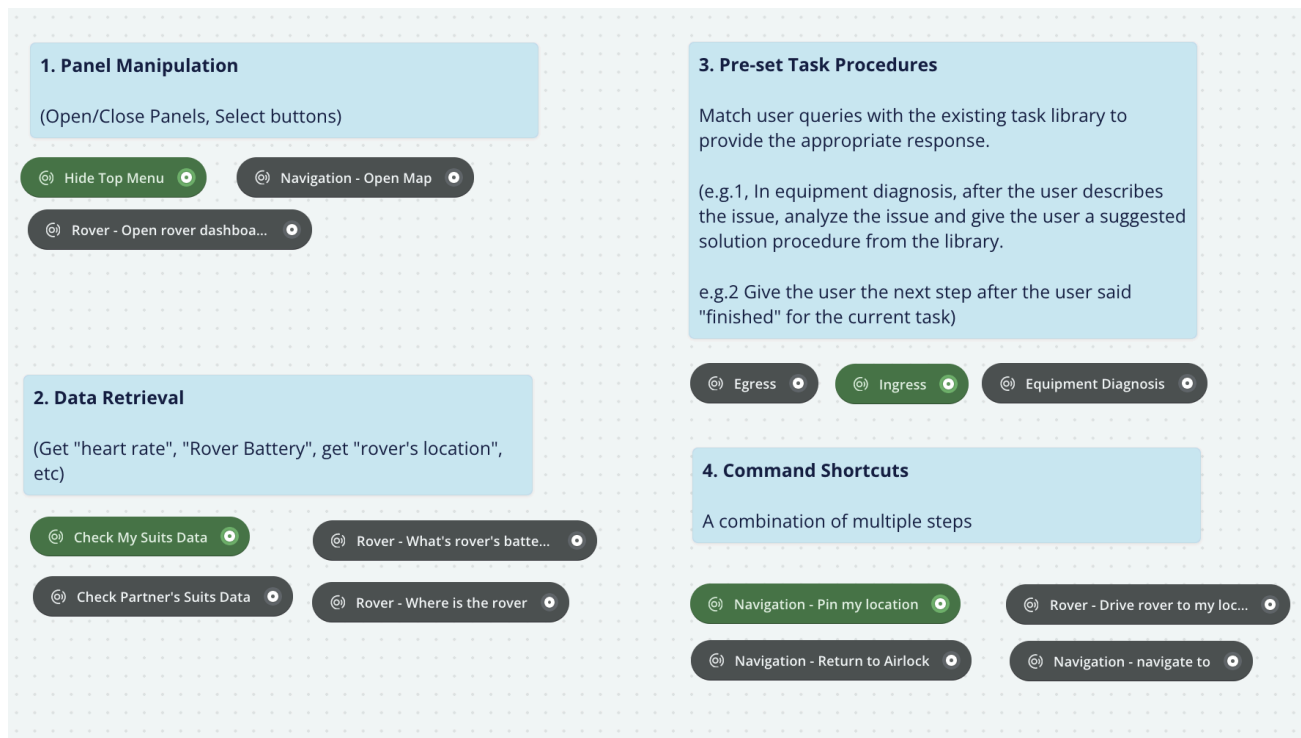


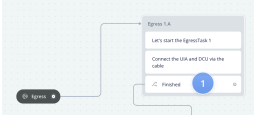
## Design:

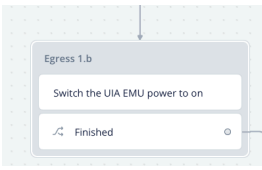
[https://creator.voiceflow.com/workspace/accept-invite?inviteToken=eyJhbGciOiJIc2EiLCJlbnMiOiJBbmJlU2R0NNln0..mfDackXjX38lfm-F.AkXovCjaKeTMly21pAmZb6Q4XdjYtBdwshfKUrIW7HN33OECepC9xAI07FKTqiVc7efQ6\\_klF6MQuFI1GRJR1BDPmSewKH\\_NEXc1r3xcj6MfozEOTA.v4L0KQYRUqt9fALXiT8kNA](https://creator.voiceflow.com/workspace/accept-invite?inviteToken=eyJhbGciOiJIc2EiLCJlbnMiOiJBbmJlU2R0NNln0..mfDackXjX38lfm-F.AkXovCjaKeTMly21pAmZb6Q4XdjYtBdwshfKUrIW7HN33OECepC9xAI07FKTqiVc7efQ6_klF6MQuFI1GRJR1BDPmSewKH_NEXc1r3xcj6MfozEOTA.v4L0KQYRUqt9fALXiT8kNA)

