

# CISC 1003 - EXPLORING ROBOTICS

---



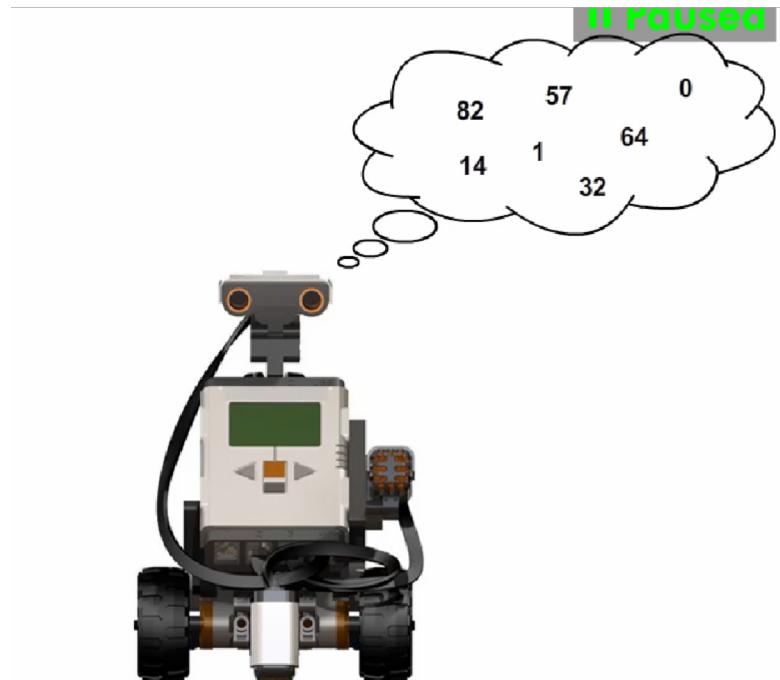
# ROBOT DECISION MAKING

---

CISC1003

# How a Robot Thinks

- Robot can 'learn' the world using their sensors
- Sensors return data in number format



# How a Robot Thinks

- The robot can answer 'yes' or 'no' questions
- Example:
  - Is the touch sensor bumped?
  - Is the audio level in the room above 50%?
- This ability is based on a special logic
  - Called '***Boolean logic***'

# How a Robot Thinks

- Programmers can give the robot its decision-making capability
  - By combining the numbers provided by the sensor with robot ability to answer questions
- This requires the following:
  - Robot is programmed to ask questions
  - Act one way if the answer is 'yes'
    - and another if the answer is 'no'

# How a Robot Thinks

- Boolean operators are used when asking the questions, such as:
  - $<$  'less than' ,
  - $>$  'more than',
  - $==$  "equal to"
  - etc.

# How a Robot Thinks

- Example: We want the robot to stop moving before it runs into a wall
  - Use the feedback from an ultrasonic sensor
  - Use 'less than' operator
    - with a certain distance threshold
      - E.g., 10 inch
- This will result in a program that moves the robot until it detects an obstacle
  - Within the distance specified (10 inch)

# How a Robot Thinks

- How does the program work?
- The robot moves forward
- It repeatedly asks the questions:
  - “Am I 10 inch away from anything?”
- If the answer is no, the robot continues moving forward
  - If the answer is ‘yes’, it stops



# Conditional Statements

- The parts of the program where the robot choose an action
  - Depending on a certain condition

# Summary

- We can create conditional statements
  - by combining sensor output and Boolean operators
- This allows the robot to make decisions

# How a robot thinks

- What kinds of questions can a robot ask?
  - “yes” or “no” questions
  - Questions that have only two possible answers
- What can a robot do with the answer to the question?
  - Use the answers to choose between two different actions
    - E.g., move forward or stop



# RSVP: ROBOT SCENARIO VISUAL PLANNING

---

CISC1003

Exploring Robotics

# RSVP: ROBOT SCENARIO VISUAL PLANNING

- Making a picture or a “visual representation” of the scenario and instructions you want the robot to perform
  - can be great way to ensure your robot performs the tasks properly
- A picture of the instructions the robot will perform allows you to think through the steps before translating them to the code

# RSVP: ROBOT SCENARIO VISUAL PLANNING

- The RSVP is composed of three types of visuals:
  - A floorplan of the physical environment of the scenario
  - A statechart of the robot and object's states
  - Flowcharts of the instructions for the tasks

