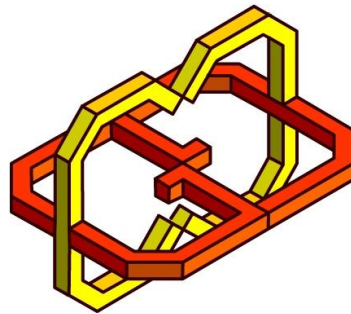


# SPIDERMAN VR

Adam Elgressy and Dmitry Vlasenko



Supervisors: Boaz Sternfeld and Yaron Honen  
Submission Date: 09/01/2019

## Contents

Who We Are:.....	2
Abstract:.....	2
Previous Work:.....	3
Tangent Systems & Development Environment: .....	4
Application Overview:.....	6
Development Process:.....	8
Future work suggestions:.....	8

## Who We Are:

We, Adam Elgressy and Dmitry Vlasenko, are undergraduate students, at the Computer Science faculty, at the Technion – Israel's Institute Of Technology.

We have participated in a Virtual Reality project, conducted by Mr. Boaz Sternfeld, and Mr. Yaron Honen, in course number: 234329.

## Abstract:

Our main goal was to simulate a first person Spiderman experience, using the HTC Vive Pro kit. We also put a goal for our self, to Learn about the new, and evolving world of VR, and gain hands-on experience, with the newest technology, alongside getting to know the Unity environment.

We have successfully made a VR application, which simulates a first person experience, of one of the most famous superheroes of all time, Spiderman. We have made it possible to walk, climb, and swing from a web, being shot from the in-game hand, all while being able to stay stationary, at a one place, but fill like you can explore the whole world.

We have simulated a physics system, which made the experience feel authentic, from the free fall acceleration, to friction with different surfaces, and many more.

We would suggest for future work to be done to add different scenarios, a leaderboard, tutorial room.

## Previous Work:

We have begun our work, on a different project, which was made to do a real-time image recognition with the HTC VIVE PRO's built-in cameras, using a pre-trained neural network, powered by TensorFlow.

In the 3 weeks of our works we accomplished:

1. Finding the right tools, for a neural network training, in the Unity environment.  
Challenges: The most famous neural network we managed to find, for image recognition, using TensorFlow, mainly uses a Python's API, and some other programming languages, but not in C#, which would have made the integration part very challenging.  
Solution: We found the following complementary tools:
  - i. TensorFlowSharp: Which allows us to use TensorFlow, with UNITY.
  - ii. ML.Net: A machine learning framework, in .net.
2. Finding the HTC VIVE PRO's SDK, for using the built-in, front cameras.  
Challenges: Probably the main challenge we have face, during the work on this project, was that the SDK, written to utilize the HTC VIVE PRO's cameras, was still in the Beta stage, and was made more to show the abilities, than to give the tools to use them. Which made it very difficult to use the hardware to our advantage.  
Solution: On the base of the available SDK, we have still managed to build several "show rooms", to demonstrate the camera's possible abilities, which we would want to exploit for our project.
3. Activating the front cameras in UNITY, and sample each frame's data, to use in the future, in the image recognition part.  
Challenges: The bottleneck we have encountered, was the low-quality of the camera's captured frames. The quality was still good enough for the image recognition, but the user experience of watching, in real time, what the cameras are filming, was horrendous.

We didn't manage to overcome this challenge, due to lack of support from the manufacturer.

The above mentioned "show rooms", were shown to our supervisor, Boaz Sternfeld. Which after several failed attempts to improve the image quality, it was decided to abandon the project.

## Tangent Systems & Development Environment:

- **HTC Vive Pro:**

The HTC Vive is a virtual reality headset developed by HTC and Valve Corporation. The headset uses "room scale" tracking technology, allowing the user to move in 3D space and use motion-tracked handheld controllers to interact with the environment.



- **Unity:**

A cross-platform game engine that can be used to create both three-dimensional and two-dimensional games, as well as simulations for desktops and laptops, home consoles, smart televisions, and mobile devices. Unity is scripted with C# in Visual Studio.



- **Visual Studio:**

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.



- **Git:**

Git is a distributed version control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity, and support for distributed, non-linear workflows.



