CISC1003 – Unit C
Locomotion

http://artimusmarching.com/locomotion/

Topics

- Modes of Locomotion
- Algorithm
- Multitasking

Locomotion

- Locomotion = locus (place) + motion
- Locomotion refers to the way a body moves
  - from place to place.
- A fundamental function of humans, animals
  - Acquired through training
  - Requiring significant “brain power”
- It’s generally the first challenge for a robot
- Many modes of locomotion exist

Modes of Locomotion

- Legs:
  - Walking, crawling, climbing, jumping, hopping etc.
- Wheels:
  - Rolling
- Arms:
  - Swinging, crawling, climbing, lifting
- Wings:
  - Flying
- Flippers:
  - Swimming

Modes of Locomotion

- Most common, legged vs. Wheeled
- Benefits and challenges:
  - Wheeled:
    - Most efficient use of power, low DOFs.
  - Legged:
    - Large DOFs, challenge of stability.

Two Kinds of Stability

- **Static stability**: robots maintain upright without constant active control
  - Are humans statically stable?
    - We as humans are not statically stable!
      - Fall if fainting, etc.

http://nursinglink.monster.com/benefits/articles/5019-top-10-qualities-of-a-great-nurse?page=3

Two Kinds of Stability

- **Static stability**: robots maintain upright without constant active control
  - – **support polygon** is a horizontal region over which the center of mass must lie to achieve static stability
  - – Maintained when center of gravity (COG) is above a certain horizontal region
    - Region called **support polygon**
  - – Statically stable walking is slow, energy inefficient

Two Kinds of Stability

- **Dynamic stability**: robots must actively balance or move to maintain stability
  - The *inverse pendulum* model for one legged balance
  - Two legged walking alternates between swing and stance phase
    - between the two legs.

Two Kinds of Stability

- A statically stable robot can use dynamically stable walking to better use energy – tradeoff between stability/speed.

Gaits

- The way a robot moves by using a particular pattern of footfall
  - 2 legged: alternating swing and stance phases.
  - 4 legged: lateral walking vs. diagonal walking
    - Lateral walking: right hind, right front, left hind, left front
    - Diagonal: the feet on opposite sides move forward in sequence

http://www.physiocapsule.com/2016/10/concept-of-gait-cycle-explained.html

Gaits

- The way a robot moves by using a particular pattern of footfall
  - 2 legged: alternating swing and stance phases.
  - 4 legged: lateral walking vs. diagonal walking
  - 6 legged: alternating tripod gait vs. ripple gait.
    - Tripod gait: weight shifts to three legs each time
      - https://www.youtube.com/watch?v=nRtJu4qrqn0
    - Ripple gait: two legs used each time
      - One leg changes each time
      - https://www.youtube.com/watch?v=3_Qk5svpUc0

Gaits

- Consideration for desirable robot gaits
  - Stability, speed, energy
  - Robustness, simplicity

Wheels and Steering

- Wheels are the choice of locomotion in robotics
  - Advantages of wheels:
    - Highly efficient
    - Simple to control
- Most wheeled robots are not holonomic
  - Following a specific trajectory (motion planning) is more difficult than simply moving from one place to another (navigation).
- Differential drive(steering):
  - Wheels are driven independently by separate motors => easier control.

https://www.tirebuyer.com/wheels

Go Beyond Locomotion - Dancing Automaton

- One or more robots come together
  - With music, dressed in costume
  - Moving in creative harmony.
- Need to develop an **algorithm**.
- Robot will be **multitasking**.

https://www.youtube.com/watch?v=Fg0AGH_TaiQ

Algorithm

- A step-by-step sequence of instructions for carrying out some task.
- Examples of algorithms outside of computing:
  - Cooking recipes
  - Dance steps
  - Proofs (mathematical or logical)
  - Solutions to mathematical problems
- Often, there is more than one way to solve a problem.

https://www.computerhope.com/jargon/a/algorith.htm

Algorithms -Solving problems

- In computing, algorithms are synonymous with problem solving.
- *How To Solve It* [George Polya, 1945]
  - Understand the problem
  - Devise a plan
  - Carry out your plan
  - Examine the solution

Algorithms –Polya[1945]

- Understand the problem:
  - Understand all the words, goal
  - Create a picture or a diagram to help solve
  - Is there enough information to solve the problem?
- Devise a plan
  - Choose a strategy: guess and check, eliminate possibilities, etc.
- Carry out your plan
  - Write the program, run the system
- Examine the solution
  - Look back, did you solve the problem?

