Gavin McRoy

Dr. Sabarish V. Babu

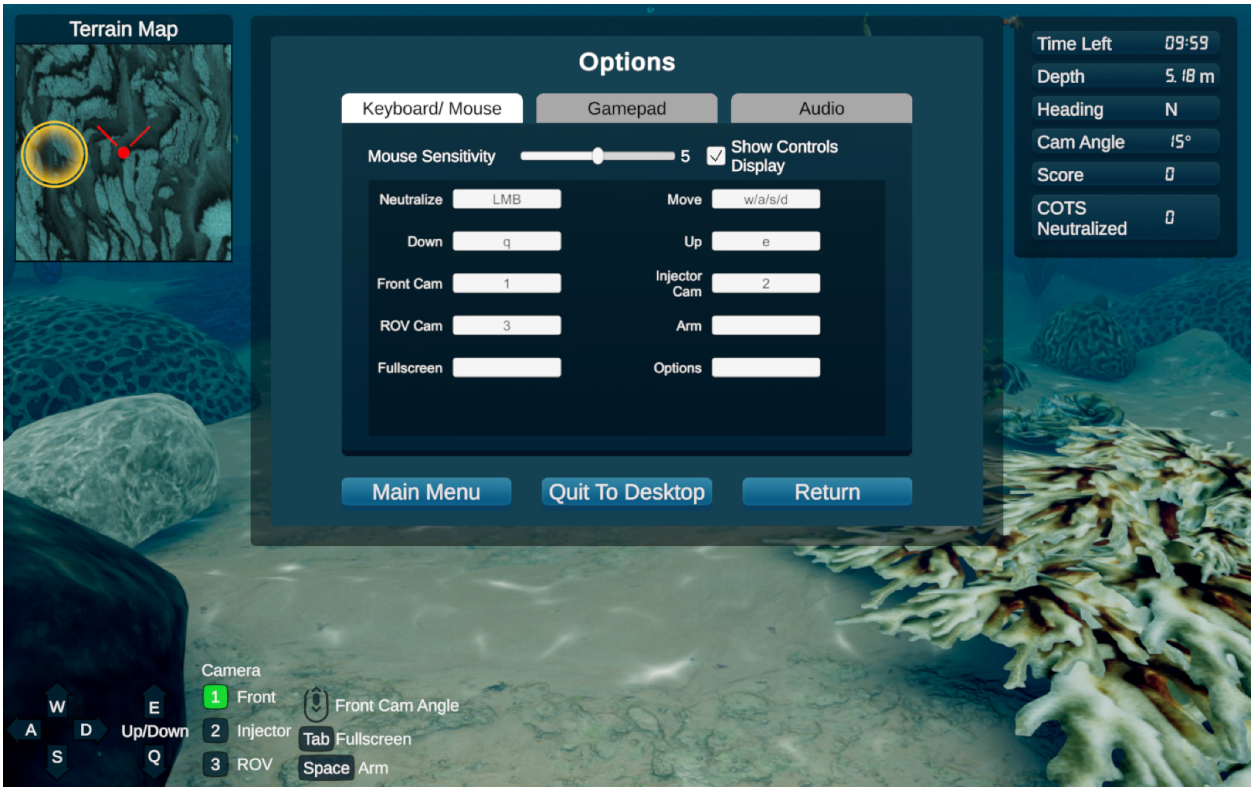
HCI – CPSC 4140/6140

July 15, 2021

Critique of the ROV Simulation

Discoverability

Discoverability, in essence, is what actions are possible within a particular system. The ROV simulation makes it reasonably easy to determine what actions are possible. Two features assist this. The most apparent and most customizable are the actual controls in the options menu. Here the user is informed of all possible actions they can take within the environment. They can see the different actions available and choose one that assists bests with achieving their goal. Secondly, in the bottom right, a small menu displays actions the user may take the most often, but not all the actions possible. The user can remove the controls on the bottom right if they find it necessary. Overall the ROV has great utilization of discovery by making it straightforward and intuitive to determine possible actions.