- Blender:

Blender is a free and open-source 3D computer graphics software toolset used for creating animated films, visual effects, art, 3D printed models, interactive 3D applications and video games. Blender's features include 3D modeling, UV unwrapping, texturing, raster graphics editing, rigging and skinning, fluid and smoke simulation, particle simulation, soft body simulation, sculpting, animating, match moving, rendering, motion graphics, video editing and compositing.



- CSCAPE:

Cscape is an Unity asset, which is an Ultra Optimized powerful Cityscape generator that is able to create thousands of unique buildings.

Main strength of this plugin is that it uses serious drawcall optimizations for extreme performance and sets new frontiers for realistic and performant rendering of massive worlds.

This magic is done by using few one-pass shaders/materials for a whole city landscape. This results in an extremely optimized performance while using Static/Dynamic Batching or Occlusion Culling. (approximately 60-160 drawcalls for a city that covers a surface of 10 km<sup>2</sup>)

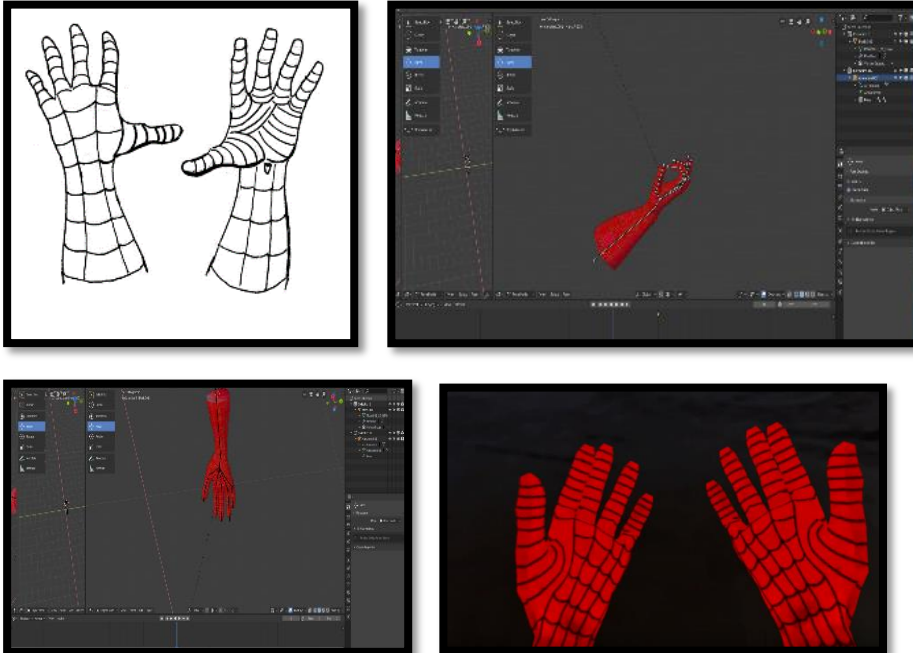


## Application Overview:

- Hand Animation:

We created a Hand model, which conforms with the classic Spiderman costume's design, using Blender.

A hand animation of the "web shooting" was created, for the full Spiderman feel. It is activated during web swinging.



- Walking:

You can walk in-game, while staying at the same place in real life.

Walking speed is adjusted to the movement speed of the controllers, when simulating walking.

To walk you need to hold the touchpad, and move your hands back, and forth, in a way, which mimics real life walking motion.

- Climbing:

You can climb on EVERY object with mesh, which includes buildings, bus stops, bridges, etc.

To climb, you need to put your hand on an object, in a way, of having your fingers touch it, then you press the application button, to initiate a holding, you will feel a haptic feedback upon success, to make the whole movement more intuitive.

Moving the same arm, in any direction, gives you the experience of climbing. Use the other hand to continue up the structures.

- Swinging:

Swing from any object with mesh.