Algorithms - features

- Speed (number of steps)
- Memory (size of work space)
- Complexity (can others understand it?)
- Parallelism (can you do more than one step at once?)

Case Study –
Boids Algorithm by Craig Reynolds

Algorithm - *Boids* by Craig Reynolds

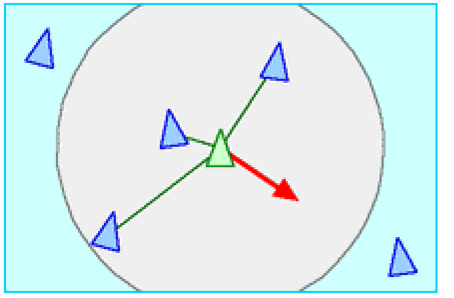
- Algorithmic for coordinated animal motion
  - Models steering behaviors
    - for animated flocking creatures.
  - Allowed individual elements to navigate their digital environments in a “life-like” manner
    - with strategies for different actions:
      - seeking, fleeing, wandering, arriving, pursuing, evading, path following, obstacle avoiding, etc.

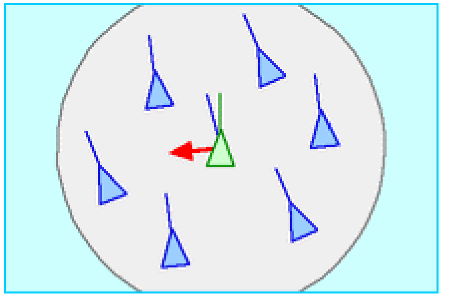
Algorithm - *Boids* by Craig Reynolds (cont.)

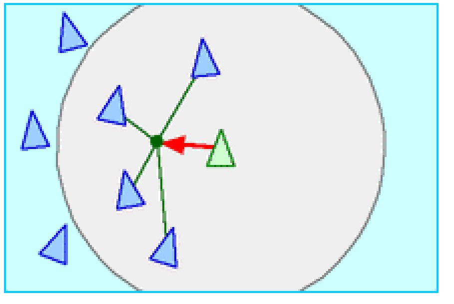
- System has multiple characters
  - each steering according to simple locally-based rules,
- Surprising levels of complexity emerge
  - the most famous example being Reynolds' "boids" model for "flocking"/"swarming" behavior.

Algorithm - *Boids* by Craig Reynolds (cont.)

- Simple steering behaviors:
  - Separation:
    - avoid crowding neighbors
  - Alignment:
    - steer towards average heading of neighbors
  - Cohesion:
    - steer towards average position of neighbors







https://www.red3d.com/cwr/boids/

Algorithm - *Boids* by Craig Reynolds (cont.)

- An animated short featuring the boids model called **Stanley and Stella in: Breaking the Ice** was created
  - Boids video

Multitasking

- **Computer Multitasking**: When multiple tasks, also known as processes, are executed concurrently
  - May share common processing resources such as a CPU.
- If only one CPU exists, only one task runs at any point in time
  - the CPU is actively executing instructions for that task.
- Multitasking involves scheduling which task may be the one running at any given time
  - And when another waiting task gets a turn.

https://talkroute.com/multitasking-vs-single-tasking-which-is-more-effective/

Multitasking

- Each program can have multiple tasks, from which one is the main task.
- The execution of the program jumps from one active task to another.
- The act of reassigning a CPU from one task to another one is called a **context switch**
  - When context switches occur frequently enough the illusion of parallelism is achieved.


