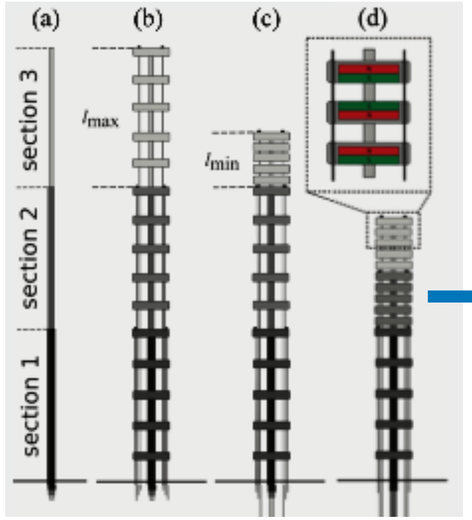
METABot Learning to Play

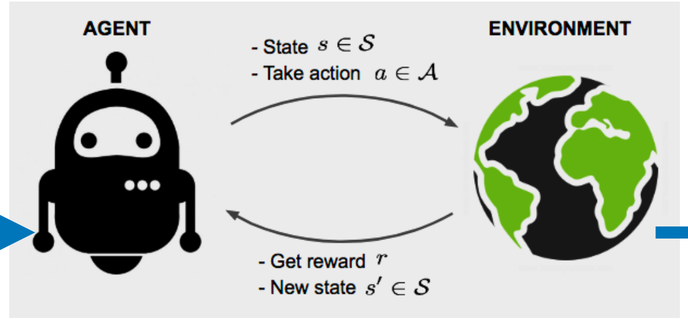
Current Progress, July 15 2020

Student: Abdulwasay Mehar
Supervision: Jessica Burgner-Kahrs & Reinhard Grassmann

