Plus+ : Exploring Using the Body as a Game Controller

Maria A Yala
OCAD University
Toronto, ON, Canada
Digital Futures, MDes

ABSTRACT
Plus+ is a body-controlled web game that aims to raise HIV prevention awareness through gameplay. The interactive installation was inspired by the notion of using digital games as an educational tool in the fight against HIV/AIDS. Using real-time body tracking, the game tracks the position of a player's right wrist and uses this data to move the player's avatar on the screen. The player is then able to guide their digital avatar using their action in the physical world. This work is additionally an exploration of the effectiveness of using one's body as a controller in a real-time gaming environment.

Keywords
computer vision, digital games, games for health, interactive art, poseNet, real-time body tracking, serious games

1. INTRODUCTION

Generally, the "keyboard, mouse, and game controllers are common input devices for most computer games. Alternative devices are rarely used, restricting the player's possible interaction with the virtual world. Computer vision techniques permit interaction with the computer that grants far more freedom than traditional devices, without the need of game-specific hardware and hindering cables" (Brehme *et al* 1). In this case, computer vision technologies in the form of PoseNet are used to provide the input to the computer. PoseNet is a "machine learning model which allows for real-time human pose estimation in the browser" (TensorFlow). Pose estimation involves the tracking of certain body parts either in images or videos in order to determine the position of the body part.

2. MOTIVATION

Plus+ was inspired by the use of serious games as educational tools in the field of games for health. The design of the game was inspired by 2-D vintage arcade games. The main goal of serious games is more than entertainment, they are "adequately grounded in behavioral, instructional, and communication theory, and can provide an engaging and safe environment to build and rehearse health protective knowledge and skills" (Sabben *et al* 2).

3. HOW THE INSTALLATION WORKS

The project setup is such that the video game is displayed on a large screen with an external webcam attached. The player stands in front of the screen to play and their

gestures are tracked as inputs. The webcam is required to capture real-time video of the player. The game is a single player game that runs on the web using p5.js and the p5.play.js library. P5.js is a javascript client library that allows for the creation of interactive experiences in the browser. The video feed in conjunction with PoseNet allow the player to control the avatar on the screen. The game setup was chosen so as to allow the player to be both a controller and a member of the audience. Having the gameplay shown on a large screen allowed observers to be included in the game experience even if they were not the controller during the time of play.

3.1 Body Tracking

Each instance of a pose that PoseNet tracks are stored in an array and consist of a list of keypoints and confidence scores. The term *keypoint* refers to "a part of a person's pose that is estimated, such as the nose, right ear, left knee, right foot, etc. It contains both a position and a keypoint confidence score" (TensorFlow). PoseNet tracks 17 keypoints with each detected pose and Plus+ relies on the tracking of the right wrist to work.

3.2 Games for Health

The goal of the game is to collect money while avoiding getting infected with the virus, however the game does not end if a player becomes infected with HIV; their HIV status changes from unknown to positive. The game ends when the player's score, which is at 50 when the game starts, falls below 0. Points are gained when the player collects symbols that represent protection i.e condoms, getting tested, and taking medication. Points are lost when the player picks up symbols that represent risky behaviour such as using needles to take drugs, and picking up love hearts when their condom count is down.

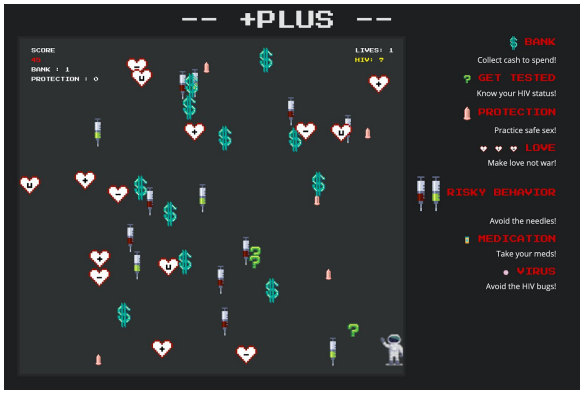

Fig. 1.0: A screenshot of the Plus+ game during gameplay

4. CHALLENGES

A number of challenges presented themselves in the creation of this game prototype as I attempted to use one's body as a game controller.

4.1 Choosing the Right Interface

Initially, the nose was the body part that was being tracked as it presented the most stable data i.e the marker for the x,y position did not jump around. However, during testing, I noticed that the interaction with the nose tracking was not natural and my neck would get stiff quickly while trying to control the avatar. The wrist, although more finicky when tracked, presented as an option that would allow the player to stand instead of sit. This action of standing instead of sitting allowed the player to engage their whole body during the game as they could move freely in front of the webcam and were no

longer tied to a chair. However, because the player had to hold out their right arm in front of the webcam as they played, the action became tiring as gameplay went on. Tracking one's hand didn't allow for room to rest like one would have when playing with a mouse, keyboard, or joystick.

```
function gotPoses(poses){
  if(poses.length > 0){
    let wX = poses[0].pose.keypoints[10].position.x; //rightWrist
    let nX = poses[0].pose.keypoints[0].position.x; //nose
    noseScore = poses[0].pose.keypoints[0].score;
    wristR = lerp(wristR, wX, 0.5);
    nose = lerp(nose, nX, 0.5);
  }
}
```

Fig. 1.1: Fetching the x coordinates of the rightWrist and nose from the keypoints, with the probability score of the nose

4.2 Choosing the Right Platform

Running the game through the web browser also presented additional challenges in terms of the amount of RAM used up. P5.js and P5.play.js running with poseNet caused the game to slow down after some time due to the load on the browser. A creative coding framework like openFrameworks, a C++ creative coding tool would have lessened the load thus making the game much faster.