QUESTIONS

Introduction to Programming

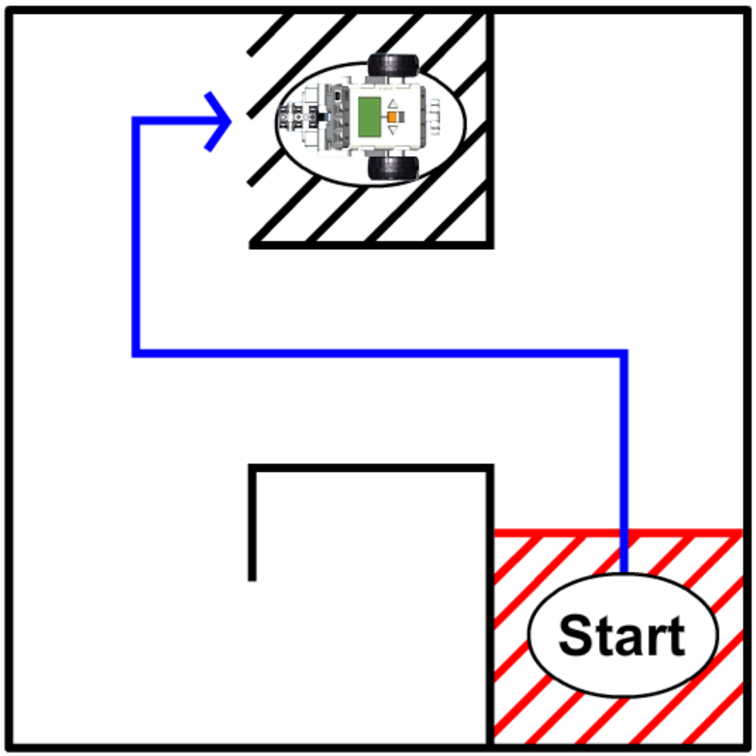
Planning and Behavior, Pseudocode

Planning and Behavior

- What is the problem?
  - Identify the behavior you need

Planning and behavior

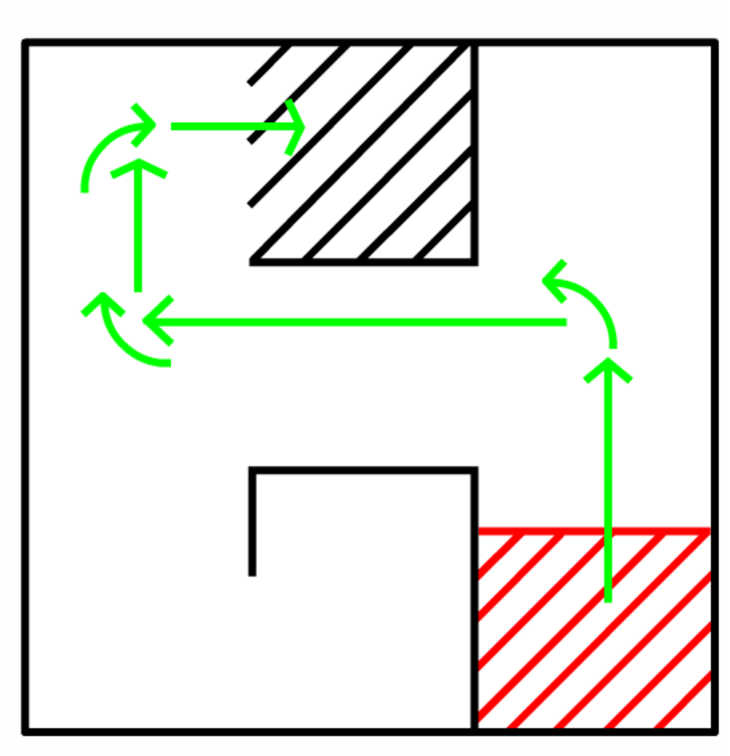
- Example: follow the path



http://cmra.rec.ri.cmu.edu/previews/nxt_products/nxt_video_trainer2/

Planning and behavior

- Break the main path into smaller paths:



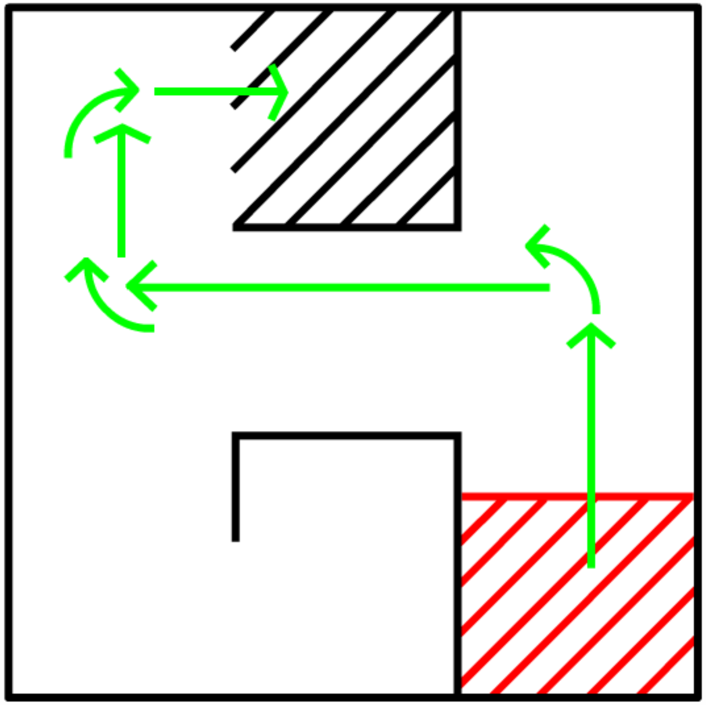
http://cmra.rec.ri.cmu.edu/previews/nxt_products/nxt_video_trainer2/

Planning and behavior

- Each of the smaller paths is called a behavior
- Write down the sequence of behaviors that is needed

Planning and behavior

- Follow the path:
  - Move forward
  - Turn left
  - Move forward
  - Turn right
  - Move forward
  - Turn right
  - Move forward



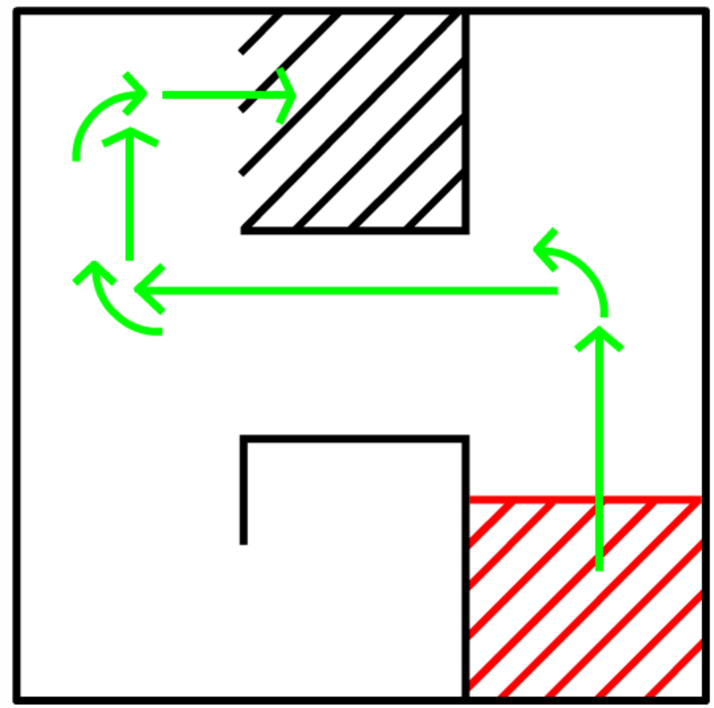
http://cmra.rec.ri.cmu.edu/previews/nxt_products/nxt_video_trainer2/

Planning and behavior

- Can we break these into smaller tasks?

Planning and behavior

- Follow the path:
  - Move forward
    - Left motor forward
    - Right motor forward
    - Wait 2 seconds
  - Turn left
    - Left motor reverse
    - Right motor forward
    - Wait 1 second
- Etc…



http://cmra.rec.ri.cmu.edu/previews/nxt_products/nxt_video_trainer2/

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*


QUESTIONS