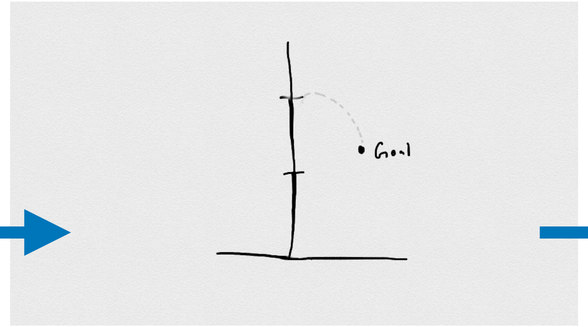
Project Recap



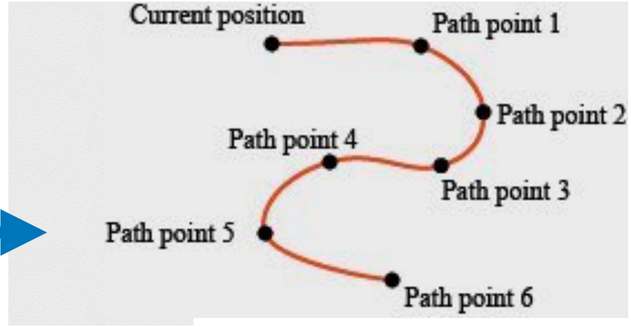
MetaBot



Reinforcement Learning



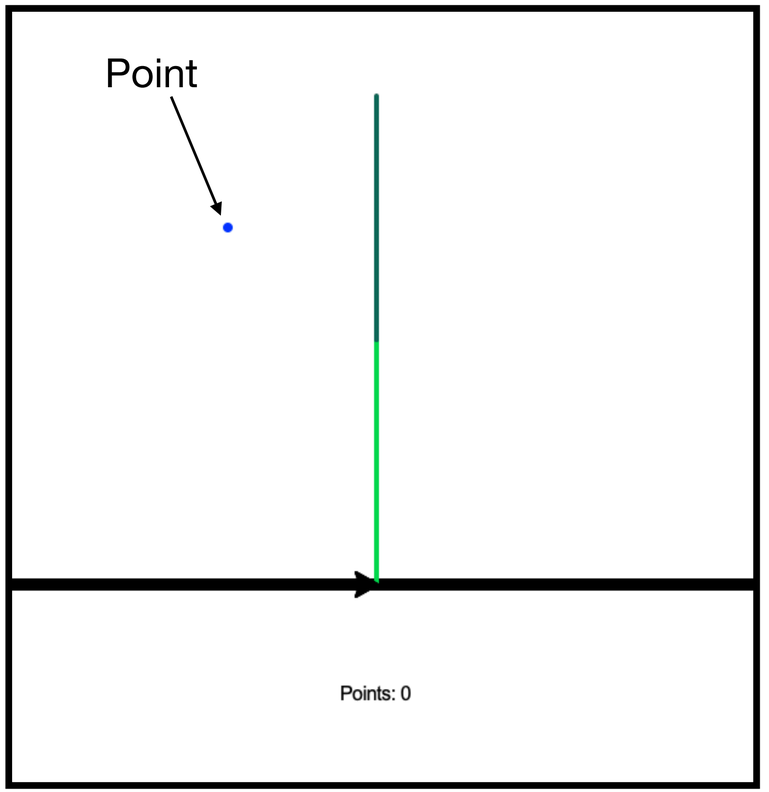
Reaching Goal Pos



Follow the Leader

Current Progress

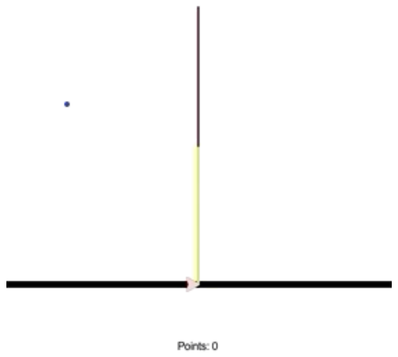
- Environment simulation is complete
- Features
  - Point Based Game
  - Constant Curvature Assumed
  - Modular
  - 2D Plane
  - Render/Non-Render



Visual Representation

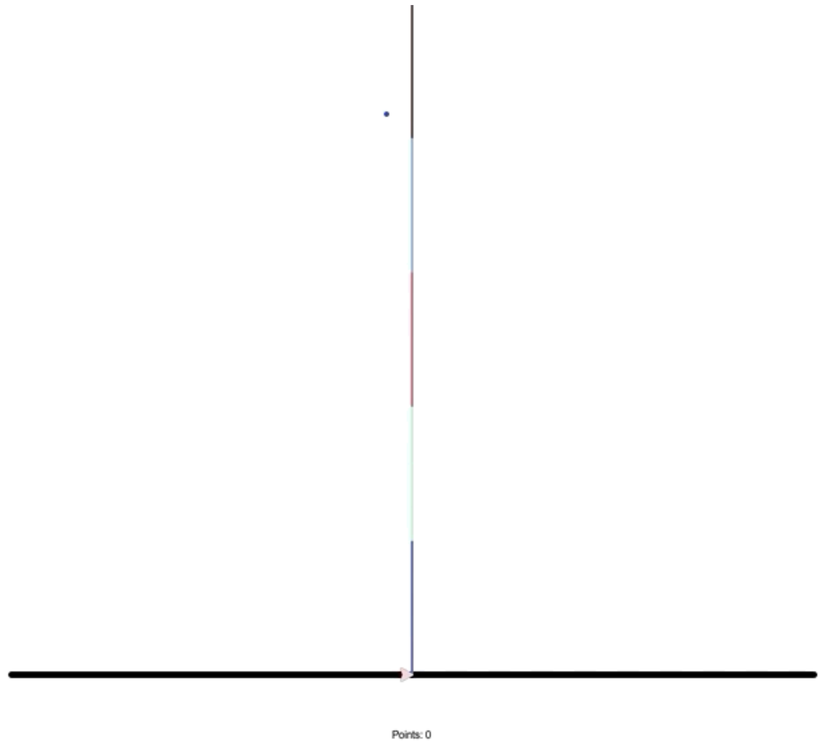
Current Progress

- Modularity: robot.newSection( )



