06/27/24

IT209 Final Project

Melissa Cianfarano

Project Proposal

As an avid animal lover and owner, I care about the physical health of my pets. I have a 12-year-old cat who is limited in his activity due to age, motivation, and dogs taking up the entire domain. It would be nice to have an interactive toy that my cat can chase and be interested in to keep him in good health. The finch would be perfect for this task. Rigging the finch to have a string of feathers following it and setting it to roam a room while avoiding obstacles and walls would be a perfect way to entice the cat to play and get some exercise. The biggest issues to face will be cord management and dealing with noise levels output by the finch robot.

The Finch actuators include wheel motors, LED lights, and noise capabilities. The wheel motors will be key and varying speeds will be useful for keeping the cat entertained. Noise and buzzers might deter the cat, so a fun light show will be entertainment enough for the pet owner. The finch has a couple of sensors which include light, temperature, obstacle, and accelerometers. The most needed sensor would be obstacle sensors which will help to keep the fun going by avoiding getting stuck. The light sensors will also come in handy and deter the Finch from entering areas where it might get trapped.

In the growing trends of robotics, there is a section of robotics as a service or RaaS. This includes a robot as a service that will help with a designated task, such as weeding and spraying farms. This pet toy is a Raas as it will keep owner and pet entertained by creating an interesting customizable toy which will aid in keeping the pet in better health by providing exercise.

Analysis

While there are automated cat toys on the market, this robot option is clearly more fun. We all know our pets best and what movements and strategies they like to engage with. With this option we can customize the toy with each pet in mind. The obstacle deterrent is also a large benefit as standard industry toys will just get stuck and be rendered useless. This toy will use a closed-loop control system. The closed-loop system will be the most useful for the pet toy as it will be dealing with varying unpredictable obstacles and environments. It’ll need to sense and navigate past them. Utilizing light sensors and infrared sensors will provide the feedback the robot needs to avoid becoming stuck on random objects. This combination gives the user more satisfaction than a set it and forget it type toy. The pet industry is large, and a lot of pet lovers will spend endless amounts of money to keep their pets happy.

When comparing this prototype to other robotic pet toys on the market, this one will have a clear advantage. If you look at a HEXBUG Mouse Robotic Cat Toy, it includes features like the Finch prototype. One major advantage is that our technology will be strong enough to work on low carpets as well. Looking at HEXBUG reviews, it is evident a lot of customers complain about the HEXBUG only lasting for a day or two before breaking. The Finch will be a superior product made with better quality parts. Another pet exercise robot is the Pet Cube. This is a stationary robot with a camera and laser toy. When the robot senses motion, the laser will activate which gives pets something to play with. While a good idea, this is ultimately harmful to pets as it incites neurotic behaviors with light and lasers. Another mobile robot is the Robotic Pet Interaction Toy by sharper image. This robot has a camera, microphone, speaker, treat dispenser, and laser pointer. While it has more mobility like our Finch prototype, this robot is bulky, expensive, and not nearly as encouraging for play as our prototype.

Software Design

A screenshot of a computer

Description automatically generated

A diagram of a diagram

Description automatically generated

Explanation of the Design

When reviewing my robot’s application design, all of the major components are included into the main class. This covers both light and infrared sensors connecting with the wheel actuators. There are private classes to collect data from the robots’ sensors, and public classes to address expected behavior for this data input. Expectations were generally met with the physical features being the most problematic. Having a robot attached to a cord limits the space to be used for the robot. The robot was unable to drag any real pet toys of interest as well. Remediation would be updating the prototype to Bluetooth capabilities as well as increasing the tow capacity.

Subsumption is distributing a robots control system into simpler segments. This prototype is broken down into smaller segments for control of light sensors, wheel actuators, and infrared sensors. All segments will come together to produce the overall product. The simulation of our robot was performing in Finch’s visual programming environment using block coding. This created the virtual environment needed to implement all of the behaviors in the robot with an easy-to-use interface.

(XML CODE WITH COMMENTS AFTER RESOURCES PAGE/ BULK CUT OUT TO HIGHLIGHT ADDED (RED) COMMENTS ONLY. WILL UPLOAD ORIGINAL XML FILE AS WELL!)

