

Getting Started

Adding the NuGet package to your project

▼ Filter by title

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>).

Getting Started (getting-started.html)

Executing a Game (executing-a-game.html)

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

+ **Locations**

Items (items.html)

+ **Characters**

Conditional Descriptions (conditional-descriptions.html)

Commands (commands.html)

Frame Builders (frame-builders.html)

End Conditions (end-conditions.html)

```
// 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.");
```

Getting Started (getting-started.html)

```
// 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.");
```

Executing a Game (executing-a-game.html)

Locations

+ **Locations**

```
// create overworld maker. the overworld maker simplifies creating in game overworlds. an overworld contains a series of regions
```

```
var overworldMaker = new OverworldMaker("Daves World", "An ancient kingdom.", regionMaker);
```

+ **Characters**

Conditional Descriptions for generating new instances of the game

```
// - title of the game
```

```
// - an introduction to the game, displayed at the start
```

```
// - about the game, 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
```

Frame Builders (frame-builders.html)

```
// - a callback that determines if it's game over, checked every cycle of the game
```

```
var gameCreator = Game.Create(
```

```
    "The life of Dave",
```

```
    "Dave awakes to find himself in a cavern...",
```

End Conditions (end-conditions.html)

```
    "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

▼ Filter by title

Classes

- BP.AdventureFramework.

Assets

ConditionalDescription

(BP.AdventureFramework.Assets.ConditionalDescription.html)

ConditionalDescription

Represents a conditional description of an object.

(BP.AdventureFramework.Assets.C

Description

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

ExaminableObject

Represents a description of an object.

(BP.AdventureFramework.Assets.E

ExaminationCallback

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

ExaminationResult

Represents an object that can be examined.

(BP.AdventureFramework.Assets.E

IExaminable

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

IPlayerVisible

Represents the result of an examination.

(BP.AdventureFramework.Assets.II

Identifier

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

Item

Provides a class that can be used as an identifier.

(BP.AdventureFramework.Assets.It

Size

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

+ BP.AdventureFramework.

Assets.Characters

Sprites (BP.AdventureFramework.Assets

+ BP.AdventureFramework.

Assets.Interaction

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

(BP.AdventureFramework.Assets

Represents a size.

+ BP.AdventureFramework.

Interfaces

(BP.AdventureFramework.Assets

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

Commands

Represents any object that is examinable.

(BP.AdventureFramework.Corr

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

Conversations

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.II

IPlayerVisible

(BP.AdventureFramework.Assets.II

Identifier

(BP.AdventureFramework.Assets.Ic

Item

(BP.AdventureFramework.Assets.It

Size

(BP.AdventureFramework.Assets.S

+ BP.AdventureFramework.

Assets.Characters

(BP.AdventureFramework.Assets)

+ BP.AdventureFramework.

Assets.Interaction

(BP.AdventureFramework.Assets)

+ BP.AdventureFramework.

Assets.Locations

(BP.AdventureFramework.Assets)

+ BP.AdventureFramework.

Commands

(BP.AdventureFramework.Commands)

+ BP.AdventureFramework.

Conversations