UNIVERSITAT DE LLEIDA
Polytechnic School
Master's Degree in Informatics Engineering
Embedded and ubiquitous systems
Fernando Guirado Fernandez

BGC - Final Presentation

Albert Eduard Merino Pulido

Date: Friday 03rd February 2017

Course: 2016 - 2017

Index

1 Introduction

A brief introduction explaining the main objectives of the project, including the goals to achieve with the project.

The project is to develop a Biometric Game Controller (BGC). This controller will be used to allow the user interaction with the collaborative project that is being doing with the Intelligent Systems and Computer Graphics subjects.

From this project at the beginning we set some goals in order to achieve them at the final of the project. This goals are the following:

a) Knowing how the Arduino components work
b) Interpret the data that the sensors give to us
c) How use them in a useful way
d) How send the lectures between Arduino and Food Collector game.

As you can see all the objectives are related with the sensors of th Arduino as well as the communication between the Arduino board and the code in C++ that we had in the project called Food Collector. This was really the most important objective, because this will do the all project work correctly or not.

2 BGC sensing and expected behaviour

We will use the temperature to determine the colour of the map. When the temperature is less than or equal to X the map's colour changes to blue palette colours. Otherwise, when the temperature is greater than X the map's colour changes to red palette colours.

We will use the distance in centimetres to determine if the user needs to pause the game or resume the game. When the distance between the user and the Biometric Game Controller (BGC) is higher than W centimetres the game will be paused automatically. Although, if the user brings its closer to the BGC, i.e. less than W centimetres, the game will be resumed automatically.

We will use joystick's movements to move the map's camera. The user will be able to change the perspective of the map moving the joystick along either the X or the Y axis. When the user moves the joystick's X axis, the map will rotate the respective X axis. . When the user moves the joystick's Y axis, the map will rotate the respective Y axis. So you can see this kind of movements on the next figures.

Figure 1 Rotate X axis

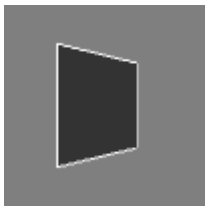


Figure 2 Rotate Y axis

We will use the joystick's button to turn on or turn off the player lights. The player will have lights, like a car, to illuminate the corridors that are in front of it. Then, the user will be able to turn on or turn off these lights whenever the user wants.

We will use BPM (Beats Per Minute) to increment or decrement the game's velocity. When the user is nervous the game will run slower trying to calm the user. When the user is calm the game will go faster. So, depending of the BPM the program decides the correct game's velocity for the user.

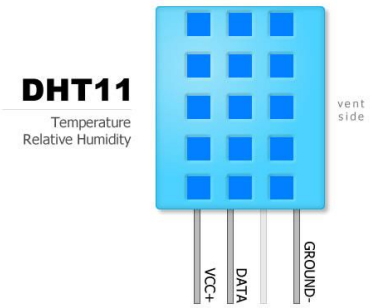
We will use accelerometer movements to move the player around the map. When the user inclines the BGC using the X axis the player will move to right or to left depending if the user inclines the BGC to the left or to the right. When the user inclines the BGC using the Y axis the player will move to up or to down depending if the user inclines the BGC to up or to down.

3 Sensors used on the project

In this section we talk about the specification of each sensor use don the BGC Project. We use 5 sensors DHT11 for the temperature, HC – SR04 for the distance, Joystick to move the camera, heart rate to increment and decrement the game velocity and finally adxl345 sensor to move the player.

3.1 Sensor DHT11

The DHT11[1] Temperature & Humidity Sensor features a temperature & humidity sensor complex with a calibrated digital signal output. By using the exclusive digital-signal-acquisition technique and temperature & humidity sensing technology, it ensures high reliability and excellent long-term stability. This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a high performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness.