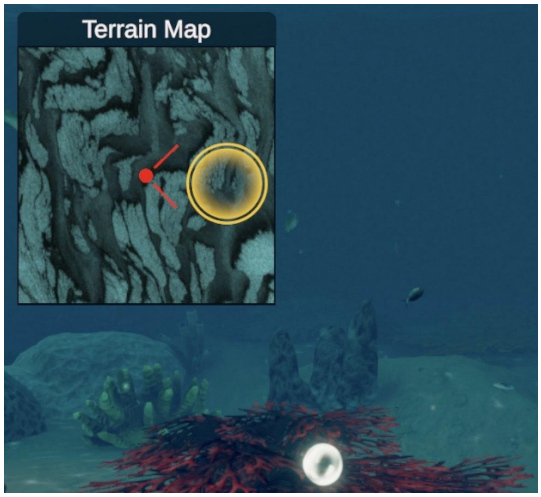
FeedBack

The simulation makes excellent use of feedback and has many different apparent examples of feedback. Feedback, in summary, is the communication of a result produced by an action. One of the more subtle feedback features is the point deduction system caused by bumping into coral. The simulation makes it very apparent by a giant -100 in red signaling the player has just lost points for running into coral. Hence the system communicated the result of the player bumping coral.

There is also a sound effect that plays when the player makes contact with any object. However, it can be slightly confusing since the same sound effect is played when you run into a non-Corel object and a coral object. So the user may associate the point deduction with the sound effect. And when they run into other objects and hear the sound effect and no point deduction, they may be confused. Other than this minor gripe, the feedback system is well built.

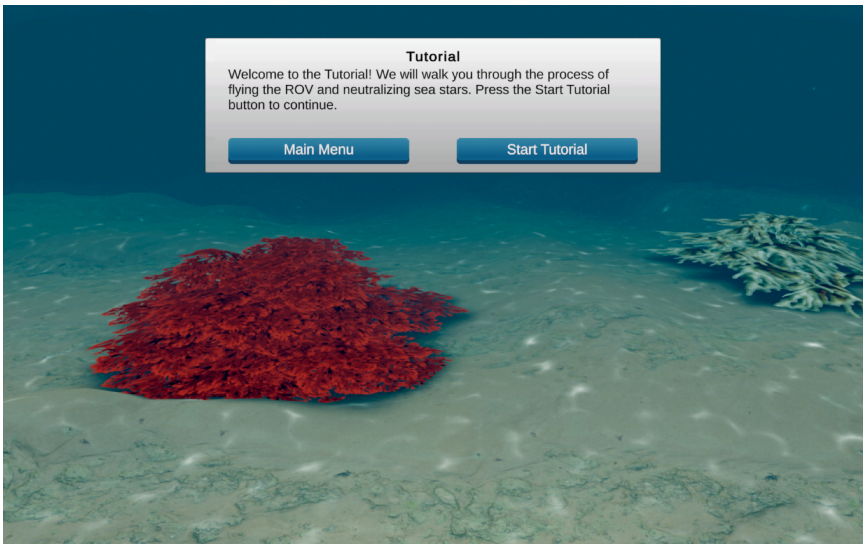
Feed-Forward

Feed-Forward is sometimes tricky to properly implement. Feed-Forward is a hint; its goal is to convey what possible actions are available to the user or what might be of interest. The Terrain Map feature is an excellent example of Feed-Forward in action. Notice the tiny orange circle that appears on the map. This hints to the player that this circle may be a location of interest. Also, note the two red arrows. They show the player which direction they are facing to take the proper action to reach the orange circle. When the player comes to the orange circle, there is an abundance of COT's to neutralize. The orange circle had successfully hinted toward the player that this area helps satisfy the player's goal of eliminating COT.