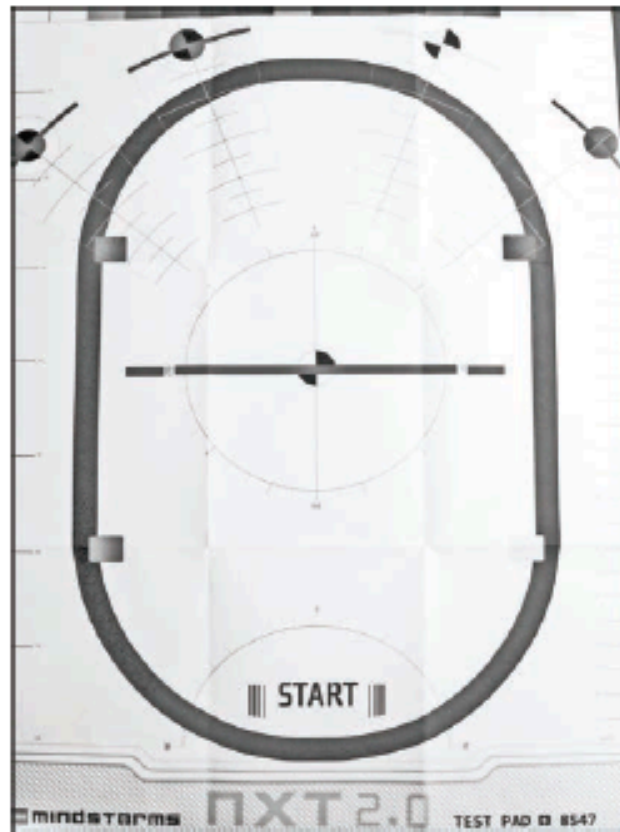
# Mapping the Scenario

- The first part of the RSVP is a map of the scenario
- A map is a symbolic representation of the environment
  - where the tasks and situations will take place
  - The environment for the scenario is the world in which the robots operate



# Mapping Example

A robot world for NXT Mindstorms Test Pad



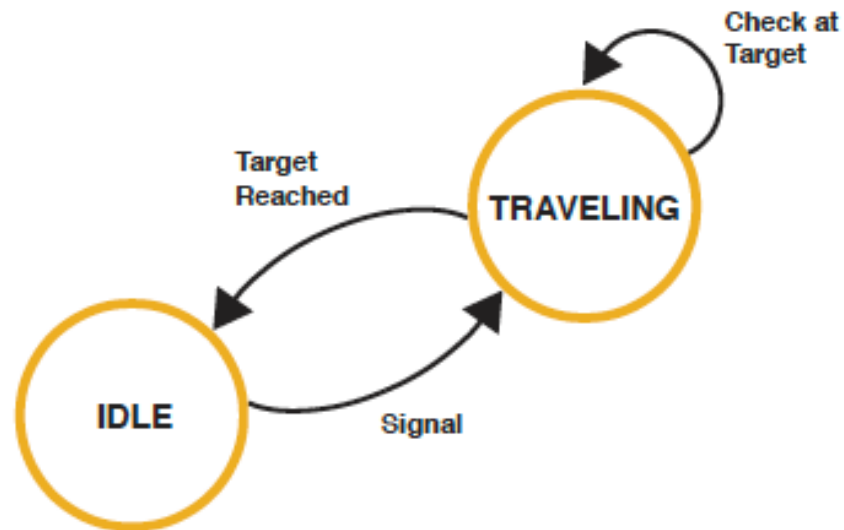
# State Chart

- A statechart is one way to visualize a state machine.
- For example , a “change of state” can be as simple as a change of location.
  - When the robot travels from its initial location to the location next to the table, this is a change of the robot’s state.
  - Another example is that the birthday candles change from an unlit state to a lit state.

# State Chart

- The state machine captures the events, transformations, and responses.
- A statechart is a diagram of these activities.
- The statechart is used to capture the possible situations for that object in that scenario

# Example – State Machine



# Pseudocode and Flowcharting

- Flowcharts are a type of statechart
  - they contain states that are converted to actions and activities
    - Things like decisions and repetitions are easily represented
      - what happens as the result of a branch can be simply depicted.
  - Some suggest flowcharting before writing pseudocode

# Pseudocode and Flowcharting

- Pseudocode has the advantage of being easily converted to a programming language or utilized for documenting a program
  - It can also be easily changed.
- A flowchart requires a bit more work to change when using flowcharting software.

