USING VR TO MODEL & CONTROL REALISTIC OCTOPUS EXPERIENCE

STUDENTS: MAYANK KUMAR, HELEN TRAN, ROBERT CRIST, TILBOON ELBERIER

UW REALITY LAB

27

Motivation and Objective

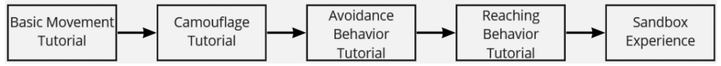
- Octopuses have extremely **complex nervous systems** that are vastly different than humans due to their *decentralized* nature
- This project seeks to create a Virtual Reality experience to help increase the user's **empathy for octopuses**
- With guidance from experts, our team is challenged to create an octopus model and implement AI models that *accurately* replicate the behavior patterns of real octopuses

Requirements

- An **immersive sandbox experience** that allows the user to explore the environment
- Create a *shader graph* that allows the model to **camouflage with the environment**
- Develop an *AI model* that enables the arms to **reach into & explore** interior space
- Implement an *AI model* that allows the arms to **bend away** from dangerous objects
- Integrate an existing *diver model* into the environment that can **interact with the user**

Introduction

In order to orient our users, we created a rough **tutorial** before they enter the sandbox experience. For the player to move on, they must **complete tasks** assigned by the diver. Each scene is crafted not only to show the *functionality* of the octopus which we created, but also to promote a sense of **empathy.**

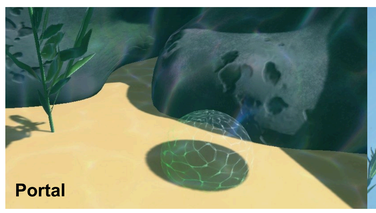


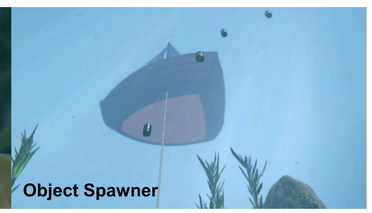
The tutorial is split up into **four distinct scenes**. To start, we provide the basic inputs for the user to *control* the octopus. Next, there are three scenes which **highlight the work** that we have completed for this project. Finally, once all tutorial scenes are finished, the user is able to roam freely in the **sandbox experience**.

General Scripts

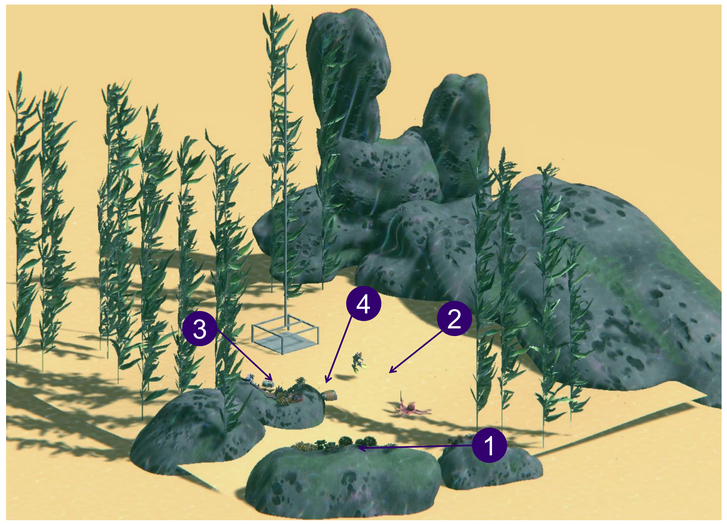
Throughout the development process, many additional scripts were identified in order to achieve our goals. The following list provides a summary of these scripts.