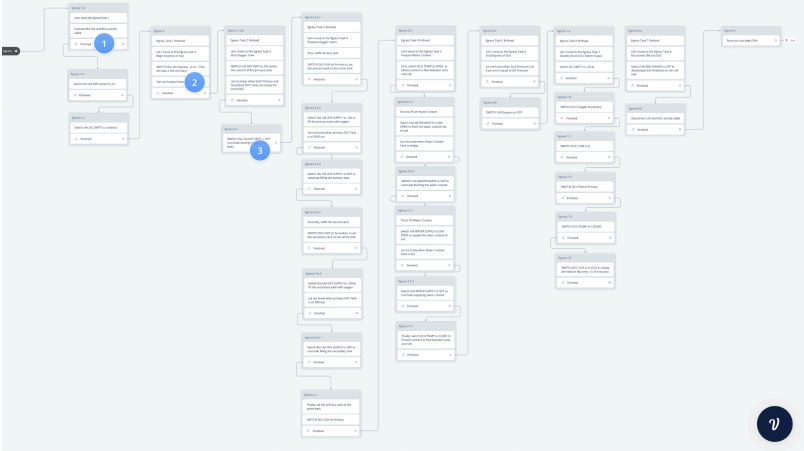
## Task Flows:




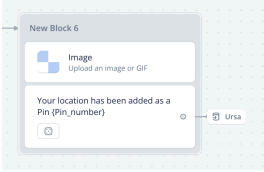
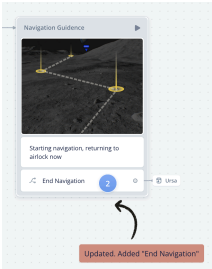
## LLM Function Calling Version 2

| Task category | LLM example prompts   | LLM function call descriptions                               | UI workflow   | LMCC -> HMD function requirement  |
|---------------|---|--|---|---|
| Egress        | 1. Perform the specified subtask <ul style="list-style-type: none"> <li>Astronaut input = Prompt given to LLM:</li> </ul> | <pre>{ "type": "function",   "function": {     "name":</pre> | 1a  | Function name:<br><b>on_egress_menu_do_subtask_1a()</b> <ul style="list-style-type: none"> <li>Input: None</li> </ul> |

|  |  |  |  |   |
|--|--|--|--|---|
|  | <p>“Perform egress 1a”</p> <p>2. Perform the <b>next</b> subtask</p> <ul style="list-style-type: none"> <li>- Astronaut input: “Perform next task”</li> <li>- Prompt given to LLM: “Currently I am on subtask 1a (<i>current subtask, stored as a variable in the system</i>), following the given egress subtask order as 1a, 1b, 1c, 2, 3ab, 3c (<i>order of subtask, stored as a string variable in the system</i>), perform next task</li> </ul> | <p><b>"on_egress_menu_do_subtask_1a",</b></p> <p>"Description": "perform on egress subtask 1a, <i>which is to ask the user to connect the UIA and DCU via the cable</i>",</p> <p>"parameters": {</p> <p>"type": "object",</p> <p>"properties": {},</p> <p>"required": [],</p> <p>},</p> <p>}</p> |  | <ul style="list-style-type: none"> <li>- Behaviour: read out the text descriptions for this task</li> <li>- Output: None</li> </ul> <p>Example:</p> <pre>def on_egress_menu_do_subtask_1a():     voice_output("Let's start on the EgressTask 1, connect the UIA and DCU via the cable")     return</pre>  |
|  | <p>3. Perform the <b>previous</b> subtask</p> <ul style="list-style-type: none"> <li>- Astronaut input: “perform previous task”</li> <li>- Prompt given to LLM: “Currently I am on subtask 1b (<i>current subtask, stored as a variable in the system</i>), following the given egress subtask order as 1a, 1b, 1c, 2, 3ab, 3c (<i>order of subtask, stored</i></li> </ul>   | <p>{ "type": "function",</p> <p>"function": {</p> <p>"name": <b>"on_egress_menu_do_subtask_1b",</b></p> <p>"Description": "perform on egress subtask 1b, <i>which is to ask the user to turn on the UIA EMU power</i>",</p>  | <p>1b</p>  | <p>Function name: <b>on_egress_menu_do_subtask_1b()</b></p> <ul style="list-style-type: none"> <li>- Input: None</li> <li>- Behaviour: read out the text descriptions for this task</li> <li>- Output: None</li> </ul> <p>Example:</p> <pre>def on_egress_menu_do_subtask_1b():     voice_output("Swicth the UIA EMU power to on")     return</pre> |