3.2 Sensor HC - SR04 (Ultrasound)

Ultrasonic ranging module HC - SR04[2] provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work: (1) Using IO trigger for at least 10us high level signal, (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back. (3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning. Test distance = (high level time×velocity of sound (340M/S) / 2,

[1] http://www.micro4you.com/files/sensor/DHT11.pdf
[2] http://www.micropik.com/PDF/HCSR04.pdf

3.3 Sensor Joystick

Thumb Joystick[3] is a compatible module which is very similar to the analogue joystick on PS2 (PlayStation 2) controllers. The X and Y axes are two ~10k potentiometers which control 2D movement by generating analogue signals. The joystick also has a push button that could be used for special applications. When the module is in working mode, it will output two analogue values, representing two directions. Compared to a normal joystick, its output values are restricted to a smaller range (i.e. 200~800), only when being pressed that the X value will be set to 1023 and the MCU can detect the action of pressing

3.4 Sensor Heart rate

Heart rate[4] data can be really useful whether you're designing an exercise routine, studying your activity or anxiety levels or just want your shirt to blink with your heart beat. The problem is that heart rate can be difficult to measure. Luckily, the Pulse Sensor Amped can solve that problem.

The Pulse Sensor Amped is a plug-and-play heart-rate sensor for Arduino. It can be used by students, artists, athletes, makers, and game & mobile developers who want to easily incorporate live heart-rate data into their projects. It essentially combines a simple optical heart rate sensor with amplification and noise cancellation circuitry making it fast and easy to get reliable pulse readings. Also, it sips power with just 4mA current draw at 5V so it's great for mobile applications.

[3] http://www.west-l.com/uploads/tdpdf/101020028_eng_tds.pdf
[4] https://www.sparkfun.com/products/11574

3.5 Sensor ADXL345 (Accelerometer)

The ADXL345[5] is a small, thin, low power, 3-axis accelerometer with high resolution (13-bit) measurement at up to ±16 g. Digital output data is formatted as 16-bit twos complement and is accessible through either a SPI (3- or 4-wire) or I2 C digital interface. The ADXL345 is well suited for mobile device applications. It measures the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion or shock. Its high resolution (4 mg/LSB) enables measurement of inclination changes less than 1.0°. Several special sensing functions are provided. Activity and inactivity sensing detect the presence or lack of motion and if the acceleration on any axis exceeds a user-set level. Tap sensing detects single and double taps. Free-fall sensing detects if the device is falling. These functions can be mapped to one of two interrupt output pins. An integrated, patent pending 32-level first in, first out (FIFO) buffer can be used to store data to minimize host processor intervention. Low power modes enable intelligent motion-based power management with threshold sensing and active acceleration measurement at extremely low power dissipation.

3.6 Sensor HC-06 (Bluetooth)

This Bluetooth module can easily achieve serial wireless data transmission. Its operating frequency is among the most popular 2.4GHz ISM frequency band (i.e. Industrial, scientific and medical). It adopts Bluetooth 2.0+EDR standard. In Bluetooth 2.0, signal transmit time of different devices stands at a 0.5 seconds interval so that the workload of Bluetooth chip can be reduced substantially and more sleeping time can be saved for Bluetooth. This module is set with serial interface, which is easy to use and simplifies the overall design/development cycle.

