

CS 175: Project in Artificial Intelligence Winter 2020

Lecture 2: Platforms

Roy Fox

Department of Computer Science

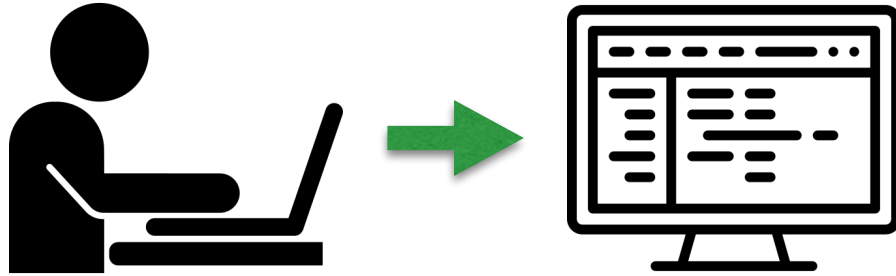
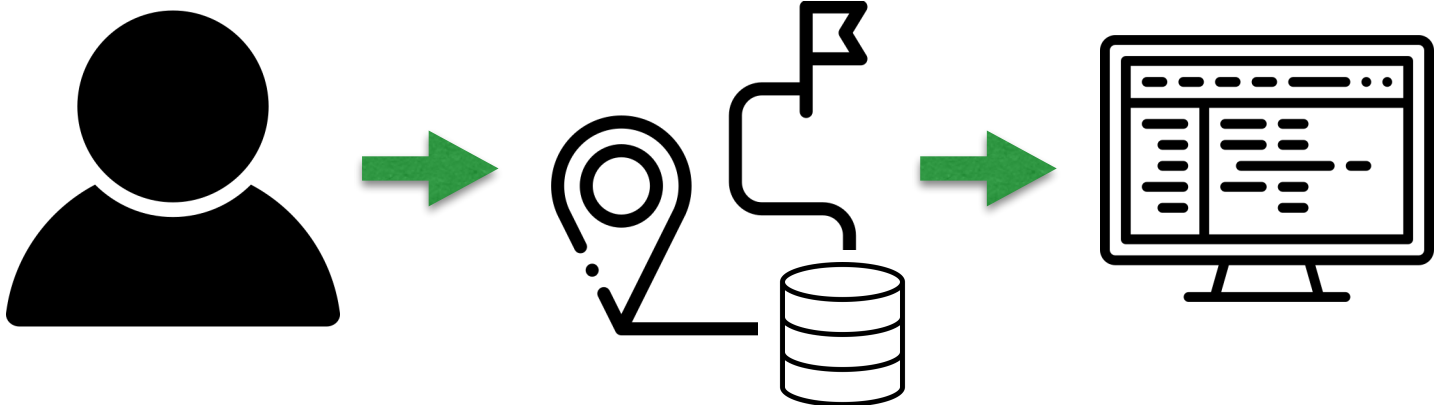
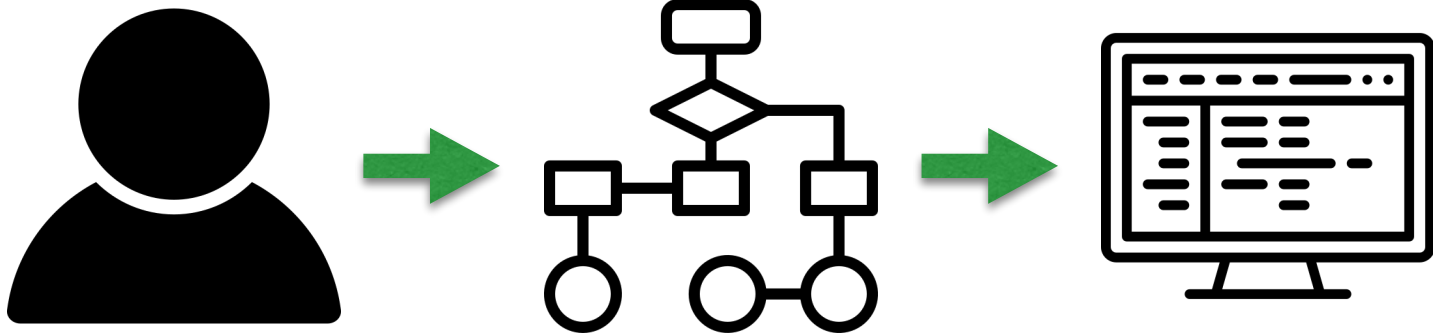
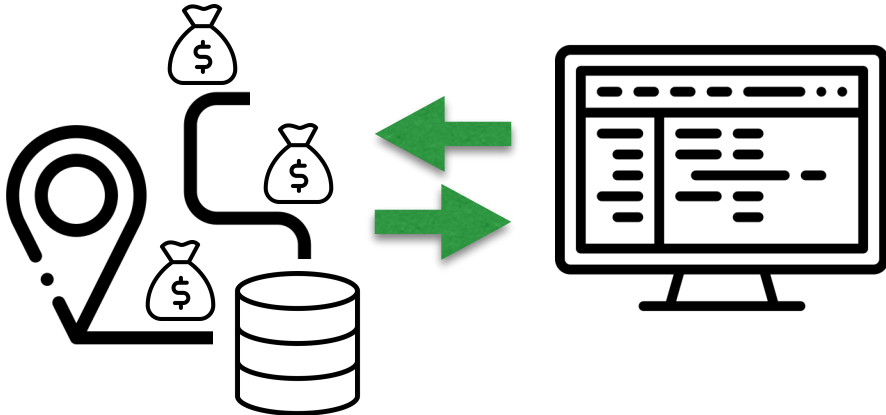
Bren School of Information and Computer Sciences

University of California, Irvine

Today's lecture

- Brief introduction to reinforcement learning
- Minecraft + Malmö + MineRL
- Duckietown + AIDO
- ColosseumRL

Learning policies

| | Explicit | Implicit |
|--------|---|---|
| "how" | <p>Programming</p>  | <p>Imitation Learning</p>  |
| "what" | <p>Specification</p>  | <p>Reinforcement Learning</p>  |

RL is ML... but special

- Test distribution of trajectories depends on the policy!
 - Cannot avoid train–test mismatch
 - To reduce it, learner interacts with the environment to collect data = exploration
 - Balanced exploration is challenging
- Policy space is strewn with local optima
 - Actions in a sequence need to be coordinated
- A good policy may require memory
 - Learning to remember is hard!

RL — the frontier

- How to perform better exploration?
- How to model / structure the agent's policy? in particular, its memory
 - Hierarchical RL
- How to jointly learn multiple tasks?
- How to learn from more kinds of data?
 - RL + imitation learning / NLP / vision / program synthesis
- How to interface with a human teacher?

