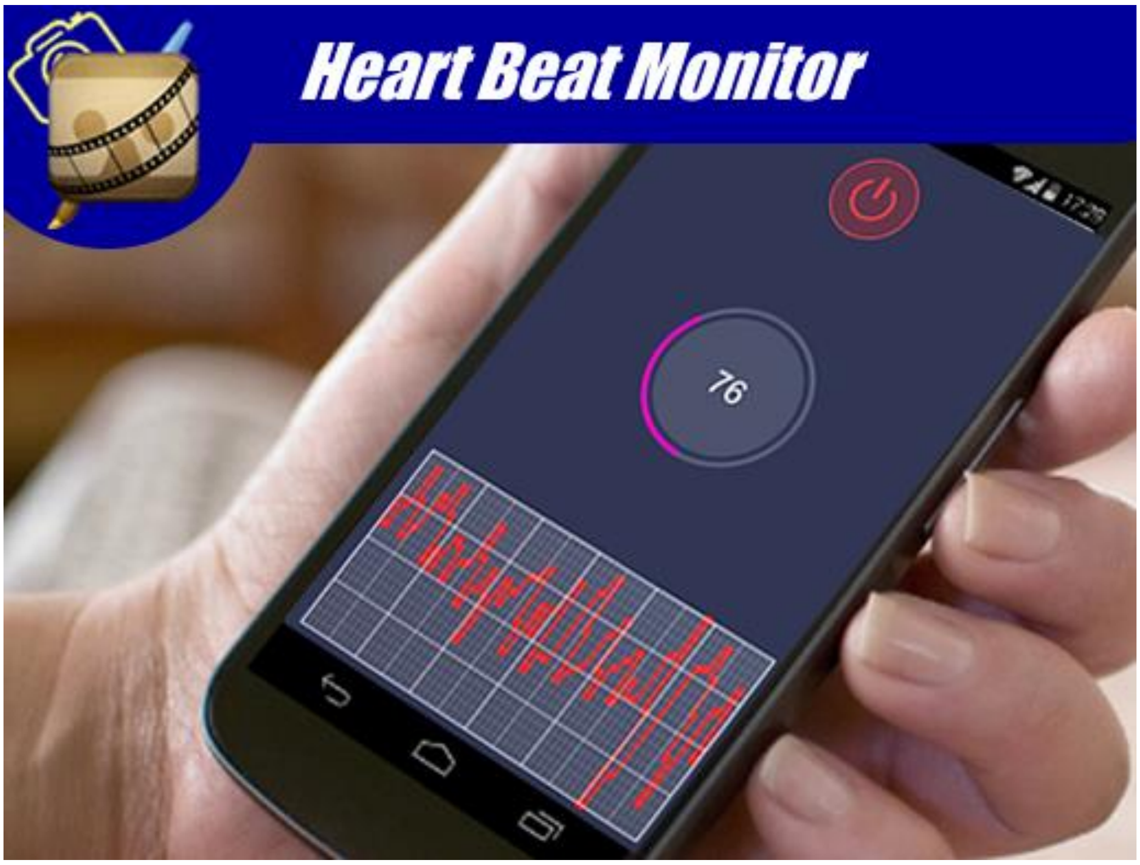


AUTHOR	**Michael Soler**
CONTACT	**michael.soler.beatty@gmail.com**
Unity Ver.	**2019.1**

Index

1.Description of the package.

With this package developers will have all they need to create apps to measure the heart rate using a camera (webcam, native Android, iOS camera, etc).

The heart beat rate is obtained through the small variations of brightness of the image when lens are covered by a finger. This is due to the amount of blood that changes at each beat. This information is stored in the script and analyzed using Fast Fourrier Transform. The maximum peak frequency represents the heart beat in real time.

The package contains all the textures, models and scripts shown in the video. For further questions, please contact michael.soler.beatty@gmail.com

2.About the FFT (Fast Fourrier Transform)

A fast Fourier transform (FFT) is an algorithm that computes the discrete Fourier transform (DFT) of a sequence. It converts a signal from its original domain (often time or space) to a representation in the frequency domain. The algorithm that we use in our scripts is the Cooley-Tukey method:

