M I M JOURNAL 2023 illuminate your work
Made in Millersville Journal | www.mimjournal.com

Modeling a Microsurgical Suture in Unity

Justin Stevens

Abstract

Virtual reality simulators are an effective way to train and evaluate future surgeons. These simulators require realistic modeling of physical objects such as bodily tissues, sutures, and instruments that are essential for their success. This paper discusses the challenges faced by us while modeling sutures, which are one of the most challenging objects to accurately simulate in virtual reality surgical simulators. The paper describes various approaches that have been attempted within the Unity game engine and discusses the various challenges in creating a realistic model and the parameter values that have been found to work best.

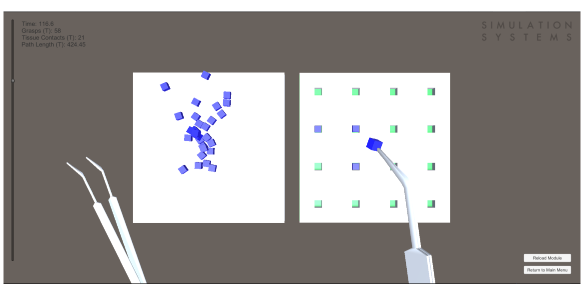

Figure 1. The current training scene for the simulator before users begin using the suture. Metrics show in the top right.

Introduction

Virtual reality simulators provide a cost-effective and safe way for surgeons to practice and perfect their craft, and to have their skills evaluated. We are attempting to build such a simulator, initially for eye surgery, in the Unity 3D engine. This engine is used to develop modern games and simulators by providing developers with various tools needed to make games and simulators. An in-game screenshot of the simulator within Unity is shown in figure 1. So far, we have succeeded in creating custom peripherals similar to real surgical instruments and sensing their location and orientation in space. An off-the-shelf 3D video system allows the user to see virtual representations of these instruments in a virtual world, in which they maintain the same position and orientation as the peripherals do in the physical world. The goal of our simulator is that users will be able to use these virtual forceps to grasp a simulated suture to stitch together simulated body tissues and tie knots, with the knot-tying as a particular point of emphasis.

The most challenging object to model has been the suture. A real suture used in microsurgery behaves similar to a fishing line, though it is much smaller. It needs to be flexible enough to easily bend when force is applied to it, yet rigid enough that it attempts to return to an entirely relaxed position in the absence of any forces. To model this behavior, we constructed a suture that is composed of a series of nodes connected by joints. Each of the nodes are a 3D object shaped like capsules that are provided by Unity. The joints are also provided by Unity. Unity provides ways to configure these nodes and joints with parameters such as mass on the nodes and maximum amount of bend on the joints. We have found ideal parameters on each of the components that produce a marginally acceptable model, but we are continuously trying to improve it.

Modeling Requirements

The simulation requires three types of objects: a tissue substrate, forceps, and a suture. In the simulator, the forceps are made up of many parts, each with Meshes, Colliders, and Rigid Bodies that are intended to appear and behave like real surgical instruments. The Meshes describe the shape of an object, while the Colliders determine how or if the object will collide with other objects, and the Rigid Bodies control everything related to the physics of an object. All of which are provided by Unity but require the settings to be tweaked. The tissue substrate allows the end of the suture to pass through it and pulls the rest of the suture along with it. When forceps tug on an end of a suture, the force should be applied evenly across the length of the suture, both deforming and moving it. When a suture has been tied into a knot, the force of friction must be sufficient to prevent the knot from unraveling.

