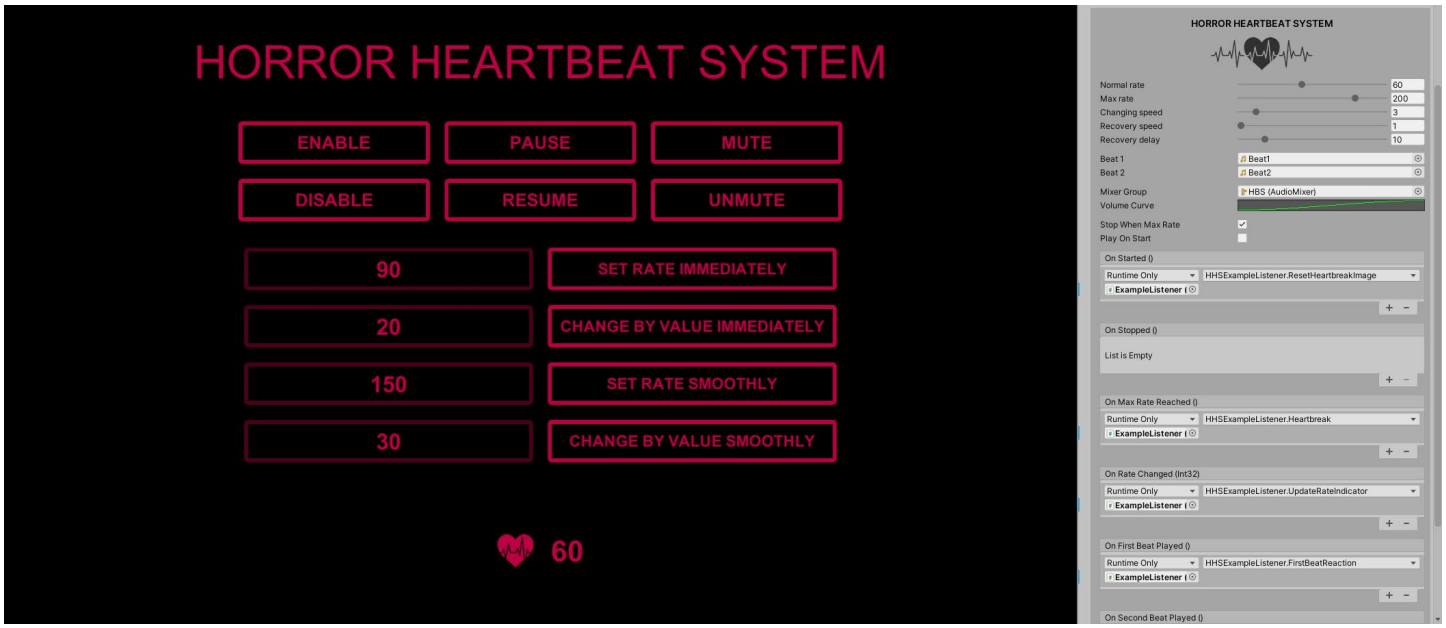
HORROR HEARTBEAT SYSTEM

HHS is a dynamic heartbeat system for your game.

You can influence the heart rate of your character through the API methods of the system.

Also, the system generates several events, you can subscribe to them and use them in the logic of your game. For example, when you reach the threshold heart rate, your character may die.

You can use the scene to familiarize yourself with the system (HHS/DemoScene).