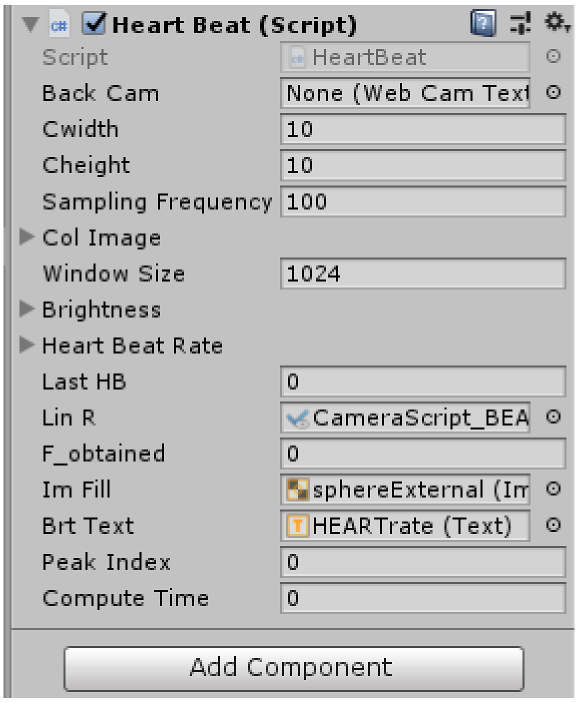
$$X_k = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} e^{\frac{-2\pi i(2n)k}{\frac{N}{2}}} + \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} e^{\frac{-2\pi i(2n+1)k}{\frac{N}{2}}} = E_k + e^{\frac{-2\pi ik}{N}} O_k$$

3.Scripting

The main script used in this asset is: “HeartBeat.cs”. The most important coroutine that implements the image treatment is:

```
/// <summary>
    ///  This corrutine obtains these steps in order:
    ///  1) reads the pixels of the image
    ///  2) obtains the brightness of the image and adds it to the list
    ///  3) calculates the FFT of the brightness
    ///  4) obtains the heart rate frequency
    /// </summary>

    IEnumerator imageTreatment()
    {
        while (true)
        {
            float time0 = Time.fixedTime;
            //only if the camera is playing
            if (backCam.isPlaying)
            {
                //get the stored pixels into the colImage
                colImage = backCam.GetPixels32();

                //loop trough the image
                double TempBrightness = 0;
                for (int ii = 1; ii < Cheight - 1; ii++)
                {
                    for (int jj = 1; jj < Cwidth - 1; jj++)
                    {
                        //matrix index to go from 1D to 2D
                        int indxMatrix = ii * Cwidth + jj;

                        //compute brightness
                        TempBrightness += colImage[indxMatrix].r;

                    }
                }
                //this is the temporal brightness computed in the image
                TempBrightness = TempBrightness /(double) (Cwidth * Cheight);

                //it is then added to the list
                brightness.Add(TempBrightness);


                if (brightness.Count > windowSize)
                {
                    //THESE ARE THE INPUT VALUES

                    //this is the window size
                    for (int ii = 0; ii < windowSize; ii++)
                    {
                        //Debug.Log(ii);Y_inputValues = new double[windowSize];
                        X_inputValues[ii] = ii;
                        Y_inputValues[ii] = (brightness[ii + brightness.Count -
1 - windowSize]);

                    }


                    //perform complex opterations and set up the arrays
                    Complex[] inputSignal_Time = new Complex[windowSize];
                    Complex[] outputSignal_Freq = new Complex[windowSize];

                    // sample array to complex that will be send to the FFT
                    inputSignal_Time = doubleToComplex(Y_inputValues);


                    //result is the output values once FFT has been applied
                    outputSignal_Freq =
FastFourierTransform.FFT(inputSignal_Time,false);


                    double[] outputV = new double[windowSize];
                    //get module of complex number
                    for (int ii = 0; ii < windowSize; ii++)
                    {
                        //Debug.Log(ii);
                        outputV[ii] =
(double)Complex.Abs(outputSignal_Freq[ii]);
                    }
                    // find peak only in the first half of the chart
                    // draw in the debug mode the FFT output
                    double MaxPeak = -1000;
                    peakIndex = 0;

                    // looping in the FFT
                    for (int i = 1; i < outputV.Length/2 - 1; i++)
                    {
                        Debug.DrawLine(new UnityEngine.Vector3((i - 1)*3,
(float)outputV[i] + 10, 0), new UnityEngine.Vector3(i*3, (float)outputV[i + 1]
+ 10, 0), Color.red);

                        // discard very low frequencies i>5
                        if (outputV[i] > MaxPeak && i>5)
                        {
                            MaxPeak = outputV[i];
                            peakIndex = i;
                        }
                    }

                    //obtain the frequency in beats/second using the FFT
                    f_obtained = (double)peakIndex / (double)windowSize *
(double)samplingFrequency * 60;

                    //display results
                    brtText.text = "" + Mathf.Round((float)f_obtained);

                    imFill.fillAmount = (float)f_obtained/220;
                }

            }
            yield return new WaitForFixedUpdate();

            float time1 = Time.fixedTime;

            //time that takes this loop to finish
            computeTime=(time1 - time0);
        }

    }
```

The variables of the script are fully documented in the “*.cs” file.