|            |  |   |  |  |
|------------|--|---|--|--|
|            | <p><i>as a string variable in the system</i>),<br/>perform previous task</p> <p>4. Perform the <b>current</b> subtask</p> <ul style="list-style-type: none"><li>- Astronaut input: “repeat this task”</li><li>- Prompt given to LLM: “Currently I am on subtask 1b (<i>current subtask, stored as a variable in the system</i>), following the given egress subtask order as 1a, 1b, 1c, 2, 3ab, 3c (<i>order of subtask, stored as a string variable in the system</i>), repeat this task</li></ul> | <pre>"parameters": {<br/>  "type": "object",<br/>  "properties": {},<br/>  "required": [],<br/>}</pre>  |  |  |
|            |  | <p>Repeat the above for Egress task 1-9</p>    |  |  |
| Navigation | <p>Astronaut input =<br/>Prompt given to LLM:<br/>“open map”</p>   | <pre>{ "type":<br/>  "function",<br/>  "function": {<br/>    "name":<br/>      "open_map",<br/>    "Description": "open map",<br/>    "parameters": {<br/>      "type": "object",</pre> |  | <pre>open_map()<br/>- Input: None<br/>- Behaviour: Open the<br/>  map and show it on<br/>  the screen<br/>- Output: None</pre> |

|  |  |  |  |   |
|--|--|--|--|---|
|  |  | <pre> "properties": {},  "required": [],       },     },   } </pre>  |  |   |
|  | <p>Astronaut input =<br/>Prompt given to LLM:<br/>"remove pin 3"</p> | <pre> {   "type": "function",  "function": {  "name": "open_map",  "description": "Remove pin with the given pin numbe",  "parameters": {  "pin_number": {  "type": "integer",  "description": number of the pin to remove",       },     },  "required": ["pin_number"],       },     },   } </pre> |  | <pre> remove_pin(pin_number) </pre> <ul style="list-style-type: none"> <li>- Input: int pin_numbe<br/>-- the pin to be removed</li> <li>- Behaviour: Remove the pin from the map, update the map<br/>(open map if the map is not opened yet)</li> <li>- Output: None</li> </ul> |

|  |   |   |   |   |
|--|---|---|---|---|
|  | <p>1. Astronaut specifies the pin number for the current location:</p> <ul style="list-style-type: none"> <li>- Astronaut input = Prompt given to LLM: “pin my location as 3”</li> </ul> <p>2. Astronaut wants to pin the <b>current</b> location</p> <ul style="list-style-type: none"> <li>- Astronaut input: “pin my location”</li> <li>- Prompt given to LLM: “pin my location as 3 (increment the current pin number by 1)”</li> </ul> | <pre>{   "type": "function", "function": { "name": "pin_my_location", "description": "Pin with the current location on map with the given pin number", "parameters": { "pin_number": { "type": "integer", "description": number of the pin to be added", }, }, "required": ["pin_number"], }, }</pre> |                             | <p>pin_my_location(pin_number)</p> <ul style="list-style-type: none"> <li>- Input: int pin_number -- the pin number to be added</li> <li>- Behaviour: Retrieve the current location on the map, add the pin to the map, update the map</li> <li>- Output: None</li> </ul> |
|  | <p>Astronaut input = Prompt given to LLM: “return to airlock”</p>   | <pre>{   "type": "function", "function": { "name":</pre>  |  | <p>return_to_airlock()</p> <ul style="list-style-type: none"> <li>- Input: None</li> <li>- Behaviour: <ol style="list-style-type: none"> <li>1. While the astronaut is walking on the worksite, record their</li> </ol> </li> </ul>                                       |

|  |  |  |  |   |
|--|--|--|--|---|
|  |  | <p>"return_to_airlock",</p> <p>"description":<br/>"Provide visual feature guidance (regarding real world objects) for astronaut so they can return to airlock following the same path",</p> <p>"parameters": {</p> <p>"type": "object",</p> <p>"properties": {},</p> <p>"required": [],</p> <p>},</p> <p>}</p> |  | <p>GPS location as a "pin" every 1-3 minutes. Note that there's no need to display those pins on the map.</p> <ol style="list-style-type: none"> <li>2. When the astronaut queries "return_to_airlock", <b>retrieve those pins</b> and assume it's the best path back.</li> <li>3. No need to show path in the map, go directly to navigation (visual elements guidance in the real world)</li> </ol> <p>- Output: None</p> |
|--|--|--|--|---|