4.3 Complex Interactions

Because of the load that PoseNet exerts on the browser, I wasn't able to track too many body parts. Therefore, controlling the avatar with just one's wrist quickly became boring. My intention was to make the interactions more complex in order to make the game more interesting. First, I attempted to add an interaction where the left wrist was also tracked. The idea was that the player would have 3 jumps, a jump would occur when the player brought their left wrist close to their right wrist. Jumps would allow the player to push virus sprites away from the player's avatar by waving their left hand. A jump would occur when the distance between the two wrists was 100 pixels or less. This turned out to be problematic as the real-time tracking was so sensitive that whenever a player lowered their wrist a jump would be registered, or when their arms were stretched wide a jump would be recognised, which was the opposite of what was expected to happen.

I then attempted to use the nose tracking as the additional element of body tracking instead. The idea here was to block one's nose so that when the nose was not being tracked by PoseNet, a jump would occur. However, during testing I found out that when a player covered their noses with their hands, the pose estimation score was still higher than 0.5% probability.

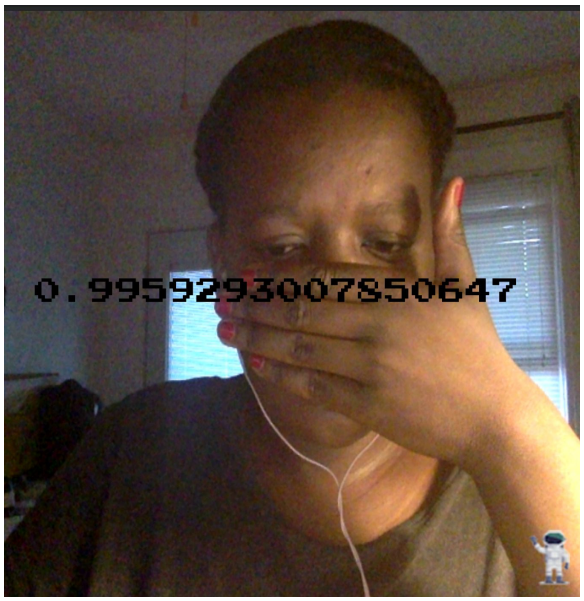

Fig. 1.2: A reading of 0.9959 when tracking the pose estimation score for the nose when covering using one's hand.

The software was still using the facial features to estimate that a nose was present. During testing, I also discovered that when the nose and lower half of the face was blocked with a white sheet of paper, the pose estimation score still fell at around 0.5%. The score would only fall lower when the players eyes were also covered, this may be because PoseNet may have been making estimation that the nose was present, because eyes were present. The software may have been assuming that a face was present as the eyes and ears were visible, thus when these facial features that are tracked are covered, the probability of nose tracking falls lower.

Fig. 1.3: A reading of 0.2641 when tracking the pose estimation score for the nose when obscuring one's eyes and nose.

Although this was successful, the interaction was not beneficial for the game as it required the player to obscure their vision. Additionally, the rate at which PoseNet tracks poses made it difficult to pinpoint a particular pose, and lowering the framerate of the p5.js sketch would prevent the real-time tracking of the application thus the jump feature was not implemented.

5. CONCLUSIONS

Plus+ as a prototype still has some kinks to iron out but it serves a purpose as a proof of concept and its creation led to the clarification of gameplay ideas. In conclusion, the development of this prototype has shown me that real-time body tracking doesn't work well in instances where one-to-one visual mapping of the interactions in the physical world are being tracked. They simply bog down the application, especially when running in the browser. Which is why in future iterations, the game will be moved to a creative coding framework off the web and will explore tracking different body parts for more complex and meaningful interactions.

6. REFERENCES

D. Brehme *et al.*, "A Virtual Dance Floor Game using Computer Vision," *The 3rd European Conference on Visual Media Production (CVMP 2006) - Part of the 2nd Multimedia Conference 2006*, London, 2006, pp. 71-78.
doi: 10.1049/cp:20061974
URL:
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4156013&isnumber=4155993

Sabben G, Mudhune V, Ondeng'e K, Odero I, Ndivo R, Akelo V, Winskell K
"A Smartphone Game to Prevent HIV Among Young Africans (Tumaini): Assessing Intervention and Study Acceptability Among Adolescents and Their Parents in a Randomized Controlled Trial"
JMIR Mhealth Uhealth 2019;7(5):e13049
URL:https://mhealth.jmir.org/2019/5/e13049
DOI: 10.2196/13049

TensorFlow. "Real-Time Human Pose Estimation in the Browser with TensorFlow.js." *Medium*, TensorFlow, 27 Sept. 2018, medium.com/tensorflow/real-time-human-pose-estimation-in-the-browser-with-tensorflow-js-7dd0bc881cd5.