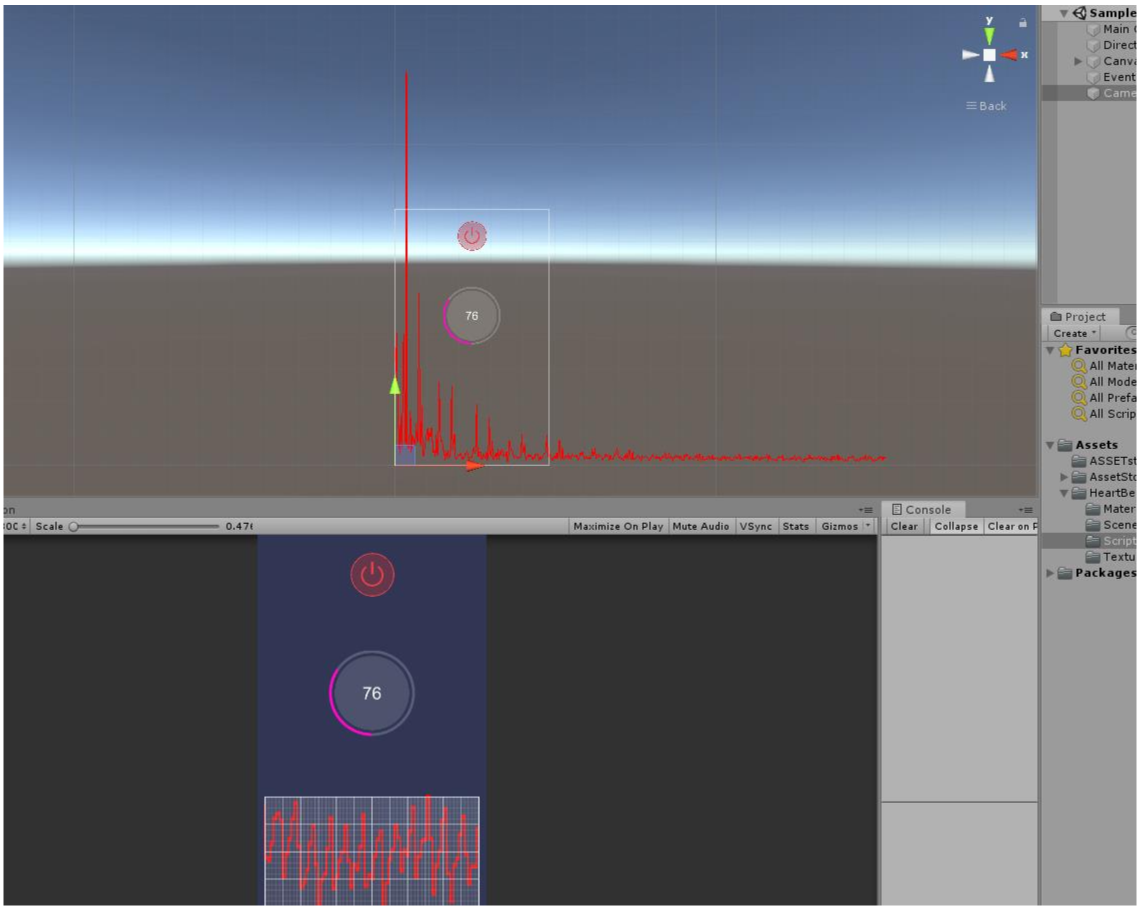
Heart Beat (Script)

Script	HeartBeat
Back Cam	None (Web Cam Text
Cwidth	10
Cheight	10
Sampling Frequency	100
Col Image	
Window Size	1024
Brightness	
Heart Beat Rate	
Last HB	0
Lin R	CameraScript_BEA
F_obtained	0
Im Fill	sphereExternal (Im
Brt Text	HEARTrate (Text)
Peak Index	0
Compute Time	0

Add Component

The peak index represent where is placed the frequency on the FFT chart.

The FFT is drawn in debug mode and will not appear on the final app. This is an example of the FFT drawn in the scene:



4. Video tutorial

We have a video tutorial explaining how the scripts and game mechanics works.

https://youtu.be/j82LdCUg-9s