## Demo on Gorilla

The screenshot shows the Gorilla OpenFunctions-v2 interface. At the top, the model is set to 'Gorilla OpenFunctions-v2' and the temperature is 0.7. A red arrow points to the prompt: 'Currently I am on subtask 1a, the egress subtask order as 1a, 1b, 1c, 2, 3ab, 3c, show me the next task'. A blue arrow points to the 'Astronaut voice-to-text input' button. Below the prompt, a JSON object is displayed, which is a function definition for 'on\_egress\_menu\_do\_subtask\_1b'. A red arrow points to the 'Function calling description' field. To the right, the function name 'on\_egress\_menu\_do\_subtask\_1b()' is shown, followed by 'The function called'. Below this, the function's JSON definition is shown. At the bottom right, there are buttons for 'thumbs up', 'thumbs down', and 'Report Issue'.

Model: Gorilla OpenFunctions-v2

Temperature: 0.7

Added to all egress related prompts: Currently I am on subtask 1a, the egress subtask order as 1a, 1b, 1c, 2, 3ab, 3c, show me the next task

Astronaut voice-to-text input

Function calling description: { "type": "function", "function": { "name": "on\_egress\_menu\_do\_subtask\_1b", "Description": "perform on egress subtask 1b, which is to ask the user to turn on the UIA EMU power", "parameters": { "type": "object", "properties": {}, "required": [] }, }, }, { "type": "function",

on\_egress\_menu\_do\_subtask\_1b() The function called

{ "name": "on\_egress\_menu\_do\_subtask\_1b", "arguments": {} }

thumbs up thumbs down Report Issue

- Currently I am on subtask 1a, perform the next subtask following the given egress subtask order as 1a, 1b, 1c, 2, 3ab, 3c and remove pin 5 and pin my location as 3 and return to airlock

```
{ "type": "function",
  "function": {
    "name": "on_egress_menu_do_subtask_1a",
    "Description": "perform on egress subtask 1a, which is to ask the user to connect the UIA and DCU via the cable",
    "parameters": {
      "type": "object",
      "properties": {},
      "required": [],
    },
  },
}
```

```

{ "type": "function",
  "function": {
    "name": "on_egress_menu_do_subtask_1b",
    "Description": "perform on egress subtask 1b, which is to ask the user to turn on
the UIA EMU power",
    "parameters": {
      "type": "object",
      "properties": {},
      "required": [],
    },
  },
}
{ "type": "function",
  "function": {
    "name": "open_map",
    "Description": "open map",
    "parameters": {
      "type": "object",
      "properties": {},
      "required": [],
    },
  },
}
{
  "type": "function",
  "function": {
    "name": "open_map",
    "description": "Remove pin with the given pin numbe",
    "parameters": {
      "pin_number": {
        "type": "integer",
        "description": "number of the pin to remove",
      },
    },
    "required": ["pin_number"],
  },
}
{
  "type": "function",
  "function": {
    "name": "pin_my_location",
    "description": "Pin with the current location on map with the given pin numbe",
    "parameters": {

```



```

    "pin_number": {
      "type": "integer",
      "description": "number of the pin to be added",
    },
  },
  "required": ["pin_number"],
},
}
{
  "type": "function",
  "function": {
    "name": "return_to_airlock",
    "description": "Provide visual feature guidance (regarding real world objects) for
astronaut so they can return to airlock following the same path",
    "parameters": {
      "type": "object",
      "properties": {},
      "required": [],
    },
  },
}

```

-

## Version 1:

### Panel Controls

Hide menu

```
{
```

```

    "type": "function",
    "function": {
      "name": "hide_menu",
      "description": "Hide the menu in the UI",
      "parameters": {
        "type": "object",
        "properties": {},
        "required": [],
      },
    },
  },
}

```

Functions needed from HMD:

- hide\_top\_menu()
  - When called, just hide the top menu
  - Return when done

## Suits Data

Interact with my suit

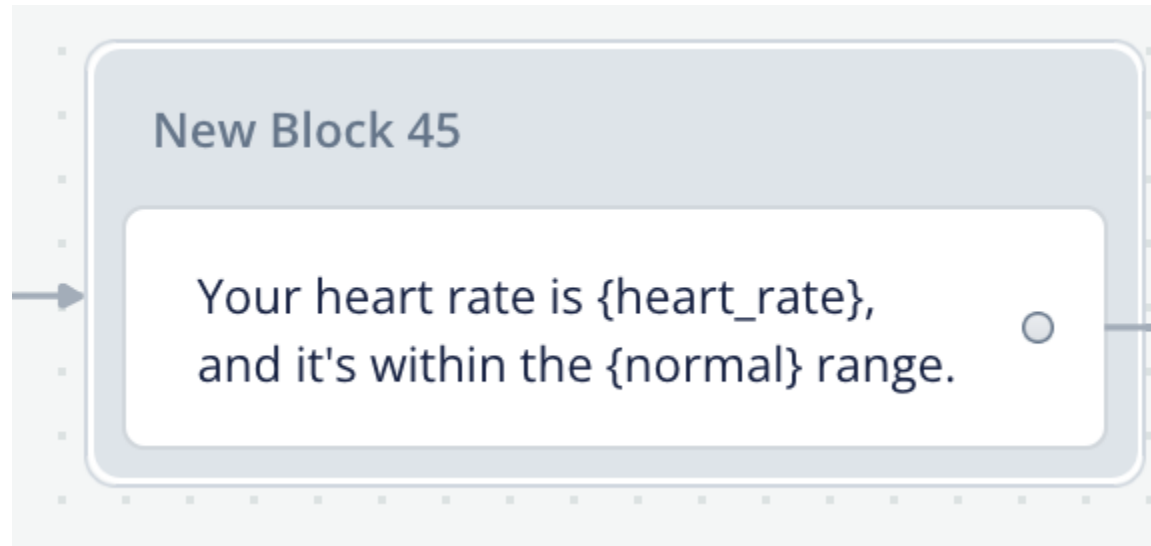
```

{
  "type": "function",
  "function": {
    "name": "interact_my_suit",
    "description": "Interact with my suit by opening it or requesting it to return data about
the suit",
    "parameters": {
      "type": "object",
      "properties": {
        "open_suit": {
          "type": "boolean",
          "description": "Open my suit or not",
        },
        "suit_data_type": {
          "type": "string",
          "description": "my suit's data type, e.g. heart rate",
        },
      },
    },
    "required": [],
  },
}

```

Functions needed from HMD:

- Open\_my\_suit()
  - Open a UI panel showing astronaut's biological data
  - Return when done
- Check\_my\_suits\_data(suit\_data\_type)
  - Input:
    - Suit\_data\_type: a string of the type of data to be retrieved e.g. heart\_rate
  - Compare it to a normal range (hardcoded for this type of data)
  - Output:



Interact with partner's suit

```
{
  "type": "function",
  "function": {
    "name": "interact_partner_suit",
    "description": "Interact with partner's suit by opening it or requesting it to return data about the suit",
    "parameters": {
      "type": "object",
      "properties": {
        "open_suit": {
          "type": "boolean",
          "description": "Open partner's suit or not",
        },
        "suit_data_type": {
          "type": "string",
          "description": "partner's suit's data type, e.g. heart rate",
        },
      },
    },
  },
}
```

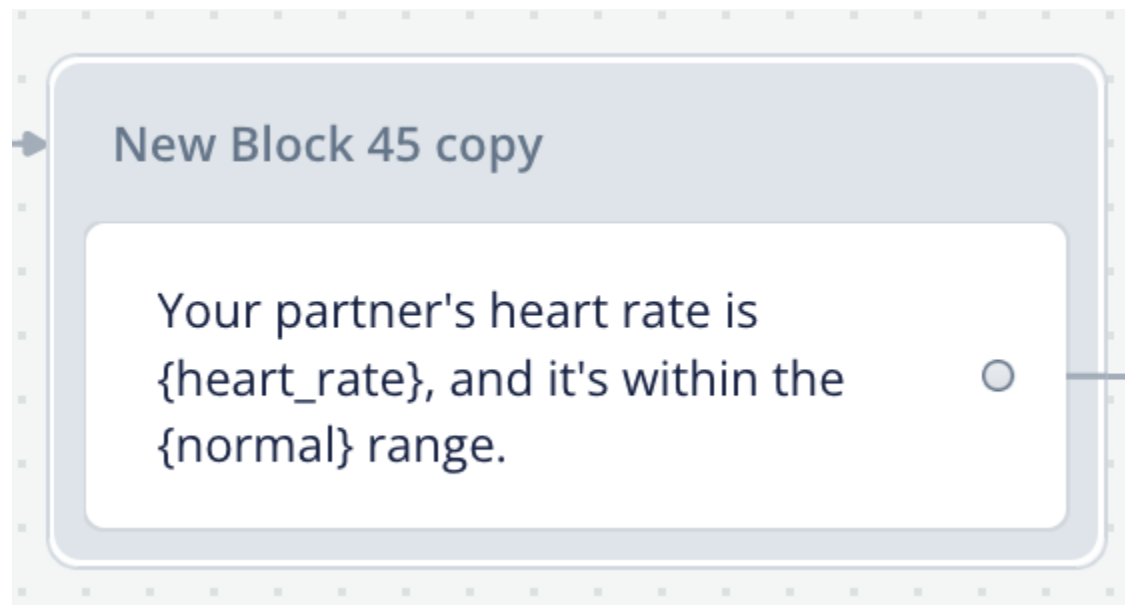
```

    },
    "required": [],
  },
},
}