What makes a good project

- Science: what have we learned?
 - Compare multiple methods
 - Demonstrate a failure mode of a method
 - Explain why the results are what they are
- Technology: how is this useful?
 - Propose a new method or component
 - Contribute an elegant design or implementation
- Art: what is the aesthetic value?
 - Make something cool!
 - Make something inspiring

Learning Goals

Practical AI/ML

- Be creative about ideas
- Understand what's practical
- Implement and debug algorithms
- **State-of-art AI application!**

Software Engineering

- Design a complex software system
- Use modern software practices
- Learn to program collaboratively
- **Small team of developers!**

Presentation Skills

- Be able to “sell” your idea in writing/images/videos
- Present your project in a convincing manner
- Document and maintain a website
- **Public presence of the project!**

Course Project



Groups for the Project

- Team size should be 3
 - Larger teams not allowed
 - Smaller in special cases (meet me)

Use Github

A screenshot of the GitHub website. The top section is dark with the GitHub logo and the text "Built for developers". Below this is a paragraph describing GitHub as a development platform and a green "Sign up for GitHub" button. The bottom section is light and features the "GitHub for teams" logo and the text "A better way to work together", followed by a paragraph about team collaboration and a blue "Sign up your team" button.

Built for developers

GitHub is a development platform inspired by the way you work. From **open source** to **business**, you can host and review code, manage projects, and build software alongside millions of other developers.

Sign up for GitHub

GitHub for teams

A better way to work together

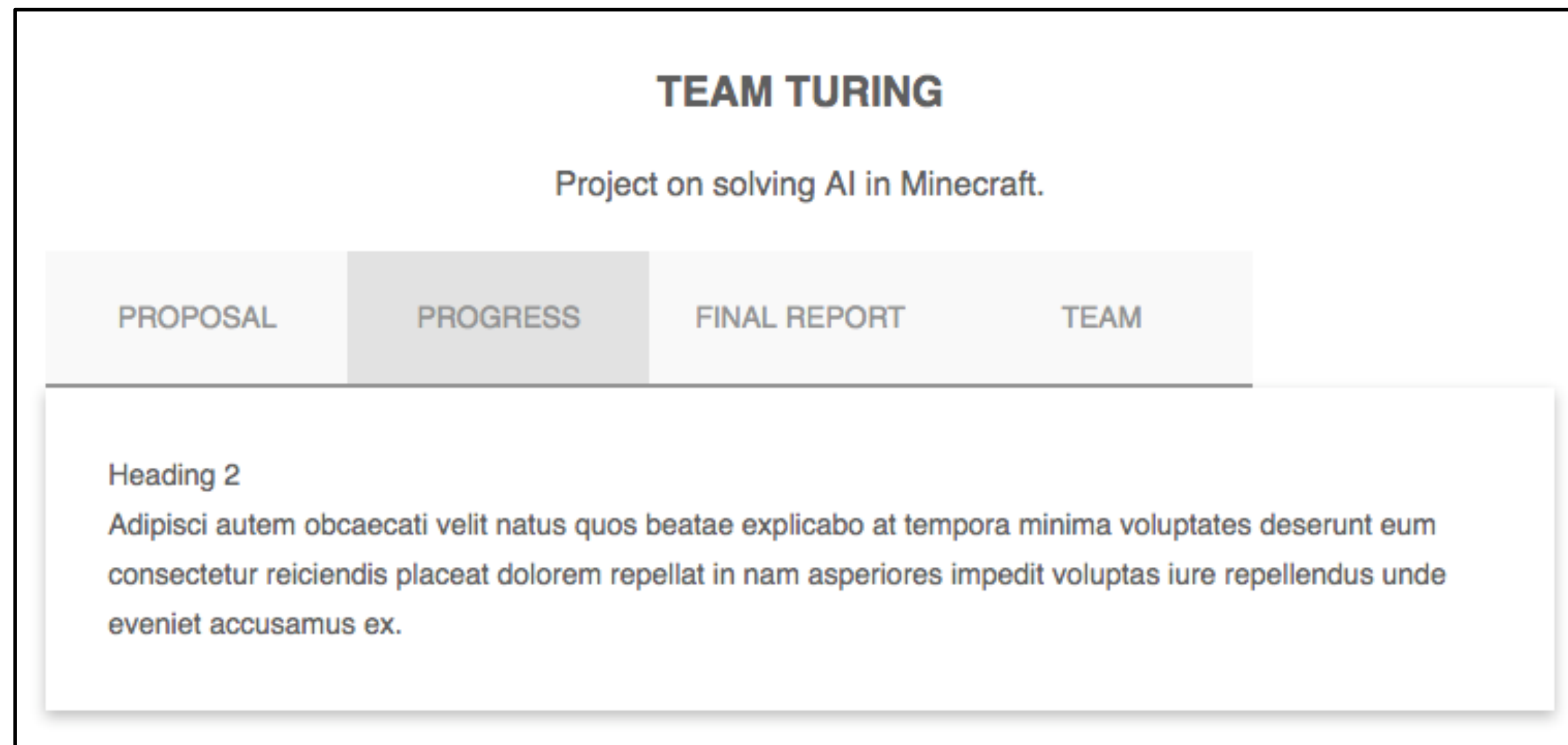
GitHub brings teams together to work through problems, move ideas forward, and learn from each other along the way.

[See how teams work together on GitHub](#)

Sign up your team

Project “Submissions”

Maintain a website with a page for every submission



Easiest option: Github Pages

What is

Original released in 2011, bought by Microsoft in 2014

Second best-selling video game ever, after Tetris

Available on almost every device possible!

“Parent” Speak: Like Lego, but in computers!

Video



<https://www.youtube.com/watch?v=LsDRgz6xZr0>

Gameplay Components

Navigation

Virtually infinite environments!
First person shooter controls
Walking, turning, strafing, turning head, ...
Jumping, climbing, crouching, falling, ...

Gathering

World is made up of blocks/cubes that you can pick up, and store
Many different types of blocks (~100)

Crafting

Blocks can be combined
Recipes describe how to combine
Build: Blocks, Tools, Weapons, Vehicles, ...

Building

Put blocks on top of each other
Different blocks interact differently
Can create all kinds of structures

