Implementation

The aim is to simulate how to improve user experience using biofeedback. For the simulation, we have developed a FPS game using Unreal Engine 4(UE4). To acquire the biofeedback, we have used sensors that can measure different states of the human body, which are changing during the time of playing the game. We have engaged two sensors (measure three body functions) for to get eye movement using EOG signal, read pulse rate of the heart using ECG signal and sweat flow-using GSR signal. At beginning, we planned to use a temperature sensor to get temperature level of the real player. When we consider about the temperature changing during the playing a game, it can be negligible. Therefore, we went with the above-mentioned sensors. In the development process, we were testing each sensor one by one to get real player biofeedback and continue the process for the remaining sensors by analyzing the sensor data and determine how to control the system and adding threshold values.

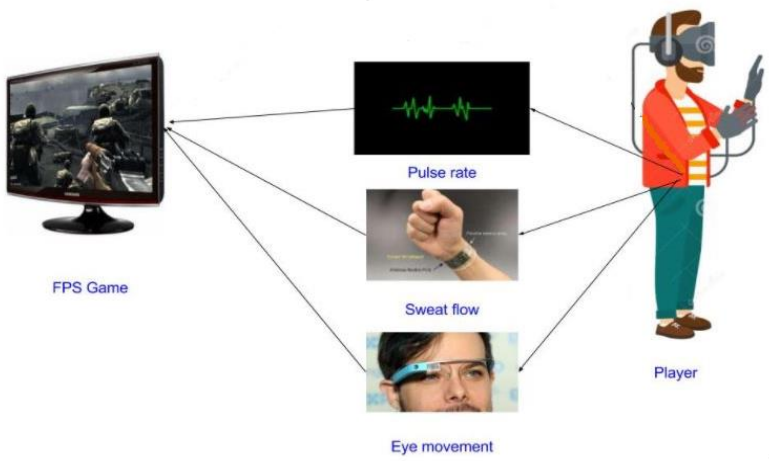

Fig. 1 Conceptual Design

As shown in the following figure 1, we have got the sensor reading from the player and given that biofeedback to the game to improve user experience of the game. The Design flow was according to the figure 2. It is the optimized idea of the complete project.

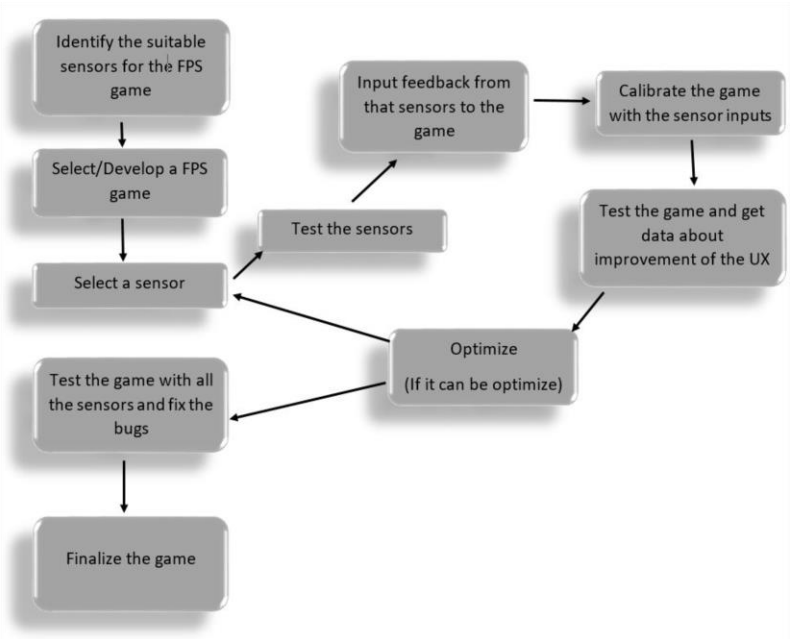

Fig. 2 Design Flow

At the beginning, Heart and Brain SpikerShield (Fig. 3) was used to get ECG and EOG signal from the real player and calculated the beat per minute (BPM). Signal that acquired by SpikerShield is clear signal. Therefore, filtering and smoothing techniques were not required. Using original signal, BPM was calculated by counting R-R interval occurrence in ECG signal pattern for a minute. Eye movement was identified by the peak occurrence in EOG signal when player looks left or right side. According to the use of electrode, peak occurrence side will change by the side of view. After calculating BPM and eye movement, those two were separately mapped to the game. Biofeedback values were taken by reading the serial port. At the end of each implementation, testing was done and threshold values were configured. When the heart rate is increasing or decreasing, the speed of the game character, aiming to a target, enemy generation, damaging to the enemy will increase or decrease for threshold value of heart rate respectively. If player looks left or right, the screen area of the game background will be rotated by 15 degrees angle.

After mapping BPM and eye movement separately, those two were needed to map at the same time. Therefore, acquiring ECG and EOG signal at the same time was the major problem with SpikerShield because it has only one

channel to get input. As the solution, we had to change the setup that fulfill the requirement of acquiring signals at the same time. The solution was the OpenBCI (Fig. 4), which has 8-channel output. When we consider about the output signal of ECG and EOG signal by the OpenBCI, there are lot of noises in those signals. Here, we have to apply noise reduction techniques such as filtering and smoothing. After applying the filtering and smoothing, we could process calculation as the previous structure.

The GSR (Fig. 5) sensor is used to get the tiredness of the player. It measures the electrical conductance of the skin and produce the result of the sweat flow of the player real time. Using the sweat flowing result and heart rate, we can check the player's physical state that the player is tired or not. Combining the GSR sensor results with the heart rate at the game engine, we can get the tiredness value. If the player is tired, the game character speed, aiming to a target, enemies on field, damaging to the enemy will decrease and waiting time will be introduced to the player to be relax. If player is in very active state, Player can play high-speed game play or if player is very tired, he will not get more tired while playing the game because game environment will be very slow. When we consider about the GSR values for different player, there are different range of reading gives. In accordance with the different values in GSR, we implemented profile for each player. At the beginning, player is given to choose the state of the body. According to the player response, configuration on system is performed. Then it acts as the same way for all the players.

Fig. 3 Heart and Brain SpikerShield

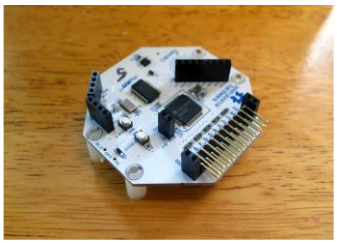

Fig. 4 Open BCI

Fig. 5 Galvanic Skin Resistant