```

Functions needed from HMD:

- Open\_partner\_suit()
  - Open partner suit
  - Return when done
- Check\_partner\_suits\_data(suit\_data\_type)
  - Input:
    - Suit\_data\_type: a string of the type of data to be retrieved e.g. heart\_rate
  - Compare it to a normal range (hardcoded for this type of data)
  - Output:



## Ingress task

*Note: according to [voiceflow](#) the ingress task has multiple substeps, no variables need to be passed in for each subtask.*

```

{
  "type": "function",
  "function": {
    "name": "run_ingress",
    "description": "Conduct the ingress task given stage number or move forward
instructions",
    "parameters": {

```

```

    "type": "object",
    "properties": {
      "subtask_redirect": {
        "type": "string",
        "description": "The action of moving to a given stage, the previous stage,
repeating the current stage, or moving to the next stage in the ingress task",
        "enum": ["1a", "1b", "1c", "previous", "next", "repeat", "start"]},
    }
  },
  "required": [subtask_redirect],
},
}

```

*Notes: If this function is called with no specification on the stage → start from beginning. If a stage number is provided such as 1c, then start at that stage. If the word “previous”, “next”, “repeat” are said, then move backward, forward, or repeat the current stage again.*

HMD:

- Run\_ingress\_task(subtask\_redirect)
  - Input:
    - subtask\_redirect → one of ["1a", "1b", "1c", "previous", "next", "repeat", "start"]
  - If this function is called with no specification on the stage → start from beginning
  - If a stage number is provided such as 1c, then start at that stage.
  - If the word “previous”, “next”, “repeat” are said, then retrieve the current stage number (stored as a variable), move backward, forward, or repeat the current stage again.
  - Output
    - Perform the above task, return when done

## Egress task

*Note: according to [voiceflow](#) the egress task has multiple substeps, no variables need to be passed in for each subtask.*

```

{
  "type": "function",
  "function": {
    "name": "run_ingress",
    "description": "Conduct the egress task given stage number or move forward
instructions",

```

```

    "parameters": {
      "type": "object",
      "properties": {
        "subtask_redirect": {
          "type": "string",
          "description": "The action of moving to a given stage, the previous stage,
repeating the current stage, or moving to the next stage in the ingress task",
          "enum": ["1a", "1b", "1c", "previous", "next", "repeat", "start"]},
      }
    },
    "required": [subtask_redirect],
  },
}

```

*Notes: If this function is called with no specification on the stage → start from beginning. If a stage number is provided such as 1c, then start at that stage. If the word “previous”, “next”, “repeat” are said, then move backward, forward, or repeat the current stage again.*

HMD:

- Run\_egress\_task(subtask\_redirect)
  - Input:
    - subtask\_redirect → one of ["1a", "1b", "1c", "previous", "next", "repeat", "start"]
  - If this function is called with no specification on the stage → start from beginning
  - If a stage number is provided such as 1c, then start at that stage.
  - If the word “previous”, “next”, “repeat” are said, then retrieve the current stage number (stored as a variable), move backward, forward, or repeat the current stage again.
  - Output
    - Perform the above task, return when done

## Navigation

```

{
  "type": "function",
  "function": {
    "name": "open_map_perform_navigation_tasks",
    "description": "Open map and perform navigation tasks",
    "parameters": {
      "type": "object",
      "properties": {
        "navigation_task": {