# PseudoCode

- What is the problem we are trying to solve?
  - Identify the behavior you need
  - Write down the sequence of behaviors that is needed
    - To achieve your goals
  - Identify the sub-tasks needed to achieve your goals

# Pseudocode

- As we increase the level of details, we will reach commands we can express directly in programming language
- This is the plan the robot needs to follow
- The steps are written in English
  - So can be understood by the human programmer
- This is called *Pseudocode*



# Pseudocode and Flowcharting

RSVP Type	Advantages	Disadvantages
<p><b>Pseudocode:</b></p> <p>A method of describing computer instructions using a combination of natural language or programming language.</p>	<p>Easily created and modified in any word processor.</p> <p>Implementation is useful in any design.</p> <p>Written and understood easily.</p> <p>Easily converted to a programming language.</p>	<p>Is not visual.</p> <p>No standardized style or format.</p> <p>More difficult to follow the logic.</p>
<p><b>Flowcharting:</b></p> <p>Flow from the top to the bottom of a page. Each command is placed in a box of the appropriate shape, and arrows are used to direct program flow.</p>	<p>Is visual, easier to communicate to others.</p> <p>Problems can be analyzed more effectively.</p>	<p>Can become complex and clumsy for complicated logic.</p> <p>Alterations may require redrawing completely.</p>

# Flowcharting

- The four common symbols used in flowcharting are:
  - Start and Stop
  - Input and output
  - Decisions
  - Process

# Flowcharting

- Start and stop:
  - The start symbol represents the beginning of the flowchart with the label “start” appearing inside the symbol.
  - The stop symbol represents the end of the flowchart with the label “stop” appearing inside the symbol. These are the only symbols with keyword labels.
- Input and output:
  - The input and output symbol contains data that is used for input (e.g., provided by the user)
    - and data that is the result of processing (output)

# Flowcharting

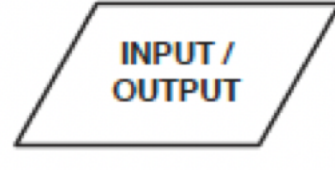
- Start and stop:



- The start symbol represents the beginning of the flowchart
  - with the label “start” appearing inside the symbol.
- The stop symbol represents the end of the flowchart
  - with the label “stop” appearing inside the symbol.
- These are the only symbols with keyword labels.

# Flowcharting

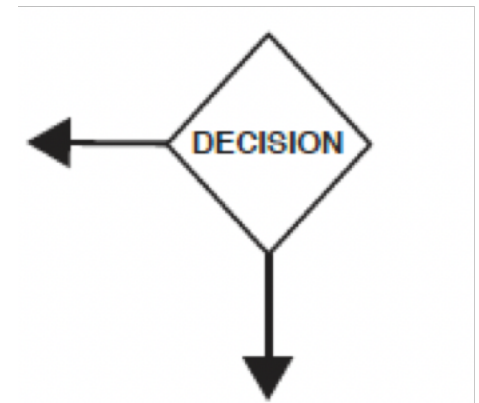
- Input and output:



- The input and output symbol contains data that is used for input (e.g., provided by the user)
  - and data that is the result of processing (output)

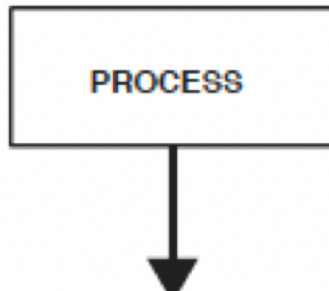
- Decisions:

- The decision symbol contains a question or a decision that has to be made.

