

# Unity plugin

This project will help to integrate bHaptics' haptic devices into Unity environments.

## Prerequisite

- bHaptics Player needs to be installed (Windows)
  - The app can be found at  
bHaptics webpage: <http://www.bhaptics.com>

## How to install

### Download from the Unity Asset Store

- <https://www.assetstore.unity3d.com/en/#!/content/76647>

### Download the package file, then import it into a Unity Project

- <https://github.com/bhaptics/tac-sharp/releases>

### Clone from the github repository, then open it in Unity

```
git clone https://github.com/bhaptics/tac-sharp.git
```

## Tutorial Videos

- [bHaptics Unity plugin With Code](#)
- [bHaptics Designer To Unity - Tactosy](#)
- [bHaptics Designer To Unity - Tactal](#)
- [bHaptics Designer To Unity - Tactot](#)

## How to use

- Default Test Scene

```
>Go to Assets > bHapticsManager > Examples > open sample.scene  
Select the [bHaptics Manager] Prefab in the scene.  
Some example feedback effects are automatically loaded, ready for testing.  
You can check each feedback effect by pushing the corresponding button while  
playing in the editor.
```

- To apply to your own project, just add the [bHaptics Manager] Prefab to your scene.
- Import namespaces into classes that will be using haptic feedback.

```
using Bhaptics.Tac;  
using Bhaptics.Tac.Unity;
```

- Get the HapticPlayer reference

```
private IHapticPlayer HapticPlayer;

void Start ()
{
    HapticPlayer = BhapticsManager.HapticPlayer;
}
```

- Apply more feedback effects: with .tact file

You can create Tact feedback effects via <https://designer.bhaptics.com>. The .tact files generated by the designer are timeline based haptic feedback effect files. You can find more details of the designer [here](#).

- Play feedback effects in C# Script: List of PathPoints

```
var motorCount = 2; // number of motors for PathPoint
List<PathPoint> pathPoints = new List<PathPoint>
{
    new PathPoint(x_position, y_position, intensity)
    /* x_position, y_position are floats in
       normalized value (0.0f to 1.0f) beginning from upper left of the device.*/
    , new PathPoint(x_position, y_position, intensity, motorCount)
};
HapticPlayer.Submit("Point", PositionType.Right, pathPoints, duration);
/* duration is a positive integer in milliseconds */
```

- Play feedback effects in C# Script: DotPoints

```
HapticPlayer.Submit("space", PositionType.Head, new DotPoint(3, 100), 1000);
```

- Play feedback effects in C# Script: Array of Bytes

```
byte[] bytes =
{
    0, 0, 0, 0, 0,
    0, 0, 0, 0, 0,
    0, 0, 100, 100, 0,
    0, 0, 0, 0, 0
};
/* Values should be an int (0~100)
/* Each number is the intensity of the point*/
HapticPlayer.Submit("Bytes", PositionType.Right, bytes);
```

- Play registered .tact feedback effects using file names
  - The plugin will automatically register tact files in the specified pathPrefix in [bhaptics Manager],

using their file name as a key.

```
/* Just play feedback of Fireball.tact file */
HapticPlayer.SubmitRegistered("Fireball");

/* play feedback of RifleImpact.tact file with transformation */
HapticPlayer.SubmitRegistered("RifleImpact", new TransformOption(0.5f, .5f)););
```

- TurnOff Signal

```
/* Turn off all current Haptic feedback effects */
HapticPlayer.TurnOff();
/* Turn off the specified Haptic feedback effect using its Key string */
HapticPlayer.TurnOff("Fireball");
```

- Check whether some feedback is playing or not

```
/* Return the bool whether 'Fireball' is playing */
bool isFireballFeedbackPlaying = HapticPlayer.IsPlaying("Fireball");
/* Return the bool whether any feedback is playing */
bool isAnyFeedbackPlaying = HapticPlayer.IsPlaying();
```

## Options in [bHapticsManager]

### visualizeFeedbacks

- Enable/disable visualization of haptic feedback

### pathPrefix

- Define the path of the .tact feedback files
- Default Path : Assets/bHapticsManager/Feedbacks/

### useStreamingPath

- Specifies to use the StreamingPath for getting feedback files rather than the pathPrefix.
- Why does this matter?  
Unity does not automatically copy .tact files to the destination build path.  
To avoid this problem, unity provides 'StreamingAssets' folder for automatic loading during building.
- If this option is checked, [bHapticsManger] reads feedback files from the 'StreamingAssets/{pathPrefix}' folder.
- If this option is not checked, you need to manually copy feedback files to the destination folder.  
The default destination folder path for windows will be '{pathPrefix}'

## Notes

- Migration from 1.0.3 to 1.0.4

```
var hapticPlayer = FindObjectOfType<BhapticsManager>().HapticPlayer();
```

To

```
var hapticPlayer = BhapticsManager.HapticPlayer;
```

- Migration from 1.0.4 to 1.1.0

Copyright 2017 bHaptics Inc.