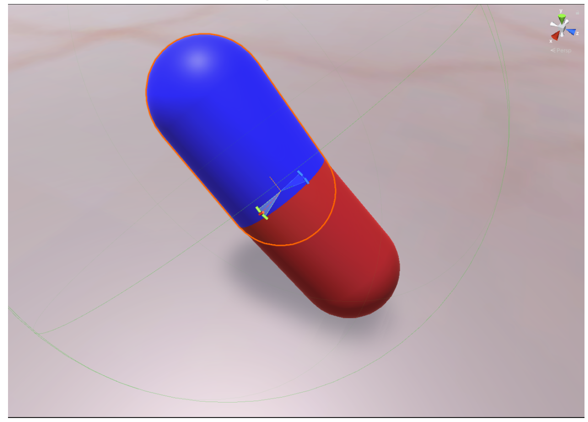

Figure 2: Two nodes, one blue and one red, connected by a configurable joint.

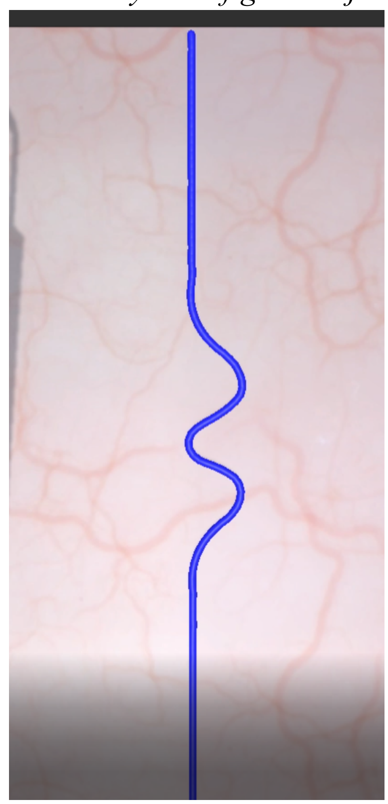

Figure 3: Screenshot of our current suture model.

Current Suture Model

The suture has been modeled as a long chain of discrete Game Objects which we call nodes, each with Capsule-shaped Colliders, Rigid Bodies, and Physics Materials, with Configurable Joints connecting each node to the next. Configurable Joints are provided by Unity and are a hidden component that restricts the movement of two objects and in our case, it restricts the movements of two nodes. When force is applied to any node, that force is distributed through the joints to the other nodes. To visualize this structure you can picture a line of nodes like the one in figure 3, but between each consecutive node, there exists a Configurable Joint that holds the two nodes together and restricts their movement. Currently, the suture is being modeled using 128 individual nodes with 127 separate Configurable Joints connecting them with the entire suture shown in figure 4. A sequence of more nodes that are smaller might allow for better behavior, but it will increase the computational load on the physics engine within Unity, which must respond immediately to changes in the system.

Parameter Variations

The parameters used for the Colliders, Rigid Bodies, Physics Materials, and Configurable Joints on each node have a significant impact on the behavior of the suture. My paper focused heavily on finding various parameter values that would work best and how changing parameters would affect the overall suture. With the current model and its most recent fine-tuned parameters, inexperienced users are occasionally able to use the virtual forceps to grasp the simulated suture and tie a knot in it. However, experienced users would be expected to consistently tie knots with the suture, although we have not been able to confirm this with the lack of microsurgeons on our team. The current model also interacts well with the forceps that the user uses as long as they are moving the forceps slowly and keeping them steady like in a real surgical scenario. In past models, joints would "break" under high stress, the suture would explode when Unity's physics engine could not find a way to bring two subsequent nodes back together, and the entire suture would fly off the screen trying to find the equilibrium point. While this can still happen, the frequency of it has been reduced, and the amount of stress required for it to explode has been increased.

Conclusion

Unity has proven itself to be an effective but overwhelmingly flexible system for modeling 3D phenomena. Modeling a microsurgical suture is especially difficult because it must balance flexibility and rigidity, distribute forces over its entire surface, and respond realistically to minute forces. We believe that a series of nodes connected through Configurable Joints is the most promising technique for building such a model and have been able to fine-tune it by experimenting with many of the parameters that are provided by that Unity component.

Recommended Citation

Stevens, J. (2023). Modeling a microsurgical suture in unity. *Made in Millersville Journal*, 2023. Retrieved from https://www.mimjournal.com/computer-science-2023