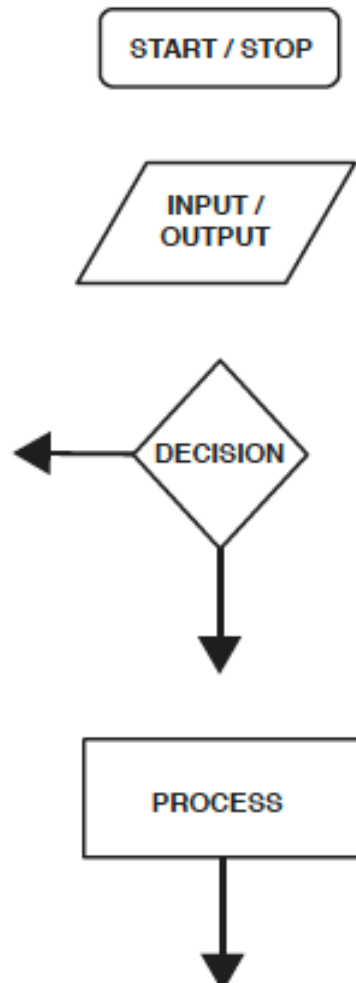


# Flowcharting

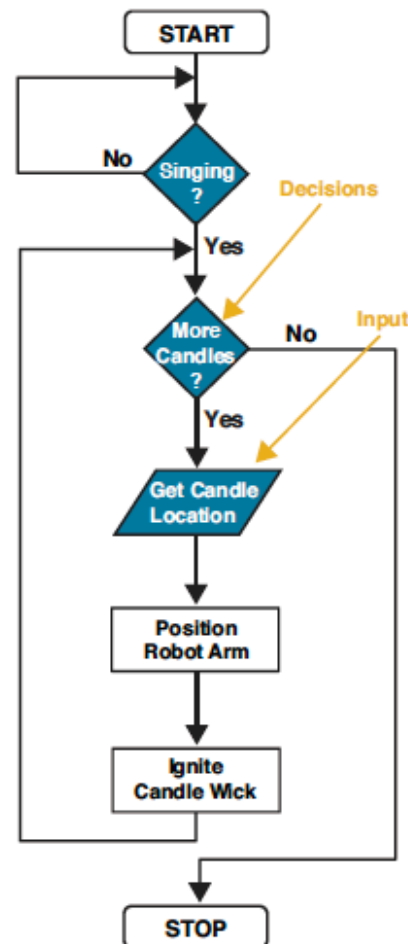
- Process:
  - The process symbol contains brief descriptions (a few words) of a rule or some action taking place .



# Common Flowchart Symbols



# Example - Candlelighting Flowchart





# Flowcharting

- The task a robot executes can be a series of steps performed one after another
  - a sequential flow of control.
- ***Flow of control*** details the direction the process takes
  - which way program control “flows
- Flow of control determines how a computer responds
  - when given certain conditions and parameters

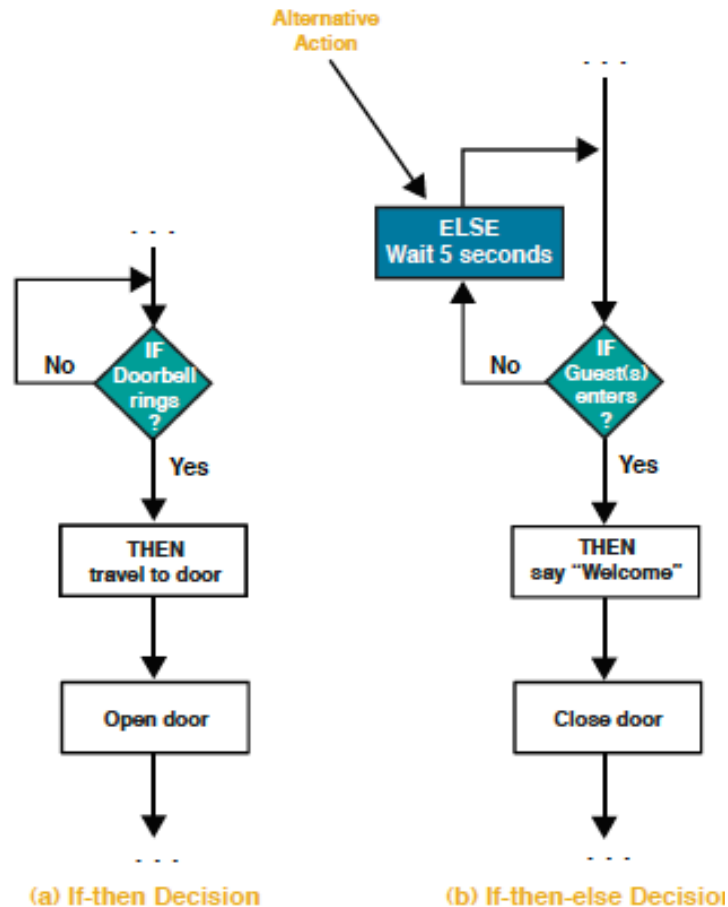
# Example: Sequential Flowchart



# Flowcharting

- A decision symbol is used to construct branching for alternative flow controls.
- Decision symbols can be used to express decision, repetition, and case statements
- A simple decision is structured as an if-then or if-then-else statement

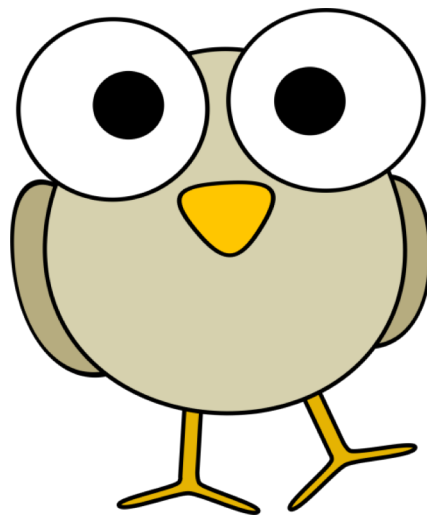
# Example – Guest Welcoming Flowchart



# Summary

- The RSVP is composed of three types of visuals:
  - A floorplan of the physical environment of the scenario
  - A statechart of the robot and object's states
  - Flowcharts of the instructions for the tasks
- These visuals ensure that you have a “clear picture” of what has to be done
  - to program a robot to save the world
    - or light the candles on a cake

- Questions?



??

# Lab time!

- Let's work with our robots!

