Amber Elferink

amberelferink.com | AmberElferink | Utrecht (The Netherlands)

Amber-Elferink | +316 8200 4212 | amber.elferink@gmail.com

Robotics and Game Technology master student with interest in Virtual Reality.

Natural locomotion and haptic feedback in virtual reality will provide lasting experiences. My goal is to make VR as immersive as possible, for as many people as possible. For more visuals go to:
www.amberelferink.com

Education

Double Master: Robotics & Game and Media Technology

View courses and material covered *here*

Master Robotics TU Delft Delft University of Technology

Sept 2020 – December 2022 (expected)

Current grade (8.2 / 10)

Utrecht University Game and Media technology Master

Sept 2019 – December 2022 (expected)

Current GPA 4.0/4.0 (8.46 / 10)

Computer Science Minor

Sept 2017 – Dec 2018

GPA 4.0/4.0 (8.04 / 10)

Chemistry Bachelor

Sept 2014 – June 2017

Skills

Languages

- ✓ C#, C++ (OpenCV, OpenVR, Arduino, OpenGL)
- ✓ Unity
- ✓ Python (OpenCV, Tensorflow)
- ✓ Basics: Javascript/NodeJS, SQLite, HTML (+ pug)/CSS
- ✓ Dutch (native), English (C1-C2)

Others

- ✓ Adobe Photoshop
- ✓ Adobe Illustrator (Basic skills)
- ✓ Arduino

Work experience

Master Thesis Internship: Freeaim

October 2021 – Expected: December 2022

With Freeaim VR shoes, you can walk in virtual reality, while the motorized shoes keep you centred within your living room. This enables you to walk naturally in the endless virtual worlds. For a video, you can check out my homepage, or the website of the product: freeaim.com.

My thesis goal was to research the best method to map the physical movement of walking on the VR shoes to movement in VR. This involved setting writing most of the code to interface with the VR shoes themselves, including the PC side software (C++ OpenVR) and embedded communication to improve the responsiveness of the VR shoes. It also included cooperation towards better tracking and positional correction. The final experiment was performed in Unity, with self-built locomotion methods that can make the player move forward in VR with different manners using the shoes with VR trackers, and testing user experience to find the best one.

VR input Developer
SenseGlove – View videos about the project
Juli 2021 – September 2021
For my master internship, I developed an input solution for SenseGlove. SenseGlove creates haptic gloves to grab, hold and feel objects in virtual reality (VR). However, most applications still use controllers as input. Some interactions such as teleport are not convenient with VR gloves. Adding this hardware would allow for faster interactions and more compatibility. My role in this internship was to add hardware to the gloves, such as a joystick and buttons.
Creating the electronics, firmware, software and CAD were all part of this internship. This was done respectively in C, C#, Unity and Autocad Fusion 360. The result was a comfortable joystick and button implementation and a well-received demo. The joystick and button functionality was fully implemented in the current SenseGlove software. I cannot share footage of the design due to confidentiality, but the result was a great user experience and a well-received demo. This is the first step to make the gloves compatible out of the box with many games and simulations available on the market.
Watch a video about it **here.**

Software Developer
Fair2Media - View videos about the project
May 2019 – August 2019
Filling the time between bachelor and master, I worked on the Ditou table at Fair2Media. Fair2Media is a small company with a total of 5 people. The Ditou table is an interactive table where a beamer projects a map of real places via Unity. On this map, you can use round disks as a cursor to place objects such as windmills and solar panels. The round disks are tracked by a camera above the table. My responsibility was to develop a new tracking algorithm for round disks on the Ditou table (C++ and OpenCV), and to make an environment in Unity loading, customizing and saving the map. Thereby, by using the tracked positions, 3D windmills and solar panels could be built on the map. Therefore, this is the entire software that is controlling the table. The tracking and Unity environment communicated via NodeJS. The table is currently used to demonstrate the outcomes for building windmills and solar panels for citizens' initiatives.
Watch a video about it **here.**

View work before 2019, videos and more info on my studies at:
www.amberelferink.com/workexperience

Projects

Computer Vision 3D – View project video
Two projects:
- Move a virtual cube to match the orientation of a physical piece of paper via the camera.
    - Detection was done with OpenCV by detecting the checkerboard on the paper.
    - With the orientation of this checkerboard a virtual cube is rendered on via OpenGL.
- Reconstruction of people in 3D from 4 camera views
    - By calibrating each camera with a checkerboard via OpenCV, you obtain the position and rotation of each camera.
    - To get the people in the image, background subtraction and thresholding is performed to get a mask.
    - By combining all four images, then certain 3D points are turned on that show for all cameras.
- Both projects graded with a 10/10
- **View project video**

Animation Engine – View project video

- Animation engine built from scratch
- Load rigged mesh files in multiple formats
- Drag an animation file to it, see it move
- Built with C++ and OpenGL
- View project video

Self-Driving Car – View project video

- Detecting objects via LiDar, detecting pedestrians via camera, and steering around them/stopping autonomously.
- Built in ROS, C++, with libraries Point Cloud Library and OpenCV
- Other course: evaluate different options for pedestrian detection (Python)
- *Comparison:*
    - Three feature types: Raw pixel values, HOG, and CNN (MobileNet)
    - Feeding features to two classifiers: a Linear Support Vector Machine (SVM) and k-Nearest Neightbors (k-NN).
- *Conclusion:*
    - Raw pixel values performed the worst, and the best with k-NN classifier
    - CNN features performed the best, and did not differ between the classifiers.
- View project video

Hand detection - View project wiki on Github

- Built from a template allowing to load an image and save an image.
- Featuring handtracking and detection without an external library. The background is a beamer table, which occludes the hand. The hand can be rotated in any direction.
- Graded with a 9/10**,** built in C#.

Physics simulations – View project video

- Rigid Body simulation and Soft body simulation
- Graded with 10/10
- Wrote in C++, and another simulation in Unity with only self-implemented physics.
- View project video

View many more projects and videos on my Portfolio:
www.amberelferink.com/projects