[5] https://www.sparkfun.com/datasheets/Sensors/Accelerometer/ADXL345.pdf

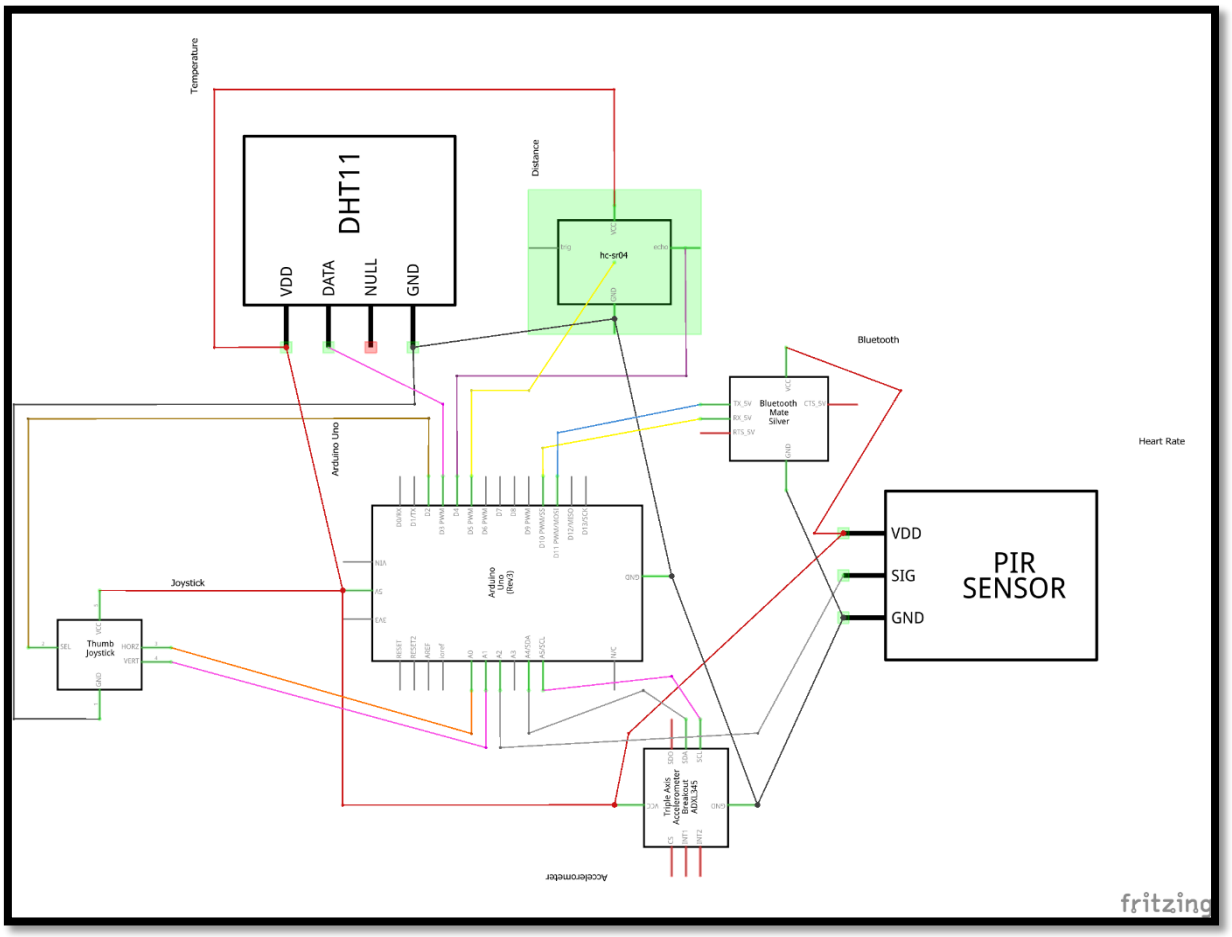
4 BGC wiring schematics

To do this activity we will need to connect all the sensors to the Arduino. We will use this table to perform all the connections. Here are all the sensors' pins to be used and the respective pins on the Arduino either digital or analogue.