Combat

Basic combat with one-button control
A few different types of enemies

Recipes and Crafting



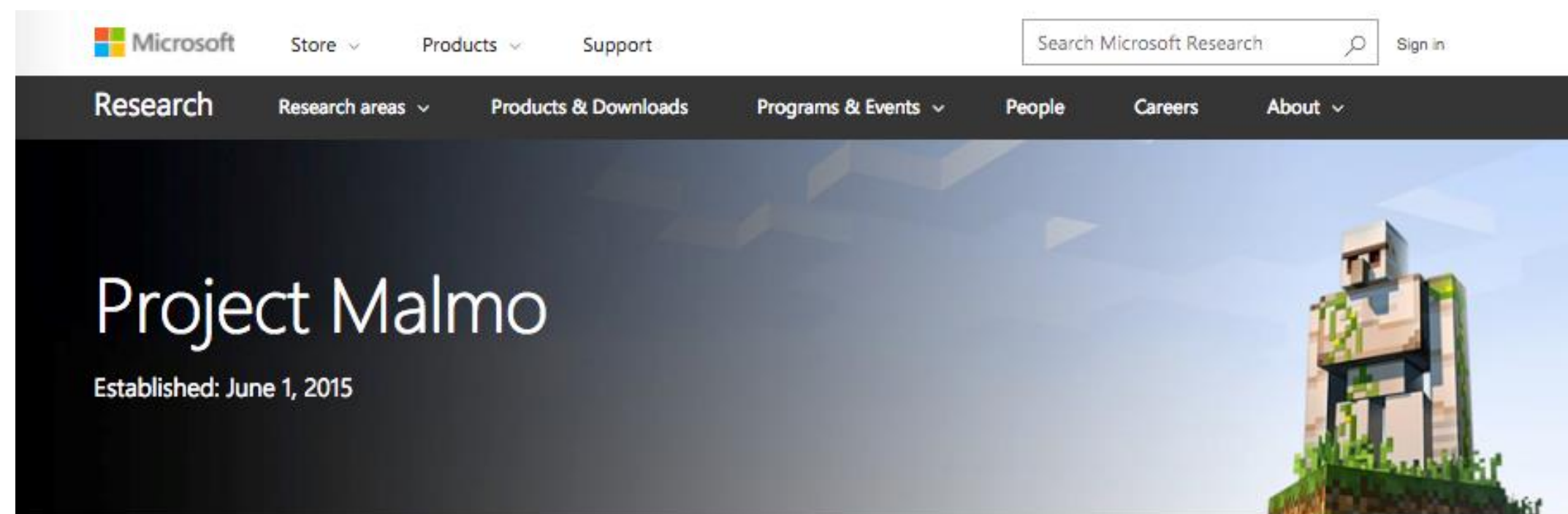
Impressive Examples



<https://www.youtube.com/watch?v=Z0sq9SR4kgI>

<https://www.youtube.com/watch?v=8o2vRGiJ0ms>

Introduction to Malmo



https://www.youtube.com/watch?v=KkVj_ddseO8



Matthew Johnson
Principal RSDE
Lead, Agile
Projects Team



Pushmeet Kohli
Principal
Researcher



Robert Schapire