2 Sections

3 Sections



6 Sections

Current Progress

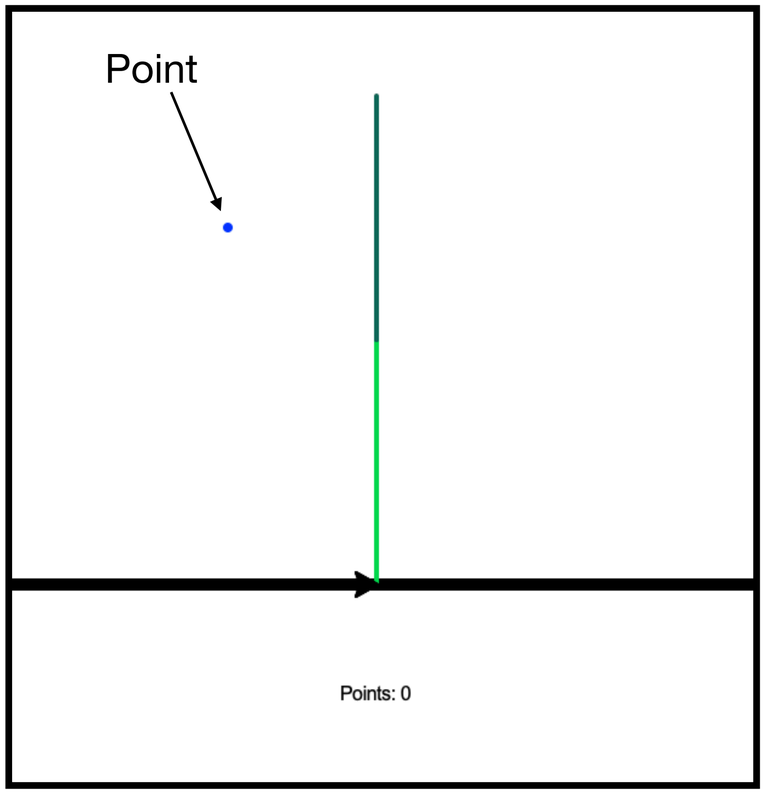
- Environment Observations
  - Current State: (Sections Configs, Distance to Point)
  - Action: (SectionNumber, Action)
  - Next State: (Sections Configs, Distance to Point)
  - Reward: Negative distance to point
  - Done: True/False (ie: Reached Goal Point?)

```
Observation(state=[1e-05, 100, 1e-05, 100, 39.539966985548986],
 nextState=[-0.01, 100, 1e-05, 100, 40.55957219441828],
 reward=-40.55957219441828, action=1, done=False)
```

Example returned observation for a 2-Section Robot

Current Progress

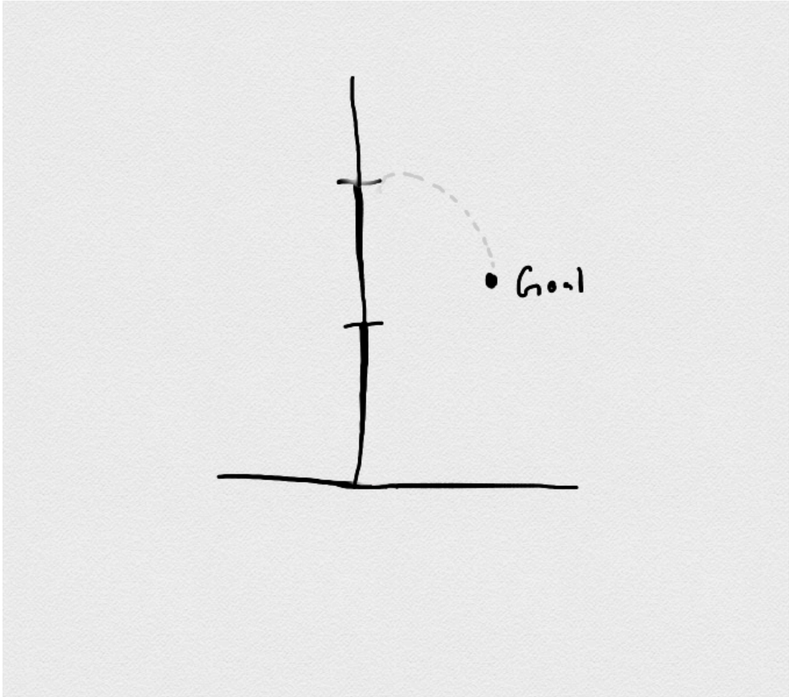
- Environment simulation is complete
- Reinforcement Learning Model in progress
  - Current task goal: Reach goal point