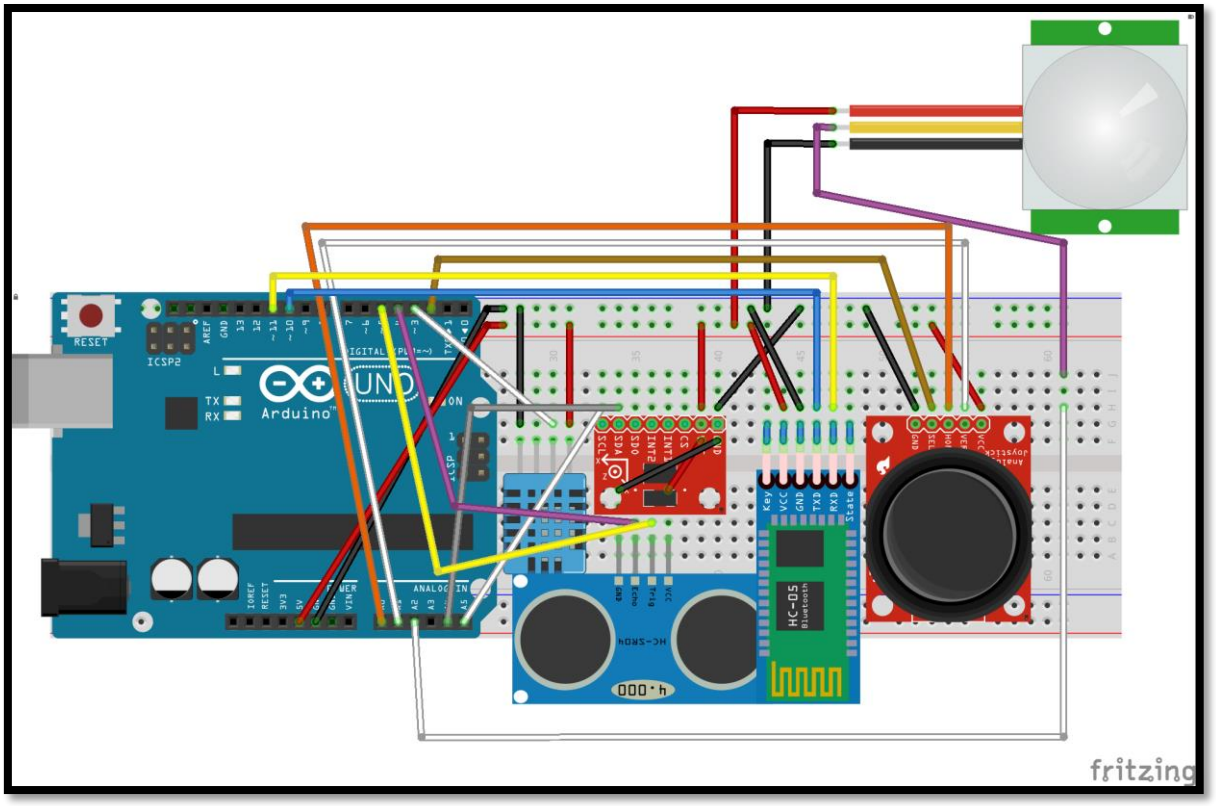
## DHT11 ## Signal -> Digital 3 GND -> GND VCC -> 5V	## Ultrasound ## Trig -> Digital 4 Echo -> Digital 5 GND -> GND VCC -> 5V	## Joystick ## HORZ -> Analog 0 VERT -> Analog 1 SEL -> Digital 2 GND -> GND VCC -> 5V
## Heart Rate ## DATA -> Analog 2 GND -> GND VCC -> 5V	## ADXL345 ## SDA -> Analog 4 SCL -> Analog 5 GND -> GND VCC -> 5V	## HC-06 ## RXD -> Digital 10 TXD -> Digital 11 GND -> GND VCC -> 5V

Table 1 Sensors connections

To draw the schematic diagram and the protoboard diagram we used the software called Fritzing[6]. Fritzing is an open-source hardware initiative that makes electronics accessible as a creative material for anyone. On the next images we can see how the connections are performed and how are the sensors put on the protoboard.