Principal
Researcher



Katja Hofmann
Researcher



Jamie Shotton
Partner Scientist
Lead



Bhaskar Mitra
Senior Applied
Scientist

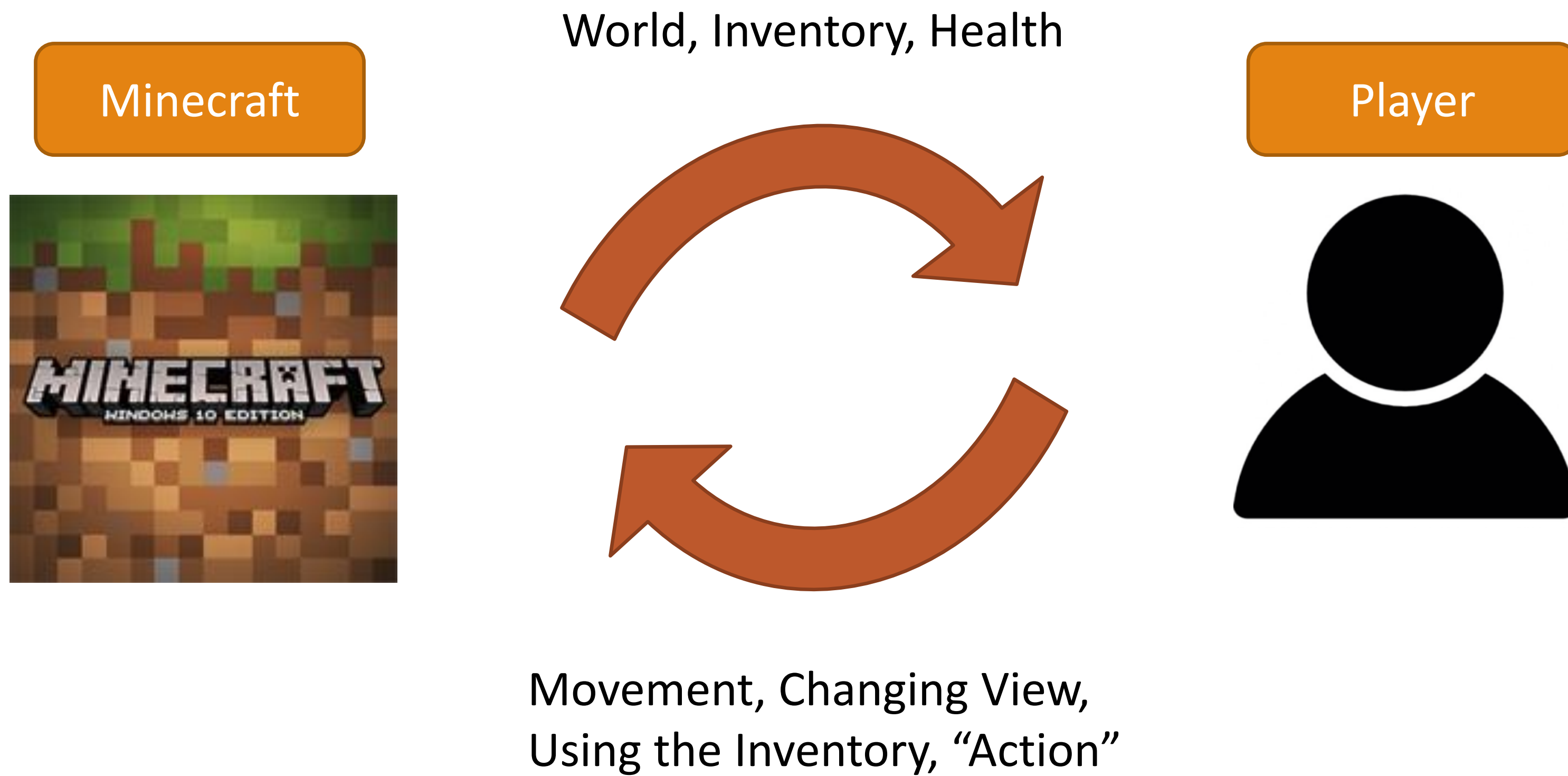


Evelyne Viegas
Director of AI
Outreach

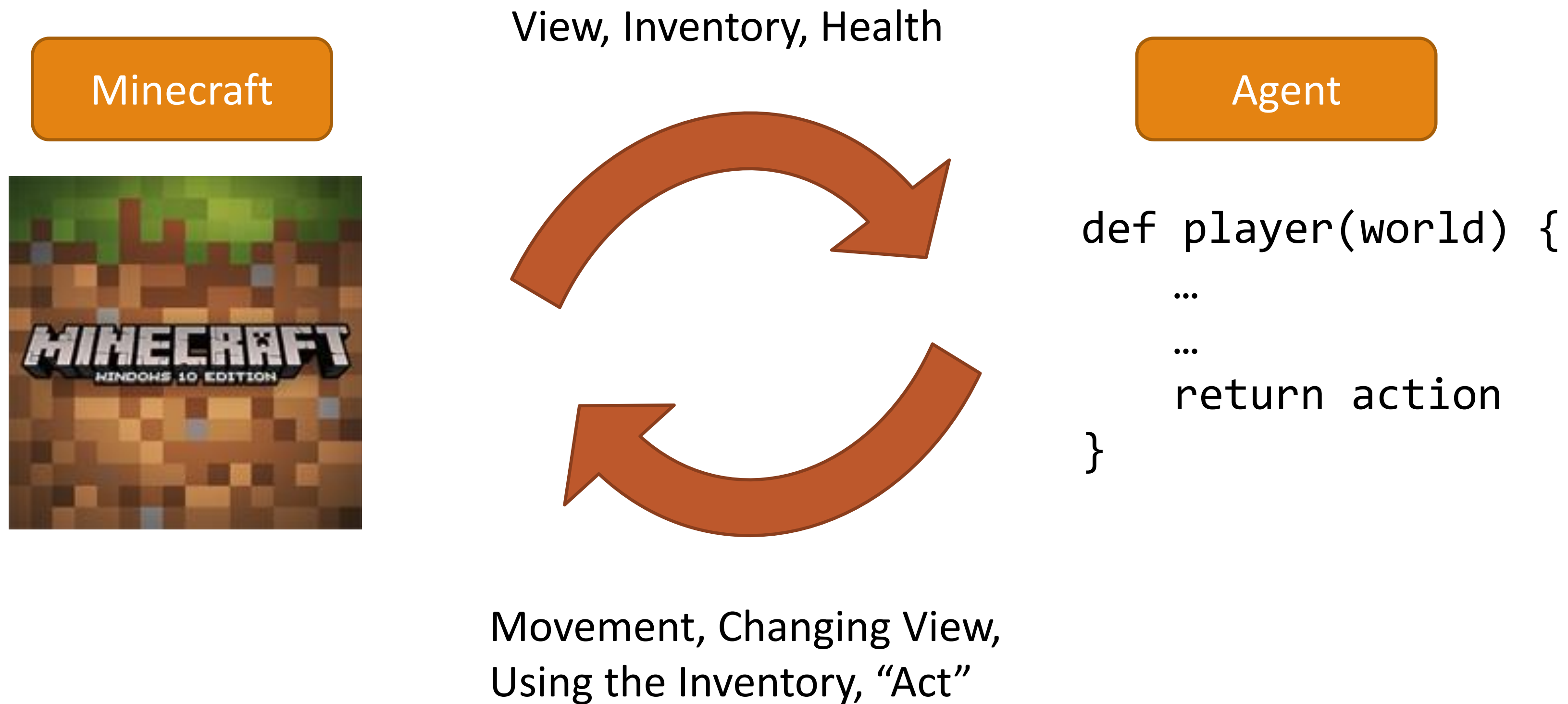


Alekh Agarwal
Researcher

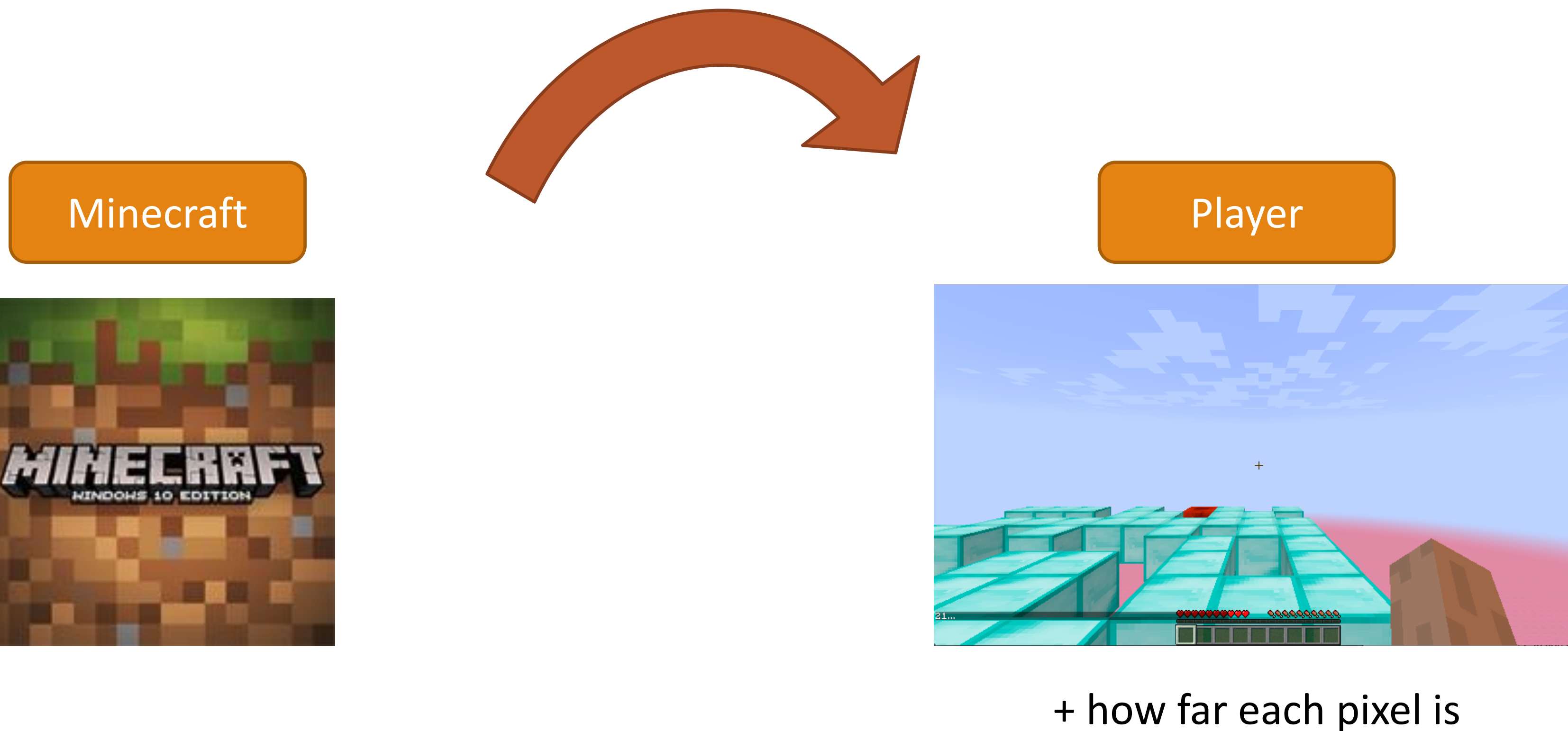
Minecraft & Player



Minecraft & ~~Player~~ Code

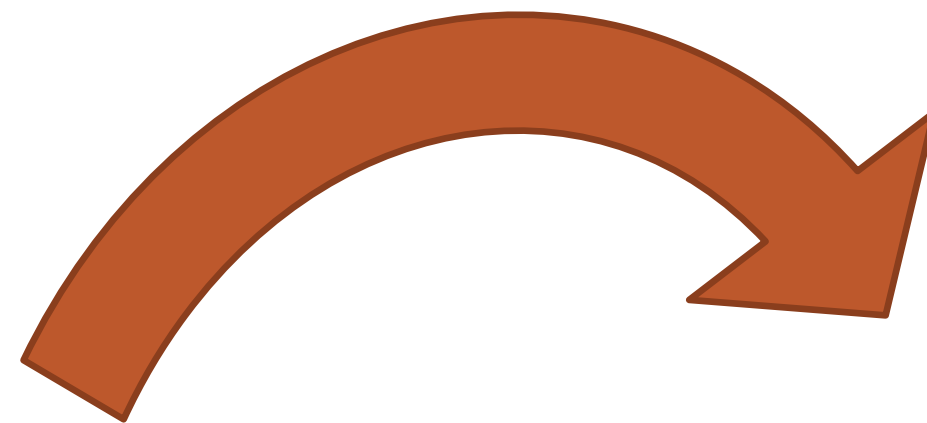


World: Player View