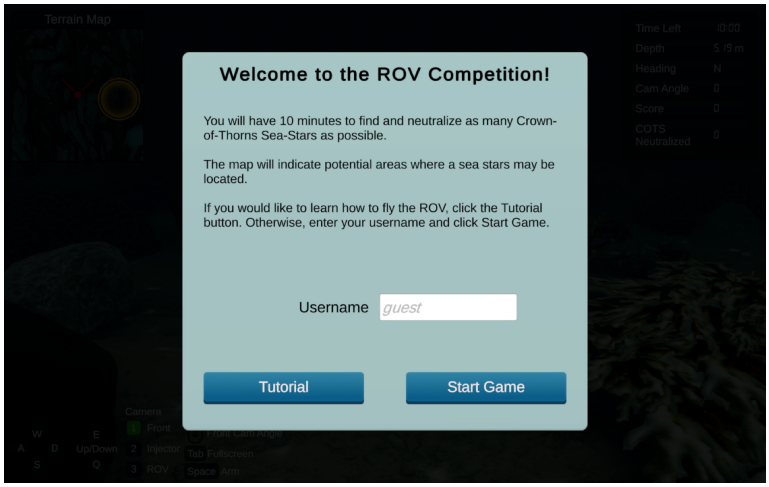
Conceptual Model

The conceptual model is a highly simplified explanation of how something works. A conceptual model mustn't conflict with a person's understanding of how things work. Such a conflict can cause confusion and an overall negative user experience. A good conceptual model allows us to predict the effect of our actions. Consider the tutorial in ROV simulation. The information provided is highly simplified yet intuitive. Nothing goes against our general cognitive understanding of things. The controls are mapped spatially, buttons do not give unexpected behavior, the fish behave expectedly, and all laws of physics seem to be obeyed as expected. All this is displayed and achieved without getting into technical jargon or unneeded information for the user. The explanation is not complete in the tutorial, and that is perfect because a good conceptual model does not have to be finished. They give the player enough information to predict the effect of certain actions. Overall the game's conceptual model is intuitive and well made.



Affordances

Affordances are the properties of an object that defines its possible uses. It's important to know that affordances exist even if their visibility is not apparent. The job of a good designer is to make the affordance visible through signifiers. Take a look at the menu the player sees before entering the game. The small box contains signifiers making it apparent typing is possible. Typing your name affords the user the ability to have their score saved on the leaderboard under their chosen name. This is only made apparent by the usage of signifiers which will be discussed later on. There are also other affordances available such as the two buttons and the information provided from the text. Clicking the button affords the user the ability to follow the tutorial or load into the game.



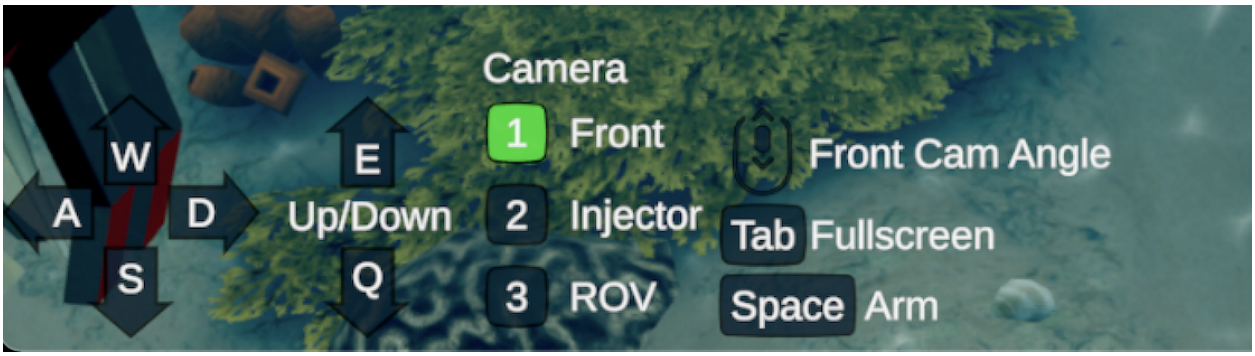
Signifiers

There are many clear signifiers in the ROV simulation. One particular example is the actual injection feature used to remove the starfish. As you get closer to the starfish, it is not immediately obvious what course of action you should take. But once you get to a certain distance away from the starfish, a small icon appears with a right-click. The icon signals that the user should press the right mouse button to initiate some action that assumingly helps achieve the user's goal. In essence, that is what signifiers are. They signal to the user that a particular action is possible. The mouse icon is a beautiful feature. If the mouse icon were not present, it would be challenging to determine what the user should do and close enough to initiate their desired

action. The signifier helps clarify this ambiguity by signaling to the player the proper step to take.

