets.Interaction
  (BP.AdventureFramework.Asse**

+ **BP.AdventureFramework.
  Assets.Locations
  (BP.AdventureFramework.Asse**

+ **BP.AdventureFramework.
  Commands
  (BP.AdventureFramework.Com**

+ **BP.AdventureFramework.
  Conversations**