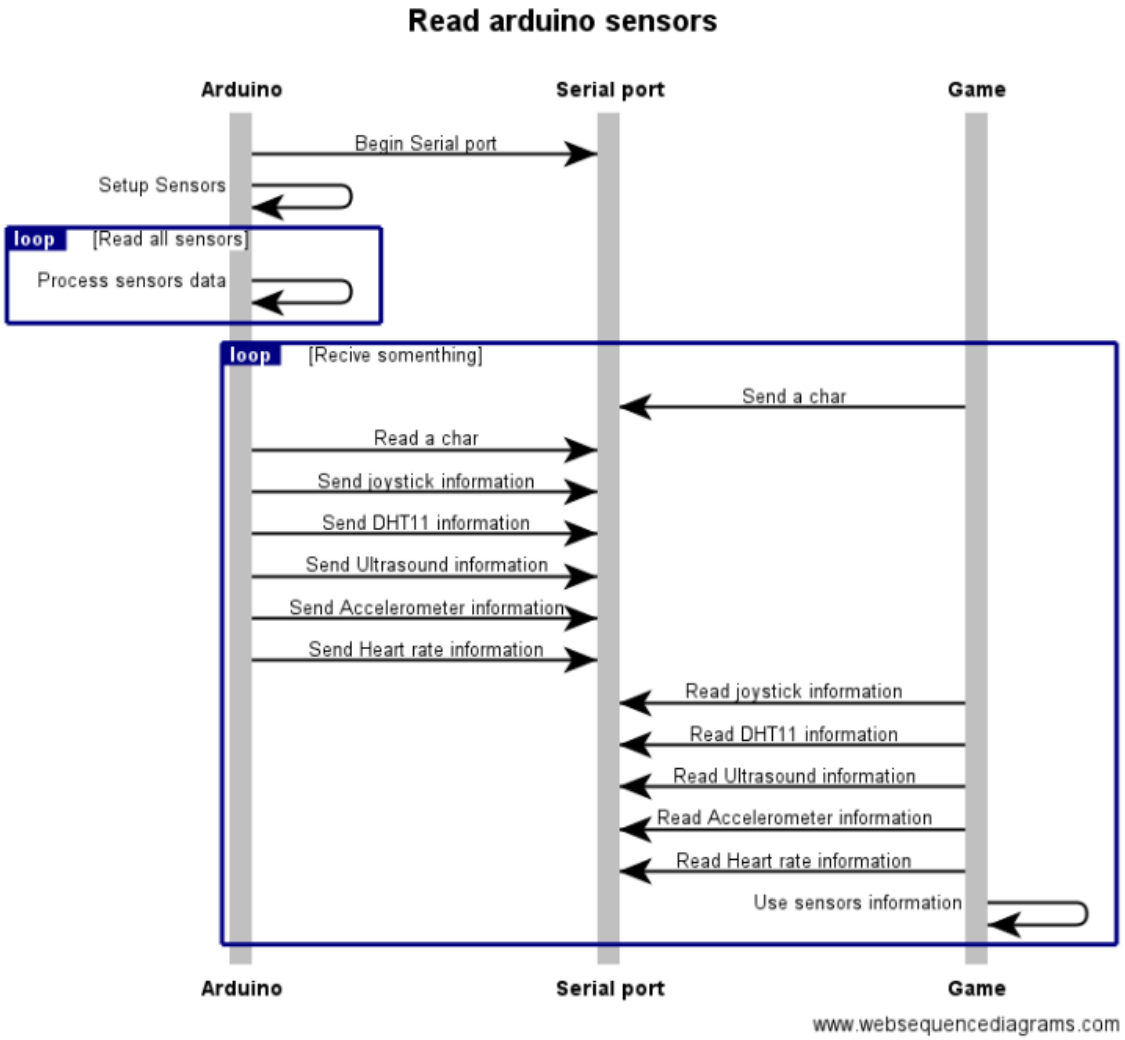


[6] http://fritzing.org/home/


RESET
ICSP2
UNO
Arduino
TX
RX
HC-05
fritzing

5 How is it implemented?

5.1 The sensors processing



5.2 The global project Food Collector interface

In this point we explain how we send the data and then what the data we transfer from Arduino to the C++ code means.

We used to different libraries in order to connect[7] the C++ code with the Arduino's serial port. The other library reads and parse[8] all the JSON information that C++ code read from serial port.

As we saw on the last DSS Arduino is continually reading and processing the sensors information, so when the game need the sensors' information send a char to the serial port and then the Arduino send a JSON string with all the information related with the sensors.

The order of the information in the JSON string is the following:

[7] https://github.com/todbot/arduino-serial

[8] https://github.com/open-source-parsers/jsoncpp

1. Joystick lecture
2. Joystick button lecture
3. Temperature lecture
4. Distance lecture
5. Accelerometer lecture
6. Heart rate lecture

This is an example of a possible lecture:

{"C":4,"S":1,"T":0,"P":1,"M":360,"V":2}

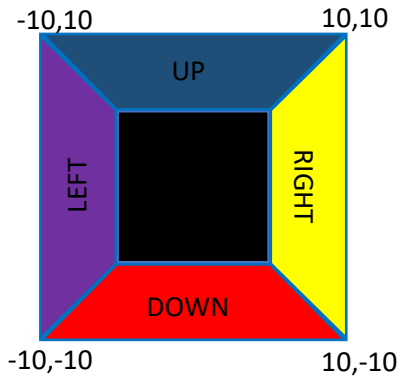
This information means for the program:

- C = 4 -> Camera UP
- S = 1 -> Shoot the bullet
- T = 0 ->Water texture
- P = 1 -> Resume the game
- M = 360 -> Don't move the player
- V = 2 -> Keep the game velocity

5.3 The BGC behaviour.

In this point we know how the data is send to the game, then we explain how the data affect the Food Collector behaviour.