Mapping

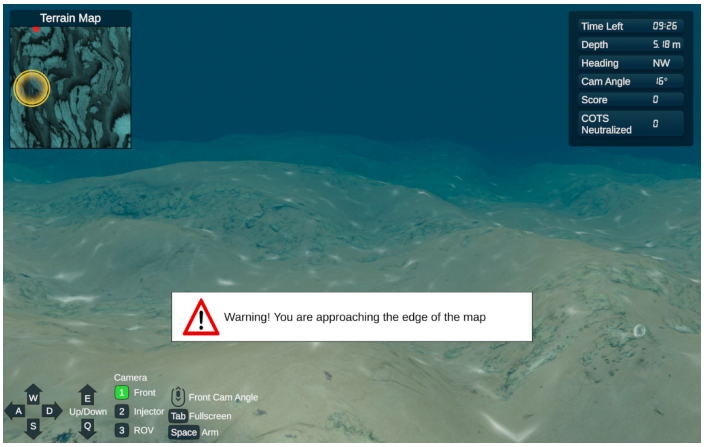
Mapping is an essential concept in design to aid in the user's understanding and learning system. Mapping is the relationship between controls and their effect. The ROV simulation does an excellent job with mapping by making the controls have spatial correspondence with the keyboard. Such mapping is called natural mapping. Notice that WASD on the keyboard form arrows in four directions. It's easy to learn that A is left, W is forward, D is right, and S is backward. Also, notice the up and down mapping. E to go up makes sense; it's the right-most key, and traveling right can be thought of as increasing height or distance. It is the same thing with the Q key; it's in the left direction, which can be thought of as decreasing height or distance. Imagine if these two controls were flipped so that E is down and Q is up. It would become substantially harder to learn to control the ROV unit, therefore a more lacking user experience overall. The ROV simulation utilizes natural mapping to create a spatial correspondence between the game and the keyboard, making it easy to learn the controls.



Constraint

The goal of constraints is to prevent a user from taking a particular action within a specific system. An undesirable goal within any game is the user escaping outside of the bounds and perhaps putting the system in an unstable state.

The ROV simulation handles this by placing constraints on the side of the map that prevents the player from crossing into an area they should not. There is also a notification that flickers, guaranteeing the player's attention. The notification informs them that they are approaching the edge of the map. This notifies the player they should begin changing directions to an area of the game capable of being explored. The box constrains the player to explorable



sections of the game only, preventing them from causing the system to crash or reach an unstable state.

Minimizing Gulf of Evaluation

The gulf of evaluation is when a user tries to figure out what happened when something failed. The goal is to have a small gulf of evaluation to work with ease and without multiple failures to get the desired result. When something goes wrong, you must notify the user of what went wrong and why. Ambiguous or non-existent notifiers leave people clueless and increase the gulf.

Take, for example, the following loading screen out of the ROV simulation. The bar's constant progression and information update below signifies that your desire to load something is processing. It's giving regular updates towards its progress and that no errors have occurred. It provides constant information regarding if the user's execution has worked successfully.

Imagine if the loading bar was taken away and just the word loading remained. It would be very confusing and uninformative to the user what is occurring. What seems like a small detail serves a crucial role in minimizing the gulf of evaluation.

Minimizing Gulf of Execution

The gulf of execution, in essence, means people are trying to figure out how something works. The goal behind the gulf of execution is it serves as the bridge to how people are to achieve a particular purpose. For example, if you had a website with a slider and wanted to go down, the slider bridges the gulf by inviting/suggesting you scroll down. When the gulf of execution works, people feel satisfaction and enjoyability. However, when the gulf of execution returns unexpected results or failures, people feel frustration and annoyance.

In the image below, multiple examples bridge the gulf of execution. Firstly the mouse arrow moves around the menu, inviting clicking, and when you slide over specific actions, a button lights up, inviting clicking. And what is achieved by clicking is clearly stated.

Like the previous example, imagine if these details were removed, such as the buttons lighting up when you slide over them. The system suddenly becomes much harder to use. The gulf of execution is instead maximized, making for amnegative user interaction with the system.


M8-R SIM
CORAL IN CRISIS
Contain the Crown-of-Thorns-Sea-Stars!
Introduction
New Game
Leaderboard
Resources
Credits
Quit
MATE II
CA²VES
CUCWD
NCAT