Computational Thinking (Gamified)

Square

Description automatically generated

Learning Objectives:

To teach the students computational thinking through the use of the web portal and the robotic car. The following concepts are components of Computational Thinking:

|  |  |  |
| --- | --- | --- |
| Concept | Description | Objective |
| Decomposition | Breaking down complex problems into simpler components | To understand and evaluate the instructions and capabilities of the car and web portal to achieve the stated objectives. |
| Pattern Recognition | Identify patterns in solutions for future application | Students will learn to program and identify similarities between the functions and their results. |
| Generalisation | Usage of previous knowledge to create new solutions | Students are to build upon previous knowledge to formulate solutions to new problems and conditions |
| Abstraction | Simplifying the problem to extract the key components | Students shall identify the crucial information found on the physical map; to identify key information for a successful execution |
| Algorithms | Sequential processing of information | Students shall input commands for the robotic car in a logical-sequence. |
| Logical Reasoning | Reviewing decision making |
| Evaluation | Compare and contrast of results against processes | Students should be able to compare their inputs to the results of the robotic car |

Background:

You are a NASA personnel tasked with commandeering the Robotic Car, Yeet. Before being able to do so, you will be training in a simulation to be come proficient in understanding the various controls provided to you in directing Yeet across 3 perilous maps.

Before embarking on the final test, real-life navigation

Restrictions:

You will not be able to physically interact with Yeet apart from the two buttons onboard (as seen in the diagram).

< TEACHING PLAN >

SESSION 1: LINKING ROBOTIC CAR & WEB PORTAL

Showcase users by preloading a *default* action plan to demonstrate the car’s capabilities

|  |  |  |  |
| --- | --- | --- | --- |
| FUNCTION | API | PARAMETERS | RETURNS |
| Default | DEF[] | - | Car runs default routine |

Diagram, schematic

Description automatically generated

SESSION 2: FEATURE INTRODUCTION

To introduce the following features, ask users to test out the following commands (alone).

|  |  |  |  |
| --- | --- | --- | --- |
| FUNCTION | API | PARAMETERS | RETURNS |
| Movement | MOV[a:b:c] | a – direction (1(up),2(down),3(left),4(right))  b – speed (1(slow),2(fast))  c – duration (5s - 10s) | car moves accordingly |
| Ultrasonic Sensor | USM[] | - | distance (cm) |
| LED | LED[x] | x – 1 to 7 (RGB of the car) | Car LED changes |
| IR Sensor | IRS[] | - | BRIGHT or DARK |
| Buzzer | BUZ[x] | x – select sound (1(start),2(stop),3(cute chime)) | buzzer activates accordingly |

SESSION 3: Delay and Loops

To teach the users about delay and looping commands. For this session we will want users to make their car travel around the parameters three times, with a small 5 second delay in between.

ROUND 1 -> DELAY -> ROUND 2 -> DELAY -> ROUND 3

|  |  |  |  |
| --- | --- | --- | --- |
| FUNCTION | API | PARAMETERS | RETURNS |
| Delay/Wait | DEL[x] | x – duration (5s – 10s) | car will wait accordingly |
| Loop | LOP[a][b] | a – Number of Loops (1 - 5)  b – Command to be executed | car will execute b for a times |

SESSION 4: Conditional Movement

< DEVELOPER >

Block Based Programming

1. Move <direction> < duration> <speed>
2. IR Sensor <White/Black>
3. USM <Value>
4. Buzzer <start sound> <stop sound> <select\_tune>

IF

WHILE

Must Have

Good to Have

Optional