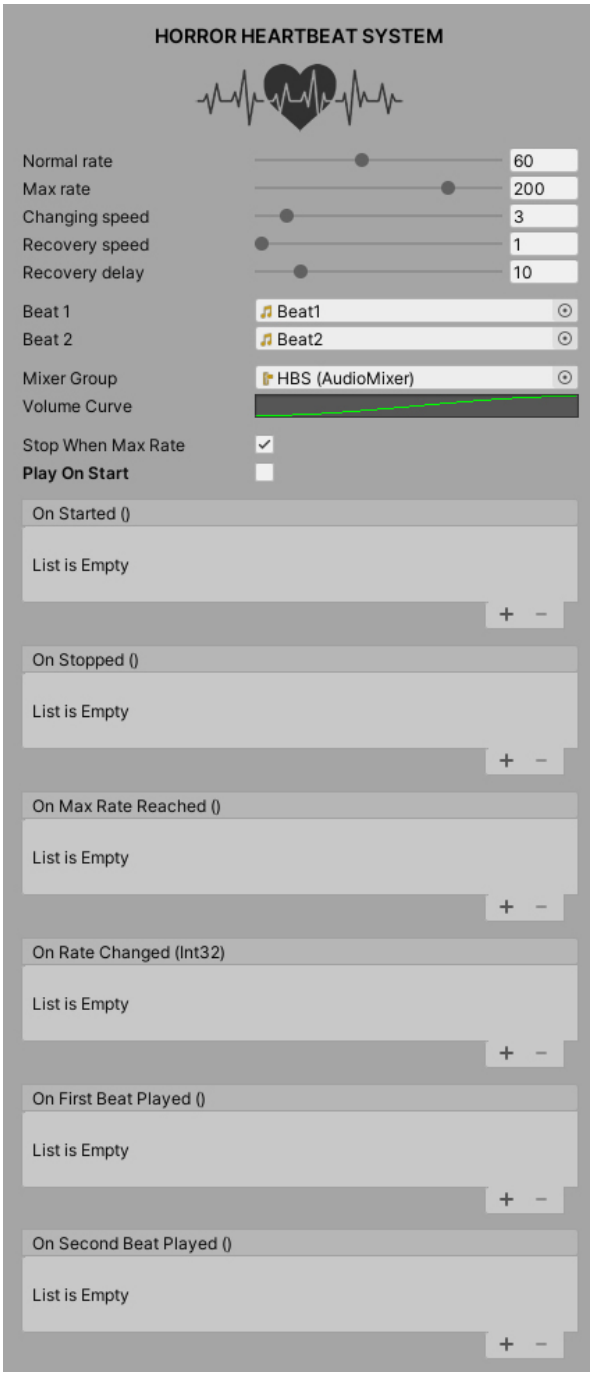


Quick start

Drag and drop the prefab HHS from the folder HHS/Prefabs.

Check the "Play on start" box. Set the normal heart rate and press the Play button. You will hear a heartbeat at a given frequency. But in order to influence its rate, - you need to use API-methods.

Settings



- **Normal Rate** - the normal heart rate for your character. When you start and restart the system, the heart will beat at this pace. It is set in the number of beats per second.
- **Max Rate** - the maximum possible heart rate. When it is reached, the system can stop its work or continue depending on the setting “Stop When Max Rate”, but the frequency cannot rise above it.
- **Changing Speed** – the rate of change in heart rate in the number of beat per second. Used when you call methods to smoothly change the heart rate.
- **Recovery Speed -** the rate of change in heart rate in the number of strokes per second. It is used when the automatic recovery of the heart rate of a character starts.
- **Recovery Speed -** the rate of change in heart rate in the number of strokes per second. It is used when the automatic recovery of the heart rate of a character starts.
- **Recovery Delay** - the time interval in seconds after which the recovery of your character’s heart rate begins automatically. It is counted from the moment the pulse stopped growing.
- **Beat 1 and Beat 2** - sound clips of your character’s heartbeats. You can replace from your own.
- **Mixer Group** - use to control the volume of heartbeat and add effects. In the demo scene, I use a reverb. You may not use a group mixer if your own sound clips already contain the necessary effects.
- **Volume Curve** - the volume of the heartbeat depends on its frequency. Using the curve, you can flexibly adjust the level of increase in the volume of the heartbeat. 0 on the curve fits to the volume at a normal heartbeat, and 1 - volume at the maximum heart rate.
- **Stop When Max Rate** - when checked, the system stops its operation as soon as the frequency reaches its maximum value.
- **Play On Start** - the system automatically starts when you start the game.

Under certain conditions, the system generates several UnityEvents that you can subscribe to.

1. **OnStarted** - triggered when the system starts.
2. **OnStopped -** triggered when the system stops
3. **OnMaxRateReached** - triggered when maximum heart rate is reached.
4. **OnRateChanged** - triggered every time the heart rate changes. Returns the current rate.
5. **OnFirstBeatPlayed -** triggered on every first heartbeat.
6. **OnFirstBeatPlayed -** triggered on every second heartbeat.

API

The system has the following public methods:

Enable - activates the system. The heart begins to beat at a frequency indicated in “Normal rate”.

Disable - deactivates the system. Stops all the processes of restoration and change of rate.

Mute - mute sound sources. All active processes for changing the reta continue.

Unmute - unmute sound sources.

Pause - pauses active processes. It should be used when pausing the game if you are not using timescale = 0.

Resume - resumes active processes after pause.

SetHeartRateImmediately(int rate) - Sets the heart rate according to the received value. Use this if something terrifying happens in your game.

ChangeHeartRateImmediatelyByValue(int value) - changes the current heart rate by the received value. For example, something creaked in your game, each such frightening event can increase the heart rate of your character by 5 (for example).

SetHeartRateSmoothly(int targetRate) - smoothly changes the heart rate to the received value.

ChangeHeartRateSmoothlyByValue(int rate) - smoothly changes the heart rate by the received value. For example, your character has entered a particularly dangerous area. You can use the previous method to smoothly raise your heartbeat to a specific rate. Or use this with a tedious periodicity until the player leaves the danger zone or dies of a heart attack.

int ***GetCurrentHeartRate***() – returns the current heart rate.

For all questions and suggestions for improving the system, write to the mail indicated on the asset page.