

**Assignment 1 – Jedi Trainer**  
**CAP6121 – Spring 2019**  
**Due: 2/6/19 11:59pm**



“Help me become a Jedi, I can’t remember anything”

In order to help train young Jedi, we require a set of training tools to help them with their lightsaber skills. In this first assignment, you will build a Jedi Training simulator. The goals for this first assignment will be to get familiar with Unity 3D, the HTC Vive, and gestural interfaces.

**Minimum Requirements**

The main goal of the Jedi Trainer is to help young Jedi with their lightsaber skills. To do this, your application must have two modes. First, it must contain a training droid that moves around randomly (or pseudo randomly) and shoots lasers at you at varying times. The user’s job is to deflect these lasers with the lightsaber so he/she does not get hit. Second, the simulator will have attack droids that come after you. The user’s job in this case is to fend off these droids by using the lightsaber or the force. Note that in this game you will not have to travel in the environment, all attack droids will come to you. When using the force, you should at a minimum be able to

1. Shoot lightning out your hands
2. Grab objects to throw them at enemies
3. Send a wave of energy that can knock down an enemy
4. Heal yourself

5. See into the future (this one should be fun)

### **The Virtual World**

The world you create for the Jedi Trainer is completely up to you and your imagination. The models you use for the training droid, the attack droids and the lightsaber are completely up to you. The only requirement that I have is that you try to incorporate sounds in the game and try to add special effects for the lightsaber. You can also make the scoring for the game anyway you wish.

### **Strategy**

For force powers, I recommend using heuristics to determine how different controller movements can be detected as certain force powers. You could also use simple machine learning techniques like template matching as well (e.g., collect some training samples, then calculate the Euclidean distance between a current pose and the average of the training poses, the smallest distance wins). One approach you might want to consider is the simple gesture recognizer found in:

Taranta, E. and LaViola, J. "Penny Pincher: A Blazing Fast, Highly Accurate  $\$$ -Family Recognizer", *Proceedings of Graphics Interface 2015*, 195-202, June 2015.

You will have to adapt this recognizer to support 3D gestures.

### **Deliverables**

You must submit your source to the isuelab drive and send me an email with a README file describing what works and what does not in your application, any known bugs, and any problems you encountered. Please note that both team members must submit an individual README describing what parts of the assignment you worked on and what parts your partner worked on.

### **Grading (based loosely based on the following):**

60% correct functionality  
30% creativity  
10% documentation