Resources:

StartUs Insights. (2023, October 12). *Top 10 Robotics Trends & Innovations in 2024 | StartUS Insights*. <https://www.startus-insights.com/innovators-guide/robotics-trends-innovation/>

*HEXBUG Mouse Robotic Cat Toy*. (n.d.). HexBug. Retrieved June 23, 2024, from <https://www.hexbug.com/hexbug-mouse-robotic-cat-toy-white.html>

Petcube. (n.d.-b). *PetCube Play - interactive Wi-Fi Pet camera to monitor and play*. Petcube Web-site. <https://petcube.com/play/>

*Robotic pet interaction toy @ SharperImage.com*. (n.d.). https://www.sharperimage.com/view/product/Robotic+Pet+Interaction+Toy/208926?pc=20GOOGLE&utm\_source=Bing&utm\_medium=CPC&utm\_campaign=&msclkid=14de8007834f1f108e63e673f56fef21

<script x="371" y="26.999999999997954">

<block s="receiveKey"> // SET UP INPUT KEY TO START FINCH

<l>

<option>up arrow</option> // START = UP ARROW

</l>

<list/>

</block>

<block s="doUntil"> // REPEAT BLOCK UNTIL

<block s="reportKeyPressed">

<l>

<option>space</option> // SPACE KEY IS PRESSED

</l>

</block>

<script>

<block s="doIfElse"> // IF ELSE BLOCK FOR RIGHT LIGHT SENSOR

<block s="reportVariadicLessThan">

<list>

<custom-block s="Finch Right Light Sensor"/> // IF RIGHT LIGHT MEASUREMENT IS LESS THAN 1

<l>1</l>

</list>

</block>

<script>

<custom-block s="Move Finch Left: %n Right: %n"> // MOVE FINCH BACKWARDS AND TURN FOR 2.5 SECONDS

<l>-50</l>

<l>-100</l>

</custom-block>

<block s="doWait">

<l>2.5</l>

</block>

</script>

<script>

<custom-block s="Move Finch Left: %n Right: %n"> // MOVE FORWARD

<l>200</l>

<l>200</l>

</custom-block>

</script>

</block>

<block s="doIfElse"> // IF ELSE BLOCK FOR RIGHT OBSTACLE SENSOR

<custom-block s="Finch Right Obstacle"/> // IF RIGHT SENSOR ACTIVATED

<script>

<custom-block s="Move Finch Left: %n Right: %n"> // FINCH MOVES BACK AND TURNS FOR 1 SECOND

<l>-50</l>

<l>-100</l>

</custom-block>

<block s="doWait">

<l>1</l>

</block>

</script>

<script>

<custom-block s="Move Finch Left: %n Right: %n"> // ELSE MOVES FORWARD

<l>200</l>

<l>200</l>

</custom-block>

<block s="doIfElse">

<block s="reportVariadicLessThan">

<list>

<custom-block s="Finch Left Light Sensor"/> // IF LEFT LIGHT SENSOR MEASURES LESS THAN 1

<l>1</l>

</list>

</block>

<script>

<custom-block s="Move Finch Left: %n Right: %n"> // FINCH MOVES BACKWARDS AND TURNS FOR 2.5 SECONDS

<l>-100</l>

<l>-50</l>

</custom-block>

<block s="doWait">

<l>2.5</l>

</block>

</script>

<script>

<custom-block s="Move Finch Left: %n Right: %n"> // ELSE MOVES FORWARD

<l>200</l>

<l>200</l>

</custom-block>

</script>

</block>

</script>

</block>

<block s="doIfElse"> // IF LEFT OBSTACLE DETECTED

<custom-block s="Finch Left Obstacle"/>

<script>

<custom-block s="Move Finch Left: %n Right: %n"> // FINCH MOVES BACKWARDS AND TURNS FOR 1 SECOND

<l>-100</l>

<l>-50</l>

</custom-block>

<block s="doWait">

<l>1</l>

</block>

</script>

<script>

<custom-block s="Move Finch Left: %n Right: %n"> // ELSE FINCH MOVES FORWARD

<l>200</l>

<l>200</l>

</custom-block>

</script>

</block>

</script>

</block>

<custom-block s="Stop Finch"/> // FINCH STOPS WHEN SPACE KEY PRESSED

</script>

</scripts>

</sprite>

<watcher scope="Stage" s="getLastAnswer" style="normal" x="0.15625" y="1.7968751562500103" color="4,148,220" hidden="true"/>

<watcher scope="Sprite" s="getScale" style="normal" x="0.15625" y="2.1250001874999924" color="143,86,227" hidden="true"/>

</sprites>

</stage>

<variables/>

</scene>

</scenes>

</project>