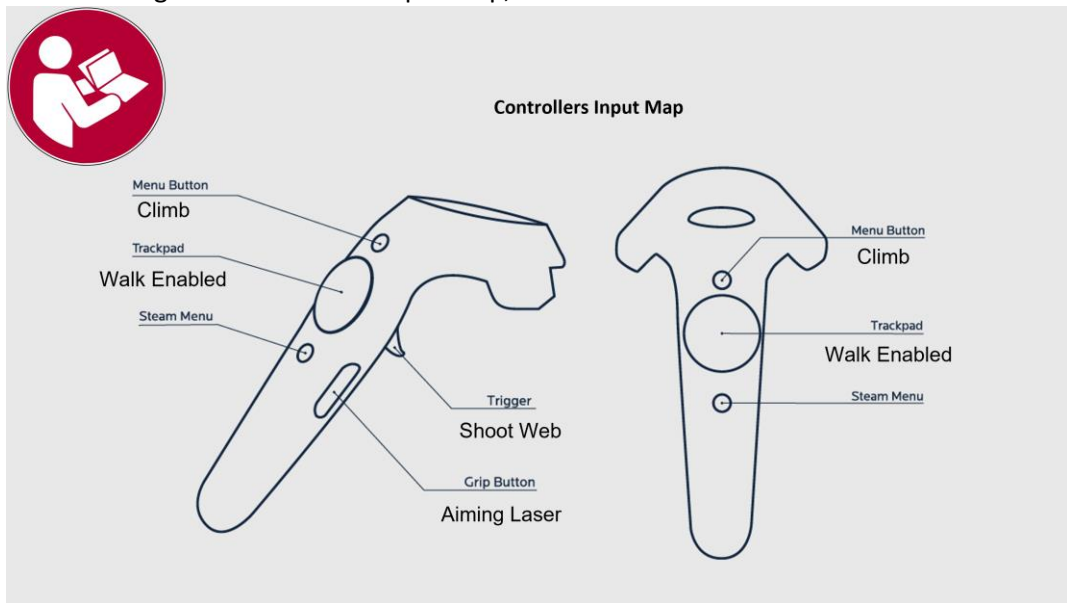
You can use either one string, or two, depending on desired direction, and velocity.

The aiming laser helps you to recognize valid actions, with green laser indicating a valid one, and the red, an invalid one.

You aim by directing your controller at different directions, and press the grip button, to show you the aiming laser. To shoot a web, and start swinging, press the trigger button.

- Controls:

The following is our controllers input map, to use the above mentioned features:





## Development Process:

The development process of our project, was characterized by modular, feature driven work, which has allowed us to adapt to the changing requirements, and risen problems, fairly quickly, and efficiently. Each of our features, such as climbing, and walking, was developed separately, so, for example, when we needed to add haptic feedback in our climbing feature, after the mid-semester meeting, The changes we have made were minimal, and we were able to update, in a way which didn't effect features, other than climbing.

We have been using the Git version control system, to allow easy tracking of our changes, and working together from different computers. Due to the large size of our project, we use Git LFS.

For a realistic physics system, to make the user feel like he is not in a simulation, from a physics stand-point, we manually tried to perfect each parameter, by implementing a trial and error approach, using multiple participants, in various stages of our development. Additionally, at the beginning, our project wasn't supposed to include walking feature, but due to feedback from those participants, we have decided to implement it, for a better user experience.

For even more of a realistic feel, we have added a dynamic sound feature, which includes traffic sound on street level, but when you go up, that sound switches to a wind breeze, which gives the user a better location perception, to enhance the user's experience.

Due to the way of our development, integrating any new feature, was practically a plug-and-play of the new feature, to the main system. This has allowed us to build this robust system, and to make it very easy to further develop, and add new features.

## Future work suggestions:

We would like to suggest a few additions in future works, to further enhance the application's user experience, and perhaps to give it various purposes:

- Leaderboard – keep track of the fastest players to go through all checkpoints, to add some competitiveness element to our system.
- More scenarios – add different scenes, for example, a climbing wall, a nature scene, with different trees, hills, and valleys.
- Tutorial room – make an intuitive tutorial room, to guide new users through all available features of the system.
- Auto aim mechanics – for the web shot, to make it go to the closest object, instead of not shooting at all, since it might be too hard to be accurate for long distance.
- VR for kids – a scenario, which will help encourage children to develop physical abilities through fun experience, such as fine motor skills, hand-eye coordination, etc.