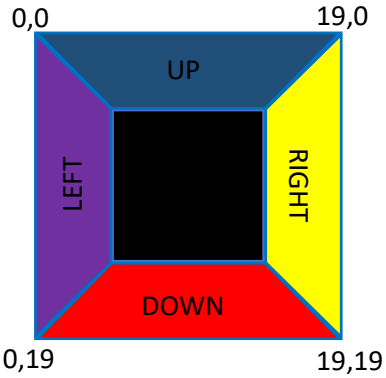
1) As we said previously, we use the DHT sensor in order to measure the temperature. With this temperature what we do is decide which texture we're going to put into the project.
   a) temperature <= 26ºC -> Water texture
   b) temperature > 26ºC -> Lava texture
2) We decided to use the accelerometer to control the movement of the player, because we mustn't use the joystick to move the player. To decide which direction we'll take from the accelerometer lecture we use the next figure:



As we can see on the last figure the directions are decided using a 5 different ranks identified with 5 different colours.

- If the accelerometer is on the black square the camera is quiet
- If the accelerometer is on the blue square the camera go to up direction
- If the accelerometer is on the yellow square the camera go to right direction
- If the accelerometer is on the red square the camera go to down direction
- If the accelerometer is on the violet square the camera go to left direction

3) We decided to use the joystick to control the movement of the camera. To decide which direction we'll take from the joystick lecture we use the next figure:



As we can see on the last figure the directions are decided using a 5 different ranks identified with 5 different colours.

- If the joystick is on the black square the player is quiet
- If the joystick is on the blue square the player go to up direction
- If the joystick is on the yellow square the player go to right direction
- If the joystick is on the red square the player go to down direction
- If the joystick is on the violet square the player go to left direction

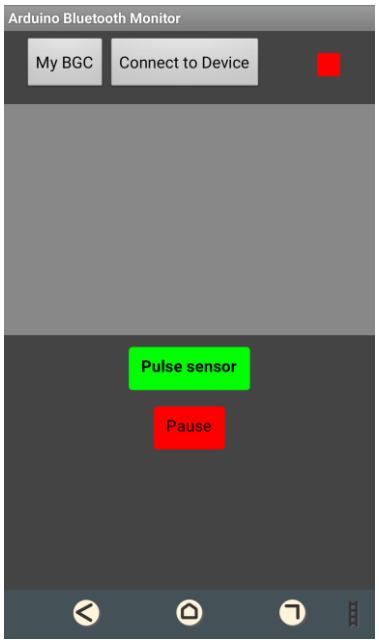
When the user click the joystick's button the tank shoot a bullet if is available.