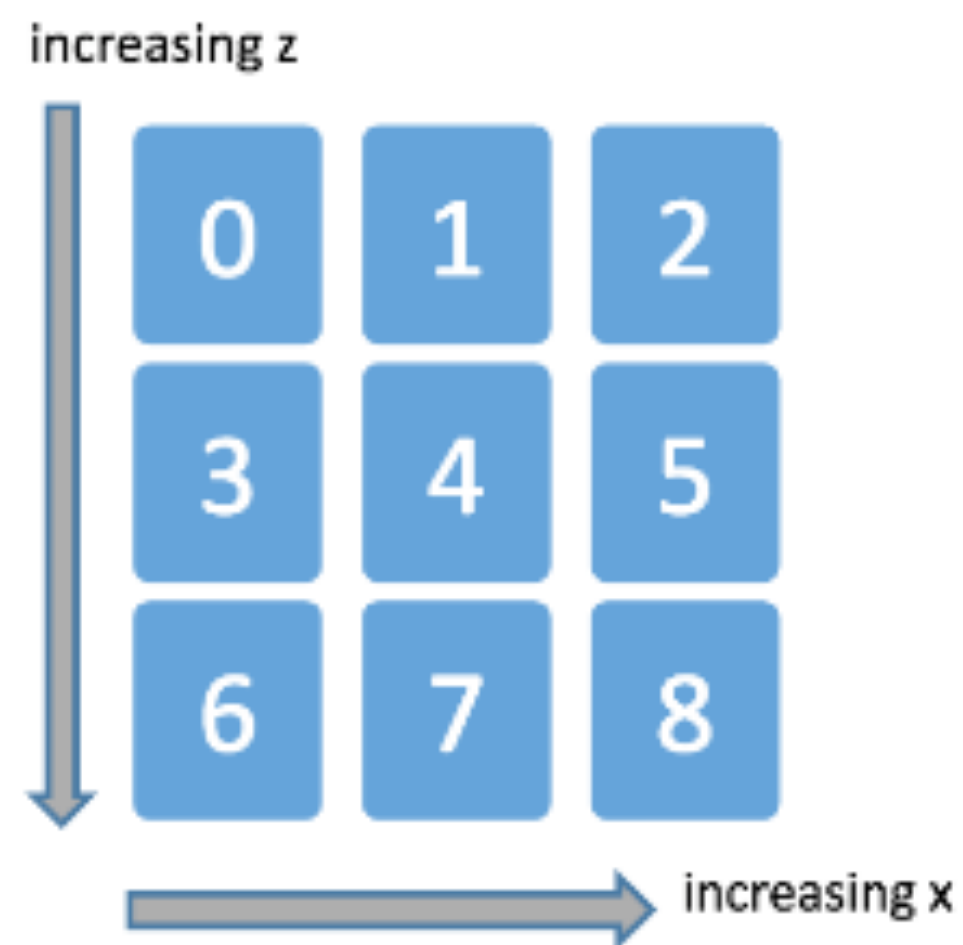


World: Simple View

Minecraft



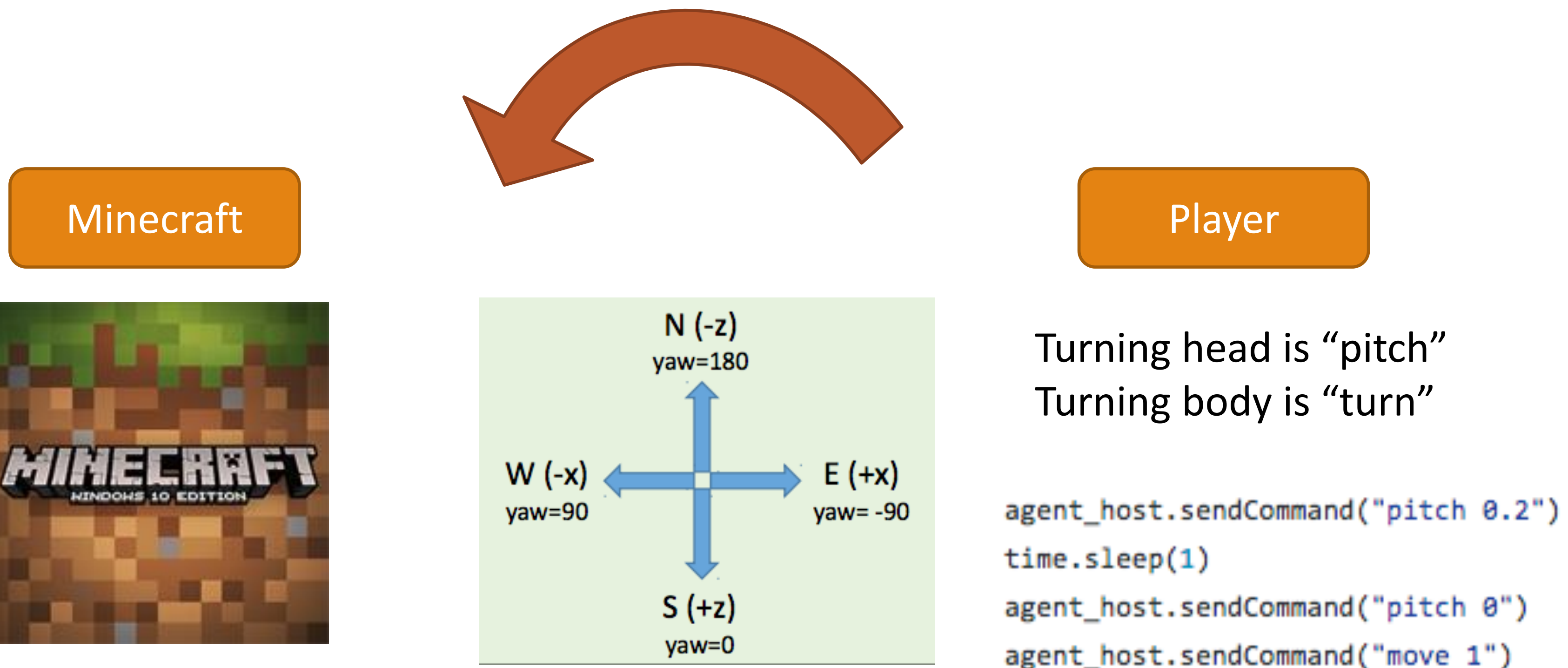
Player



```
<ObservationFromGrid>  
  <Grid name="floor3x3">  
    <min x="-1" y="-1" z="-1"/>  
    <max x="1" y="-1" z="1"/>  
  </Grid>  
</ObservationFromGrid>
```

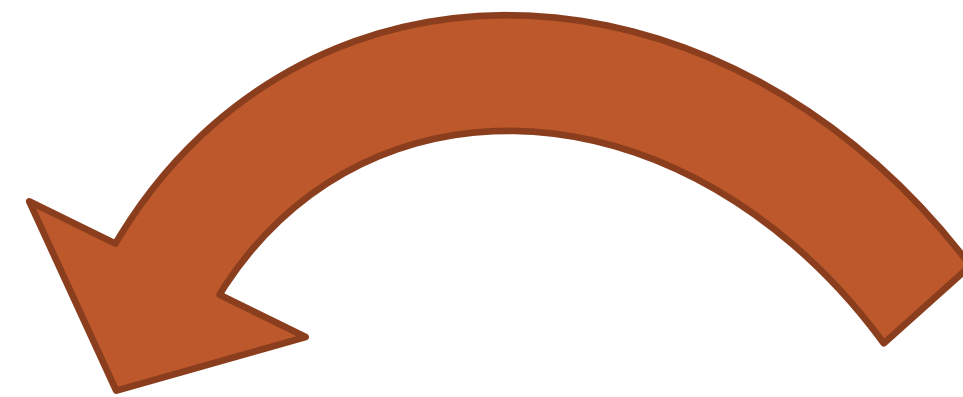
floor3x3: ['lava', 'obsidian',
'obsidian', 'lava', 'obsidian',
'obsidian', 'lava', 'obsidian',
'obsidian']