Visual Representation

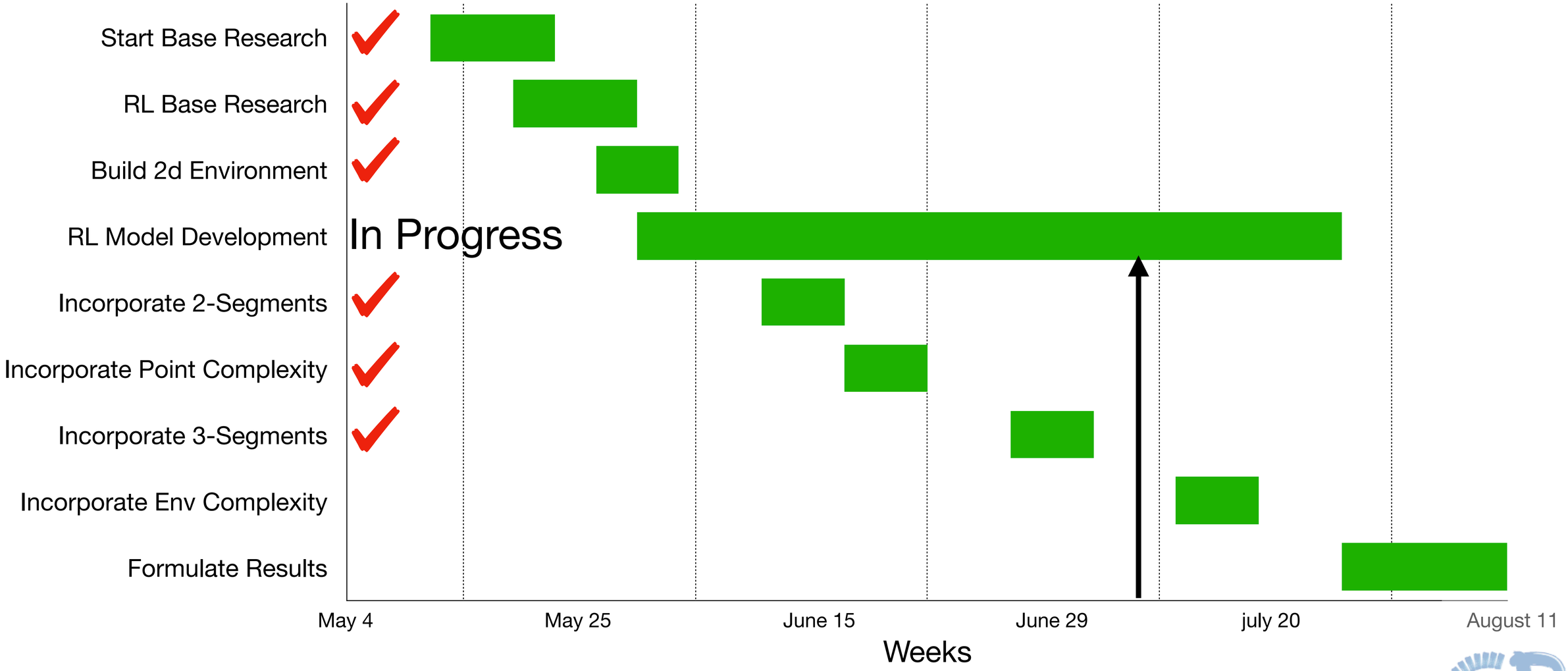
Current Progress

- Environment simulation is complete
- Reinforcement Learning Model in progress
  - Current task goal: Reach goal point
  - Future Incorporation: Follow the leader Policy



Sample End Goal

Timeline


Start Base Research
RL Base Research
Build 2d Environment
RL Model Development
In Progress
Incorporate 2-Segments
Incorporate Point Complexity
Incorporate 3-Segments
Incorporate Env Complexity
Formulate Results
May 4
May 25
June 15
June 29
july 20
August 11
Weeks

Thank You !

Any Questions?

Model

- Base Architecture: Double Deep Q Network
  - Input: Current State
  - Output: Next Possible Action and section
- Dataset: Replay Buffer
- Loss Function

$$l = \left(r + \gamma \max_{a} Q(s', a', \mathbf{w}) - Q(s, a, \mathbf{w}) \right)^2$$

r = Reward γ = Discount Factor s' = Next State s = Current State

Model

- Base Architecture: Double Deep Q Network
  - Input: Current State + Reward
  - Output: Next Possible Action and section
  - 2 Hidden Layers, with Relu Activation Functions
- Dataset: Replay Buffer
- Loss Function

$$I = \left(r + \gamma \max_{a} Q(s', a', \mathbf{w}) - Q(s, a, \mathbf{w}) \right)^2$$

r = Reward γ = Discount Factor s' = Next State s = Current State