- **Scene Manager:** In order to create a tutorial for the experience, a script was created to sequence events in a particular order.
- **Audio Manager:** Manages all audio clips in a particular scene so that individual clips can be played when desired
- **Portals:** Used in the tutorial to transition from one scene to the next
- **Barriers:** A barrier script was created to confine the user to the map. Instead of being a hard wall, it acts more as a forcefield
- **Object Spawner:** Spawns random objects at the surface to create the illusion that a fisherman is throwing these objects overboard


Portal


Object Spawner

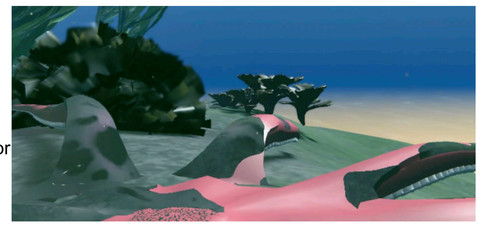
Map Layout, Model and Experiences



1 Camouflage Behavior

Camouflage enables the octopus to hide when it encounters predators

- Modeling of this behavior uses **real-time rendering** capability of *ShaderGraphics*
- **Dynamically collects data** from the environment & passes it to the shader to create new material in real-time
- **Each node** of *PBR graph* is **processed individually** to render the actual behavior
- **Colliders on the object** in the scene provide **trigger** information to start the camouflaging behavior of octopus



2 Diver Interaction

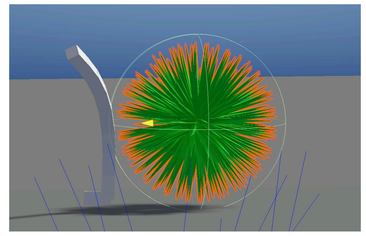
To increase empathy from a human's perspective towards octopus

- Used as a technique to simulate the human counterpart
- Implement vector *RayCast* functionality to simulate the **diver searching and following the player.**
- Implemented **animations** to communicate between diver and player
- **Diver leads the tutorial** which acquaints first time users into the VR experience
- Normal gameplay – the human diver follows the player as they navigate the scene within a target distance

3 Avoidance Behavior

AI implementation enables the arms to **recognize** and **bend away** from *dangerous species*

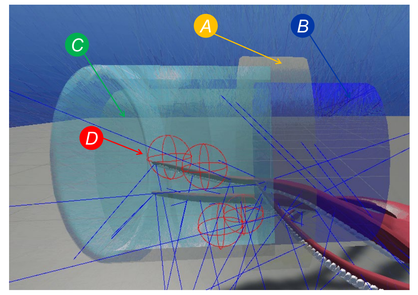
- Different types of *sea anemones* and *sea urchins* are implemented and inserted into the scene
- Strategy for the AI is to **orient the octopus's arms** in the opposite direction when they interact with threatening object, such as *sea urchins*
- Mesh collider (green): the surface of the object for arms to bend away from
- **Visual and audible feedback** for users to avoid the object in the next phase



4 Reaching Behavior

Checkpoints created with triggers are used to orient the tentacle reaching AI about where it currently is in relation to the opening

- **Opening Region** (Yellow *A*): The edge of the opening which tells the AI to orient the tip of the tentacle towards the opening
- **Opening** (Blue *B*): The physical opening which leads to the interior space and tells the AI to reach further inside the opening
- **Inside** (Green *C*)**:** The interior space which defines the region in which a **random point** (Red *D*) is generated for the AI to move the tentacle to



To ensure that the model is realistic:

- Strategic placement of checkpoints allows for the AI to engage only when a real octopus would be aware of an opening

Future Work & Conclusion

Although the core mechanics which we developed are in a solid state, there is still work that needs to be done that is outside the scope of our project:

- Further *refinement* towards the story aspect & creating defined experiences within the sandbox experience
- **Menus** as well as **UI** to provide feedback for the user
- Additional **audio and visual effects** for the environment
- Deliver this experience to **aquariums** as well as the **general consumers**

Our team accomplished the following goals for the project:

- **Completed identified missing mechanics** from Octopus model or environment
- **Created numerous scripts** which can be implemented in a variety of settings
- Assembled all work into a single **sandbox experience**
- **Created a foundation** which can be built upon after our team exits

Acknowledgements & References

Huge thanks to Terrell Strong, Natalie Burke, Dave Hunt, Dominic Sivitilli, Youjean Cho, Oliver Abate, Yerim Heo

- Unity API: https://docs.unity3d.com/ScriptReference/

ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY of WASHINGTON

ADVISORS: BARBARA MONES, TERRELL STRONG, DOMINIC SIVITILLI, PAYMAN ARABSHAHI, DANIEL KING
SPONSORS: PAUL G. ALLEN SCHOOL OF COMPUTER SCIENCE & ENGINEERING, UNIVERSITY OF WASHINGTON