Navigation: Continuous



Navigation: Discrete

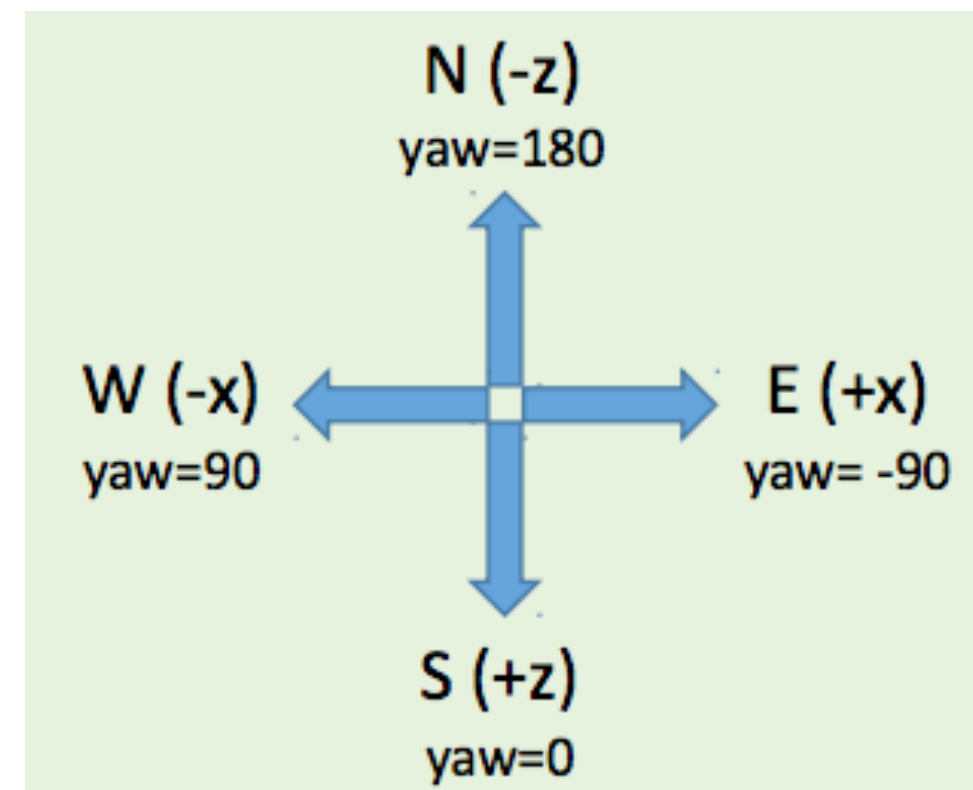
Minecraft



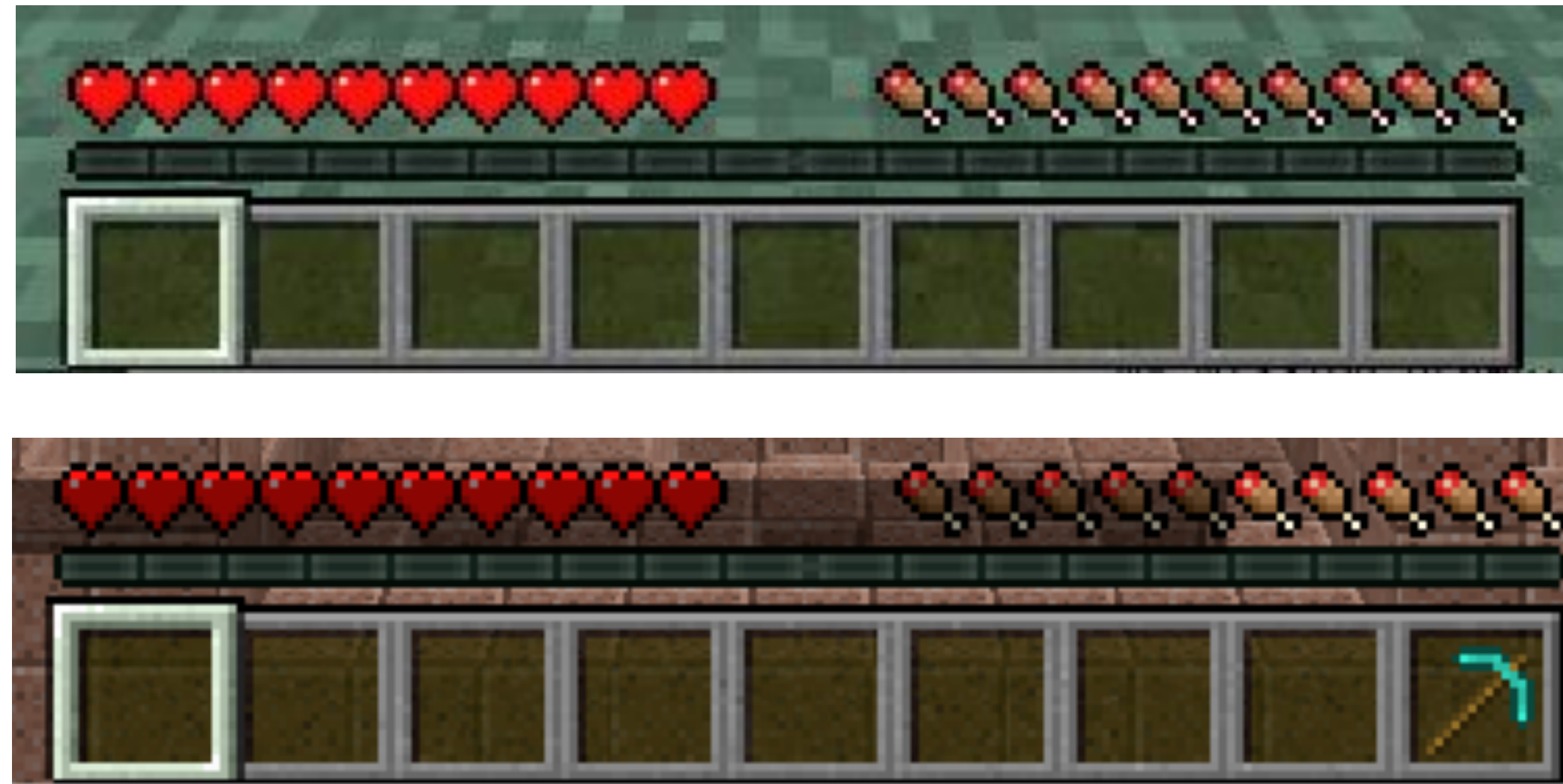
Player

Move one block at a time

["movenorth 1",
"movesouth 1",
"movewest 1",
"moveeast 1"]



Items



```
agent_host.sendCommand("hotbar.9 1") #Press the hotbar key  
agent_host.sendCommand("hotbar.9 0") #Release hotbar key - agent should now be holding diamond_pickaxe
```