```

```

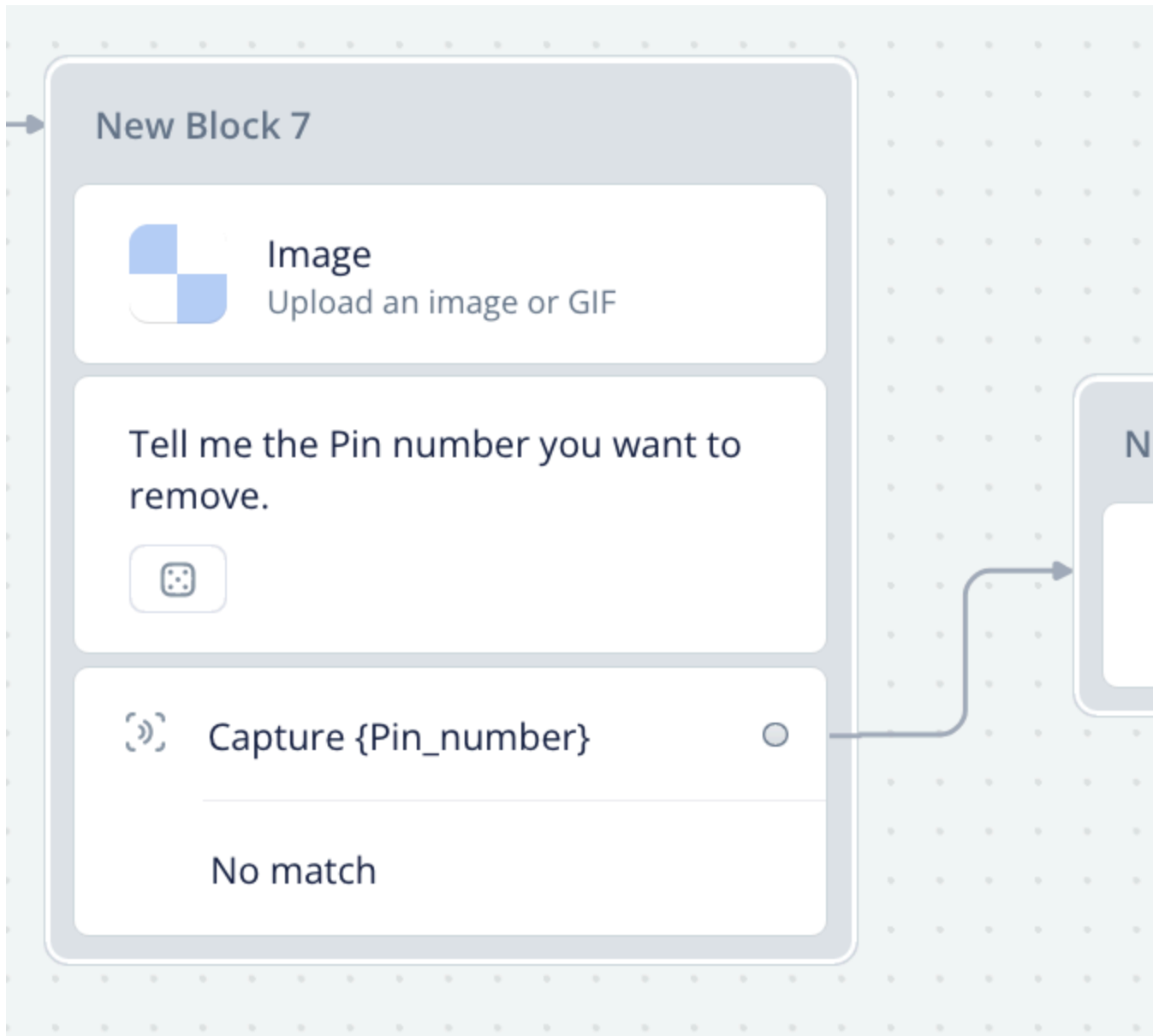
        "type": "string",
        "description": "Choose one of the following tasks: remove pin, pin my location,
and return to airlock. If none of the above are chosen, then only open map",
        "enum": ["remove pin", "pin my location", "return to airlock", "open map"]
    },
    "Pin_number": {
        "type": "integer",
        "description": "number of the pin to remove",
    },
},
"required": ["navigation_task"],
},
}