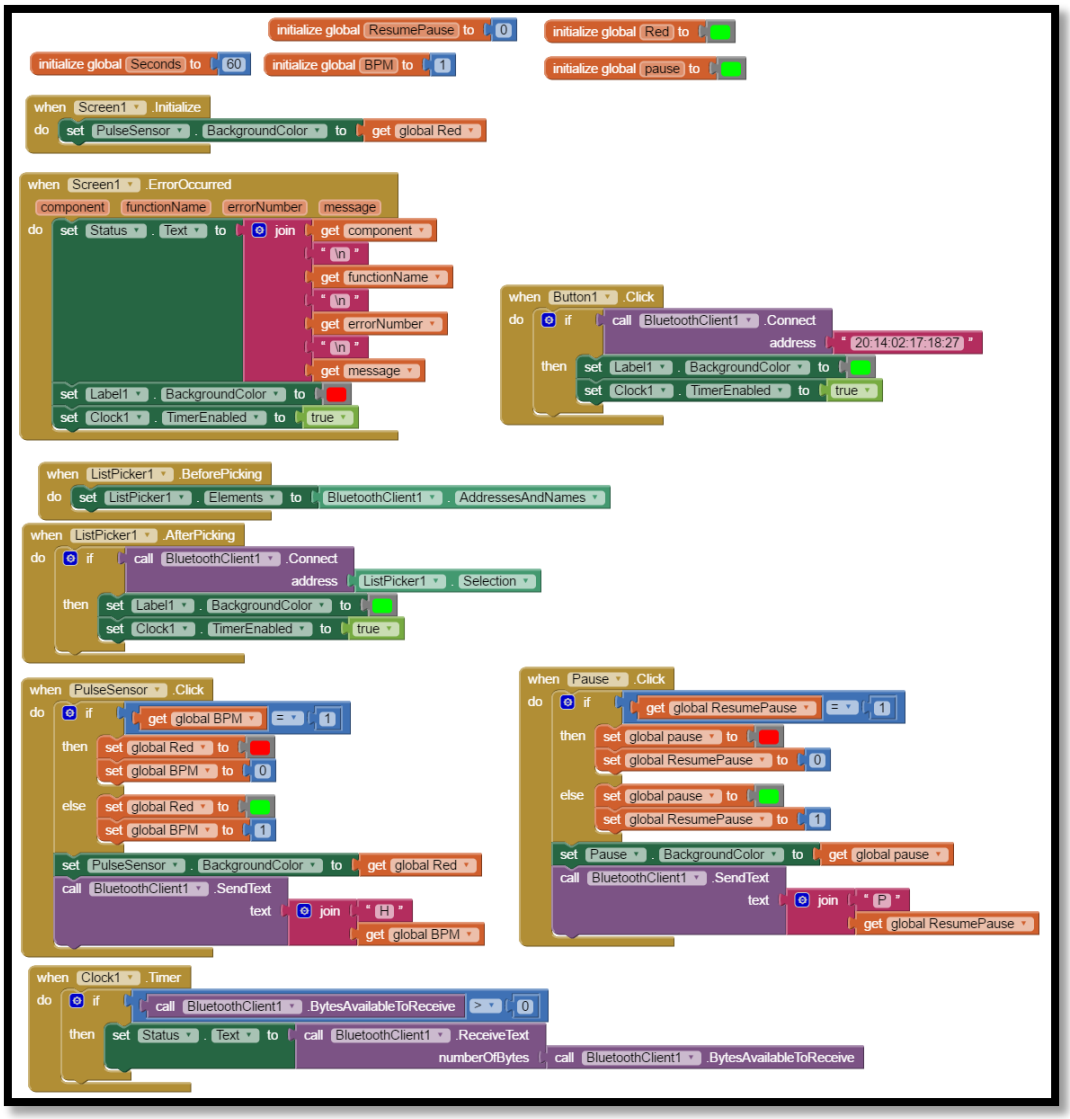
4) The heart rate of the player determines the speed of the game. If the heart rate is slow, then the speed of the game is slow and if the heart rate is fast the speed is also fast.

We determine the calm state is when the BPM is between 70 and 80 beats per minute. If the BPM is higher than 80 and less than 110 beats per minute the game reduce the game's velocity trying to change the user's state to calm. So if the BPM is less than 70 the game increase the game's velocity.

5.4 The monitoring application

We use MIT App Inventor 2 to develop an application to connect an Android device with the BGC by Bluetooth. When the application and the Android device are connected you can see a green square that it indicate the successful connection between the application and the BGC.
Once the Android device and the BGC are connected, the BGC sends trough the Bluetooth module all the information related with connected sensors. On the next images we can see the blocks used to program the application and a screenshot of the application. With the app you can deactivate the pulse sensor and pause/resume the game as well.





6 Conclusions

During the experimentation with the game controller we have seen that the accelerometer sensor is not the best option to move the player, because sometimes the response time is too high and it's difficult to change the direction quickly, although we use it to move the player.
Another problem was the heart rate sensor. We have observed that sometimes is not very accurate, the difference in the obtained BPM between two readings is not real. For this reason we are experimenting some violent changes on game's speed. We have solved the problem limiting the minimum and the maximum speed of the game.
Related with the sensors behaviour, the majority of the sensors' behaviour are the same that we decide on the previous delivery. Just one change it behaviour that it is the joystick's button. Before we decided to use it in order to switch the player and enemy lights, but we realised that we need something to shoot the player's bullet, so we implement this button in order to shoot the player's bullet.
In general we can conclude that the behaviour of the BGC is very good although there are times that the lectures aren't correct and the behaviour of the game isn't correct.