Example Malmö Project

<https://www.youtube.com/watch?v=9XRL6d-yxp4>



Designing Worlds: XML

```
<DrawCuboid x1, y1, z1, x2, y2, z2, type/>  
<DrawLine x1, y1, z1, x2, y2, z2, type/>  
<DrawBlock x, y, z, type/>  
<DrawSphere x, y, z, radius, type/>  
<DrawItem x, y, z, type/>
```

```
<DrawingDecorator>  
  <DrawSphere x="-27" y="70" z="0" radius="30" type="air"/>  
</DrawingDecorator>
```


Designing Worlds: Code!

```
def Menger(xorg, yorg, zorg, size, blocktype, holetype):
    #draw solid chunk
    genstring = GenCuboid(xorg,yorg,zorg,xorg+size-1,yorg+size-1,zorg+size-1,blocktype) + "\n"
    #now remove holes
    unit = size
    while (unit >= 3):
        w=unit/3
        for i in xrange(0, size, unit):
            for j in xrange(0, size, unit):
                x=xorg+i
                y=yorg+j
                genstring += GenCuboid(x+w,y+w,zorg,(x+2*w)-1,(y+2*w)-1,zorg+size-1,holetype) + "\n"
                y=yorg+i
                z=zorg+j
                genstring += GenCuboid(xorg,y+w,z+w,xorg+size-1,(y+2*w)-1,(z+2*w)-1,holetype) + "\n"
                genstring += GenCuboid(x+w,yorg,z+w,(x+2*w)-1,yorg+size-1,(z+2*w)-1,holetype) + "\n"
            unit/=3
    return genstring
```

Designing Worlds: Code!



Designing Worlds: Code!



Designing Worlds: Mazes



Designing Worlds: Third-Party

SUPERFLAT GENERATOR
Last Update: Nov 4, 2014 (MC 1.8)

Code: 3;7,230*1,5*3,2;3;stronghold,biome_1,decoration,dl 1.8 and above

Preset: Tunneler's Dream + New Preset... Delete

Layers: Height: 237 Biome: Extreme Hills Features: 5 selected

New Layer: Enter block name or id + Add Layer

- 1 Grass Block (Layers: 236) Grass Block [2:0]
- 5 Dirt (Layers: 231-235) Dirt [3:0]
- 230 Stone (Layers: 1-230) Stone [1:0]
- 1 Bedrock (Layers: 0) Bedrock [7:0]

Like 129 Tweet G+

Layers: Height: 4 Biome: Plains Features: 1 selected

| | | |
|------------------|--------------------|--------------------|
| Ocean | Plains | Desert |
| Extreme Hills | Forest | Taiga |
| Swampland | River | Hell |
| Sky | FrozenOcean | FrozenRiver |
| Ice Plains | Ice Mountains | MushroomIsland |
| MushroomShore | Beach | DesertHills |
| ForestHills | TaigaHills | Extreme Hills Edge |
| Jungle | JungleHills | JungleEdge |
| Deep Ocean | Stone Beach | Cold Beach |
| Birch Forest | Birch Forest Hills | Roofed Forest |
| Cold Taiga | Cold Taiga Hills | Mega Taiga |
| Mega Taiga Hills | Extreme Hills+ | Savanna |
| Savanna Plateau | Mesa | Mesa Plateau F |

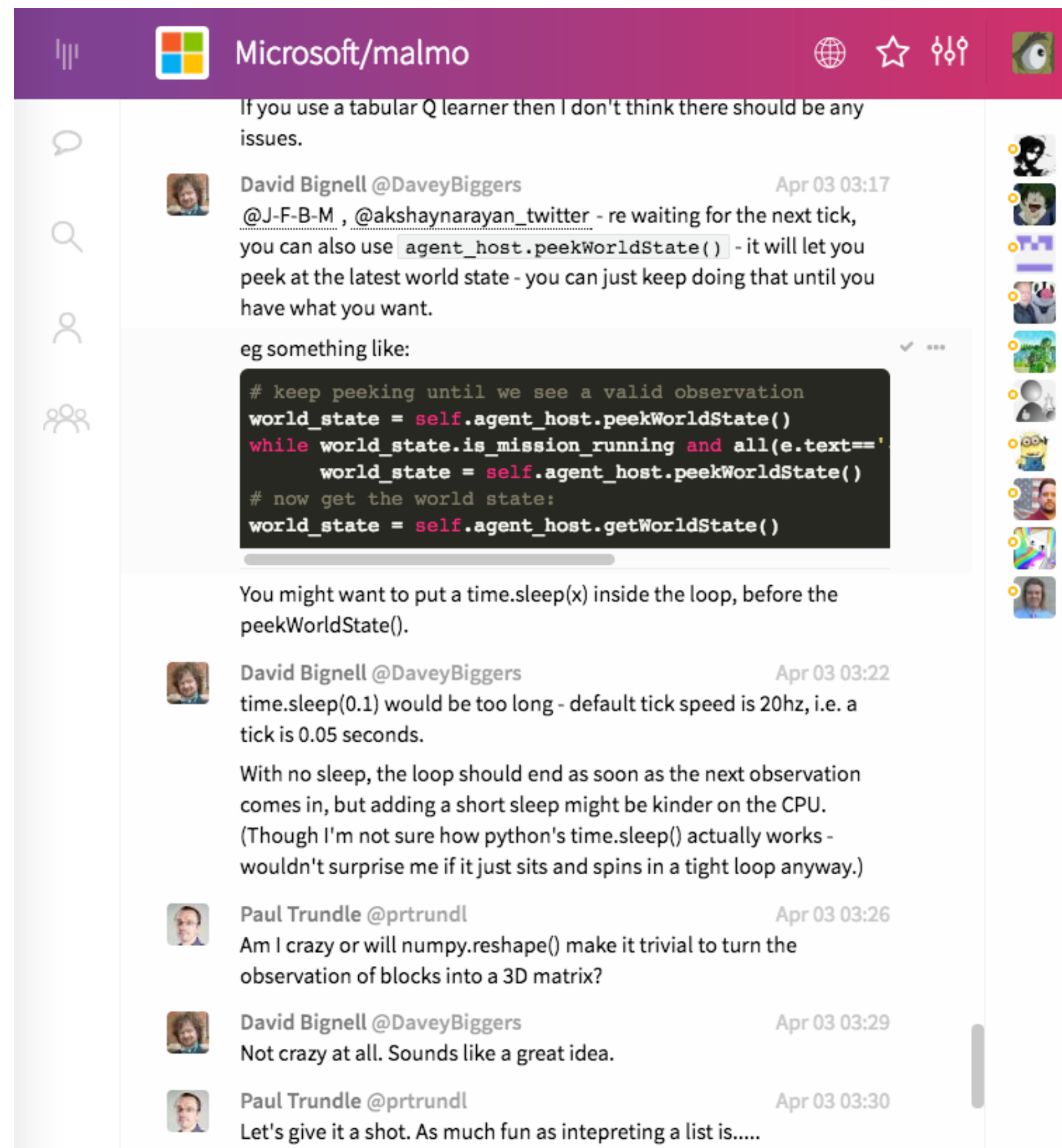
Malmo Github

<https://github.com/Microsoft/malmo#getting-started>

The screenshot shows the GitHub repository page for Microsoft/malmo. At the top, there's a navigation bar with 'This repository', a search bar, and links for 'Pull requests', 'Issues', and 'Gist'. Below this, the repository name 'Microsoft / malmo' is displayed along with statistics: 271 Watchers, 2,374 Stars, and 349 Forks. A secondary navigation bar includes 'Code', 'Issues (66)', 'Pull requests (4)', 'Projects (1)', 'Wiki', 'Pulse', and 'Graphs'. The main content area features a description of Project Malmo as a platform for AI experimentation on Minecraft, with a link to installation instructions. Below the description, repository statistics are shown: 1,012 commits, 12 branches, 14 releases, 14 contributors, and MIT license. Action buttons for 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download' are present. A commit history table follows, listing recent changes by DaveyBiggers.

