Heart Rate/Beat Monitor

Documentation & API Reference Manual for Unity3D
v1.5

Firstly, thank you for purchasing the Heart Rate/Beat Monitor. This C# script is very easy to use and can be implemented in just a few clicks. The script should also be compatible and performance friendly with all of the platforms supported by Unity3D.

We intend to bring you updates and great future assets for your Unity3D projects.

If you have any questions, suggestions, comments or of course feature requests please contact us at support@sketchworkdev.com where we will be able to help you. Our support forum will be arriving soon, but for now please use our support email address above.

Table of Contents

Installation Unity 3.5

This asset will automatically install the scripts, prefabs and examples for Unity 3.5 straight from the Asset Store.

Installation Unity 4+

This asset will automatically install the scripts, prefabs and examples for Unity 4+ straight from the Asset Store.

Removing the Example Projects

For both Unity 3.5 and Unity 4 installations to remove the example projects you can safely remove the following folder from inside Unity:

SWP_HeartRateMonitor / SWP_HeartRateMonitor (Example)/

Usage for a New Project

This asset is extremely easy to use. Please follow the following steps to get the component into a project.
Note: I have used HRM as an abbreviation for Heart Rate Monitor.

1) Create a new Scene.
2) Import the SWP_HeartRateMonitor package.
3) Add the "SWP_HeartRateMonitor / Prefabs / HeartMonitor" prefab to the new scene.
4) Click PLAY.
5) You should now see the HRM scrolling across the screen… Easy!!

Usage within an Existing Project

With this example we will be using the Angry Bots demonstration project that ships with all current versions of Unity3D. You can however use your own project instead if you like as the steps will be nearly identical.
Note: I have used HRM as an abbreviation for Heart Rate Monitor.

1) Import the SWP_HeartRateMonitor package to the project.
2) Load up the AngryBots scene from the root of your project.
3) Add the "SWP_HeartRateMonitor / Prefabs / HeartMonitor" prefab to the scene.
4) Create a new layer and call it Monitor (or what you want to call it).
5) Add a new camera (this will be used for your HRM) and set the following:

* Clear Flags	Don't Clear
* Culling Mask	Monitor (only monitor nothing else)
* Projection	Orthographic (can be either, but I use orthographic here)
* Size	10

6) Turn off the audio listener for the new camera.
7) On the Main Camera edit the culling mask and make sure Monitor is not selected.
8) Run the project. The HRM will be running at full screen and following the screen.

We will now talk about how to place the HRM within your scene. For this example we will be putting it in the bottom right corner, but you can place it where you like.
Note: The transform X and Y may need tweaking depending on your screen size and resolution.

9) For the placement go to the HeartMonitor game object and set the following:

* Transform X	15
* Transform Y	-8
* Blip Size	0.15
* Blip Trail Start Size	0.1
* Blip Trail End Size	0
* Blip Monitor Width	5
* Blip Monitor Height	0.2

10) Run the project and this time the HRM will be displayed nicely in the bottom right corner.

Using a simple script you can connect the actual Heart Rate to the player health (Player / Health script / Health) so it gets faster when he is hit and maybe slow right down when he is about to die. When the player dies (Player / Health script / Dead) you can activate the Flat Line function to kill the heart beat completely.

API Reference

Use the API reference to refer to what each public parameter in the control does.

Beats per Minute (Integer Value)

This is what controls how fast of how slowly the heart rate is. The higher the value the more blips you get and the lower the value the less blips you get. This is "90" by default to indicate 90 BPM.

Flat Line (Boolean Value)

This controls if the heart is beating or not. When activated the line will not blip and will display a continuous straight line to indicate death. This is "False" by default.

Show Blip (Boolean Value)

This controls if the sphere that leads the line is visible. When set to "false" there will just be a line with no proceeding sphere. This is "True" by default.

Blip (Game Object)

This points to the default Blip Prefab. This value should not be changed.

Blip Size (Float Value)

This is the overall size of the blip sphere that precedes the line.

Blip Trail Start Size (Float Value)

This is the start trail size which is used straight after the blip sphere has passed.

Blip Trail End Size (Float Value)

This is the final trail size at the end of the line before it disappears.

Blip Monitor Width (Float Value)

This is the main width of the control. Do you want it to stretch the width of the screen or just fit in a part of your GUI. Change this value to suit how wide you want the control.

Blip Monitor Height (Float Value)

This is the main height of the control. Do you want it really high or just a small rise and fall? Change this value to suit how high you want the blip to go.

Enable Sound (Boolean Value)

This controls if you want the sound on or off (bypassed).

Heart Volume (Float Value)

This controls the volume of the heart/flatline sounds. This must be a value between 0 and 1 (with 1 being the loudest).

Heart 1 (Audio Clip Value)

This points to the initial heart beat SFX.

Heart 2 (Audio Clip Value)

This points to the second heart beat SFX.

Flatline (Audio Clip Value)

This points to the flatline SFX.

Notes for Mobile Build Optimisations

We suggest for mobile builds that you do switch off the Blip in the inspector and this will free 1 draw call reducing it to 1 to 2 draw calls. We are currently looking at adding further mobile build optimisations with the aim to reduce the number of draw calls down to only 1.

Version History

Version 1.5

Added support for Unity 3.5.
Updated support email details.

Version 1.4

Added heart beat sound effects.
Added flatline sound effect.
Added “Enable Sound” control.
Added “Heart Volume” control.

Version 1.3

Updated documentation to include mobile build optimisation notes.

Version 1.2

Updated documentation.

Version 1.1

Added full documentation to package.
Added more C# code comments to the main script.

Version 1.0

Initial Version.

Support and Additional Information

Once again, if you require any additional information, help or have a feature request please contact us at support@sketchworkdev.com. The support forum will be arriving soon, but for now please use our support email address above.