```

#### HMD functions

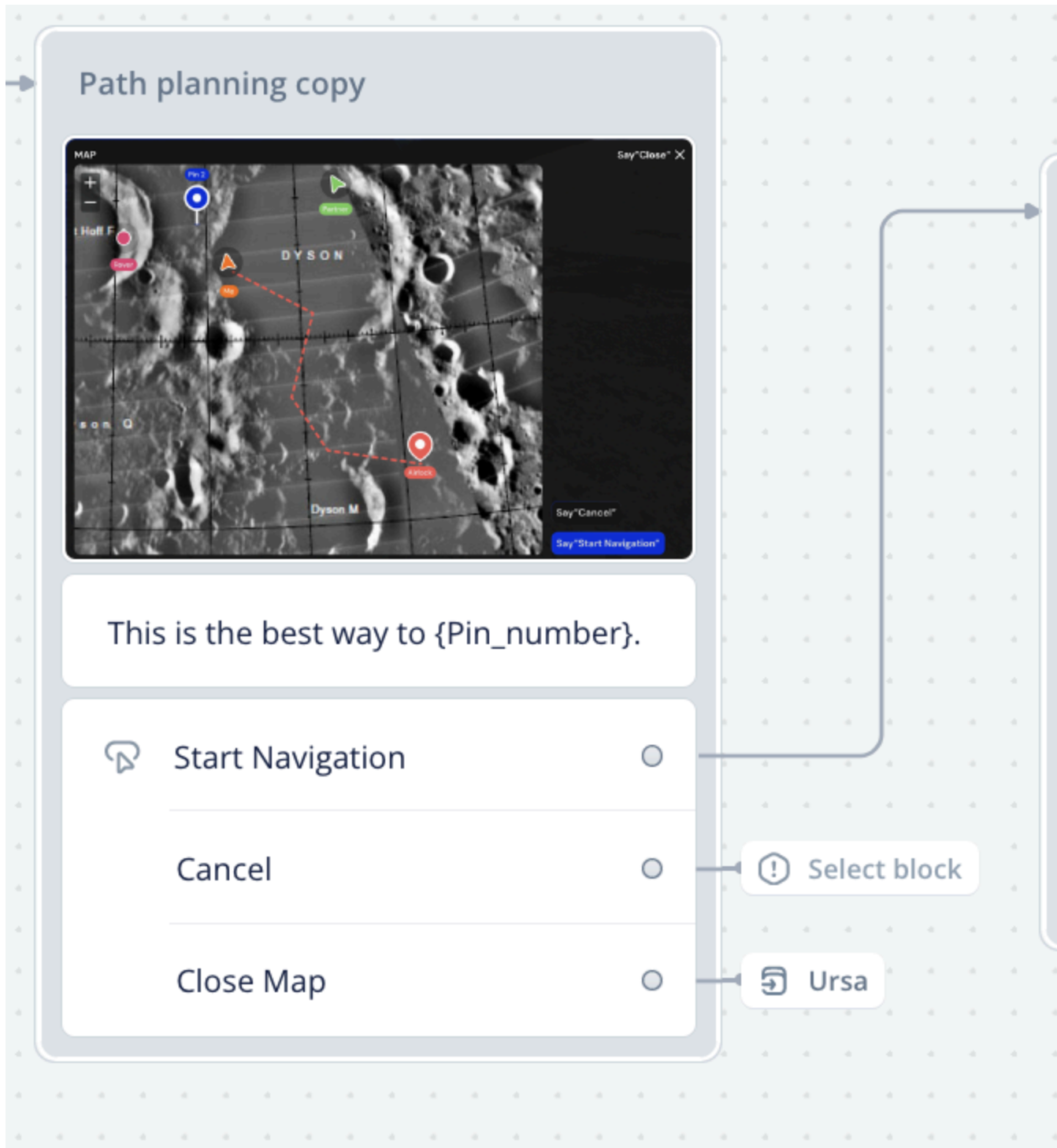
- open\_map()
  - Open the map and show it on the screen
- remove\_pin(pin\_number)
  - Input: pin\_number, integer, the pin to be removed
  - Remove the pin from the map, update the map and the database
  - Return when done
- pin\_my\_location()
  - Retrieve the current location (probably from the database?)
  - Add the pin to the map, update the map
  - Return when done
- return\_to\_airlock()
  - Retrieve the previous pin points
  - Generate a path based on the previous pin points assuming that is the best way to return
  - Show the path on the map
  - Return when done

UI Team:



- This is the current flow, but would it make more sense if the astronaut just specifies the pin number together with the remove pin command?





- Please confirm that this is needed for astronaut navigation. If not, please delete it from the voice flow

## Equipment Diagnosis

(work in progress...)

```

{
  "type": "function",
  "function": {
    "name": "start_equipment_diagnosis",
    "description": "Start equipment diagnosis process",
    "parameters": {
      "type": "object",
      "properties": {},
      "required": [],
    },
  },
}

```

*The function above will ask the astronaut to move to the work site, once arrive, the astronaut should say "I have arrived"*

```

{
  "type": "function",
  "function": {
    "name": "equipment_diagnosis_with_MCC",
    "description": "Based on the given issue, connect to MCC and walk the astronaut through the repair process",
    "parameters": {
      "type": "object",
      "properties": {
        "issue": {
          "type": "string",
          "description": "The current issue the equipment is experiencing, e.g. the cable is not functioning",
        },
        "required": ["issue"],
      },
    },
  },
}

```

## Geologic Sampling

(work in progress...)

## Rover

(work in progress...)