Tracking Eye Gaze in Unity3D for Training

Alberto Melchor-Maldonado

University of California Merced, School of Engineering, Merced, CA, 95343

Project Advisor: Salam Daher, Ph.D. New Jersey Insitute of Technology, NJ 07012

Keywords: Eye-tracking, eye gaze, training, regions, unity3d

Introduction:

In healthcare, it is essential for medical providers to respond to certain cues in evaluating patients. As a result, all medical providers must undergo training in improving their skills in evaluation. It is difficult to maintain consistent evaluation of where trainees are looking at, therefore making it difficult to assess whether the trainees are able to detect cues in these areas. To improve these simulations, I created a package using eye-tracking capabilities. . Eye-tracking is defined as the measurement of eye movement relative to a visual array (Smith,2017). The applications of eye-tracking can be demonstrated in the developments of virtual reality and gaming platforms but there have been applications as well in the field of science such as tracking the smooth eye-tracking patterns of schizophrenia patients (Holzman,1973). Instead of general eye-tracking devices, trainees may opt instead for another form of eye-tracking such as a head-mounted tracker. Head-mounted devices have been stated to allow for head movement, and use multiple data points to record eye movement such as "allowing the pupil glint to be measured from multiple angles"(Cooke, 2005). Although the data may become more precise and efficient to provide analyses, head-mounted eye-trackers may be more costly than traditional eye-tracking devices, instead, the availability of gaming and web-based eye trackers is an alternative.

Intent:

The project's intent is to create regions of interest in space, detect eye gaze, record time stamps, coordinates, and region name should the region be gazed at. The regions are instantiated at the start of the interaction scene and the recording of the eye-gaze is dependent on the casts the eye-gaze sets on them.

Materials and Methods:

The materials used in this project are the Unity Development platform(Unity) version 4.5(Unity 4.5), the Tobii Eye Tracker version 5, and the Tobii Unity SDK are required to integrate the eye gazing environment. Also, a laptop with the specification of Intel(R) I5-7200 processor, an Intel(R) HD Graphics 620 system graphics component, and running on the Windows 10 operating system are used to connect and play the scene. Installation of the Tobii Unity SDK package is necessary for scripting components that allow for interaction with the Unity scenes(SDK). I created a C# script that allows for a given input CSV file with components: x,y,z, width, height, and depth. These parameters are read to generate regions across space in the Unity environment which are draggable as well. The script works by integrating Unity Raycasts (Raycast) that transforms the eye gaze's ray from the eye to the screen in order to detect the hits(eye-gaze) the user makes with the regions. The detection results in the following elements: name of the region, time stamp, and coordinates, being recorded as a string statement. As a result of these casts, the statements are appended to a list and the list is written to a text file.

Results:

As a result, trainees can now repeat eye-gaze training again but now it tells trainees in real-time where and what they are looking for, this allows for new forms of feedback that address other factors other than the eye gaze being read. I was able to create a unity package that can take the desired regions as input from a CSV file. Simulation Developers can use the unity package and modify the regions of interest to fit the needs of their simulation project. The eye-tracker records the eye movement and hits it casts on regions. The hits resulted in its information being stored and outputted. The value of this development is that provides a tool for trainers to detect where the users are looking and determine if they looked at the correct region. Future work may

include testing for accuracy with users and collecting data, and running simulations using head-mounted eye trackers. Machine learning techniques may be used to train the model in determining which gazes are correct or not.

Conclusion:

This work is important as it may improve medical training for both simulations and real-world simulations with patients. I developed a unity package that takes in input files and instantiates regions in space. Users are able to change and transform the regions of interest. The user's eye-gaze is detected using Tobii Eyetracking. I used raycasting in unity to detect regions where the user looks. The raycasting produces a timestamp for when it was hit, the coordinates in space where the casts were detected, and the name of the region that was cast. This information could help in determining where trainees were looking at, and allow for medical trainers to evaluate the resulting data and give feedback for a trainee. This technology can be used in healthcare simulation as well as other fields such as veterinary science and oceanography.

References

COOKE, L. (2005). Eye Tracking: How It Works and How It Relates to Usability. *Technical Communication, 52*(4), 456-463. Retrieved September 6, 2020, from http://www.jstor.org/stable/43093613

Holzman, P., Proctor, L., & Hughes, D. (1973). Eye-Tracking Patterns in Schizophrenia. *Science, 181*(4095), 179-181. Retrieved September 9, 2020, from http://www.jstor.org/stable/1736872

Raycast https://docs.unity3d.com/ScriptReference/Physics.Raycast.html

Smith, T. J., & De Urabain, I. R. S. (2017). Eye tracking. In B. Hopkins, & R. G. Barr (Eds.), *The Cambridge encyclopedia of child development* (2nd ed.). Cambridge University Press. Credo Reference: https://search.credoreference.com/content/entry/cupchilddev/eye_tracking/0

Tobii SDK https://developer.tobii.com/pc-gaming/unity-sdk/

Unity https://unity.com/

Unity 4.5 https://unity3d.com/unity/whats-new/unity-4.5