| Commit | Message | Time |
|--------------|---|--------------|
| DaveyBiggers | Merge pull request #481 from Microsoft/ps_fixes | 21 days ago |
| DaveyBiggers | Removed experimental sudo | 4 months ago |
| DaveyBiggers | Applied MIT license. | 9 months ago |
| DaveyBiggers | Updated python samples for new entity data. | a month ago |
| DaveyBiggers | Added distance to ObservationFromRay | 21 days ago |

Malmo Gitter



Microsoft/malmo

If you use a tabular Q learner then I don't think there should be any issues.

David Bignell @DaveyBiggers Apr 03 03:17
@J-F-B-M , @akshaynarayan_twitter - re waiting for the next tick, you can also use `agent_host.peekWorldState()` - it will let you peek at the latest world state - you can just keep doing that until you have what you want.

eg something like:

```
# keep peeking until we see a valid observation
world_state = self.agent_host.peekWorldState()
while world_state.is_mission_running and all(e.text==
    world_state = self.agent_host.peekWorldState()
# now get the world state:
world_state = self.agent_host.getWorldState()
```

You might want to put a `time.sleep(x)` inside the loop, before the `peekWorldState()`.

David Bignell @DaveyBiggers Apr 03 03:22
`time.sleep(0.1)` would be too long - default tick speed is 20hz, i.e. a tick is 0.05 seconds.

With no sleep, the loop should end as soon as the next observation comes in, but adding a short sleep might be kinder on the CPU. (Though I'm not sure how python's `time.sleep()` actually works - wouldn't surprise me if it just sits and spins in a tight loop anyway.)

Paul Trundle @prtrundl Apr 03 03:26
Am I crazy or will `numpy.reshape()` make it trivial to turn the observation of blocks into a 3D matrix?

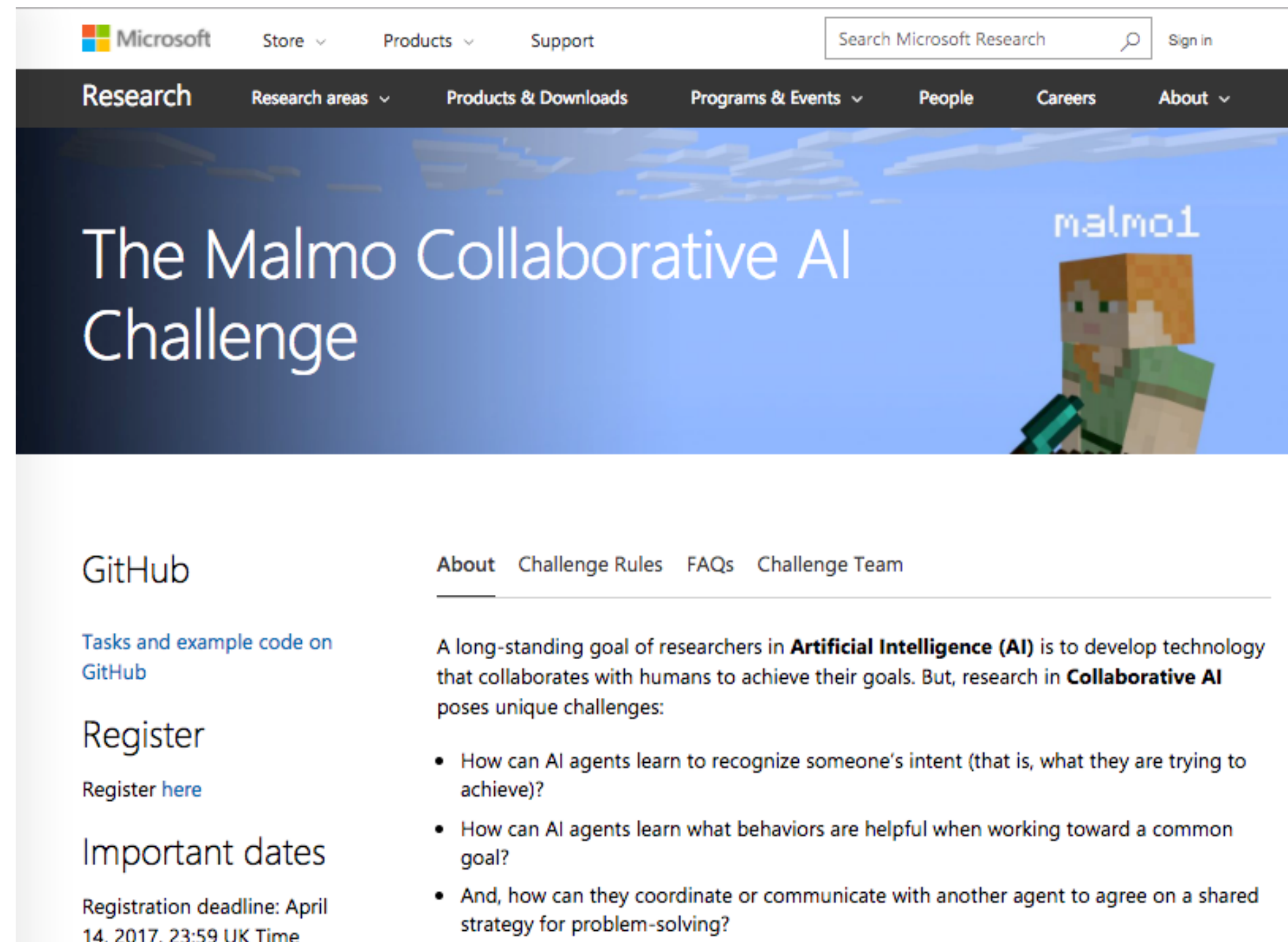
David Bignell @DaveyBiggers Apr 03 03:29
Not crazy at all. Sounds like a great idea.

Paul Trundle @prtrundl Apr 03 03:30
Let's give it a shot. As much fun as interpreting a list is.....

<https://gitter.im/Microsoft/malmo>

The Malmo Challenge

<https://www.microsoft.com/en-us/research/academic-program/collaborative-ai-challenge/>



The screenshot shows the Microsoft Research website for the Malmo Collaborative AI Challenge. The page features a dark blue header with the Microsoft logo and navigation links. The main content area has a blue background with a Minecraft character and the text "The Malmo Collaborative AI Challenge". Below this, there are sections for "GitHub", "Register", and "Important dates".

Microsoft Store Products Support Search Microsoft Research Sign in

Research Research areas Products & Downloads Programs & Events People Careers About

The Malmo Collaborative AI Challenge

malmo1

[GitHub](#)

[Tasks and example code on GitHub](#)

Register

[Register here](#)

Important dates

Registration deadline: April 14, 2017, 23:59 UK Time

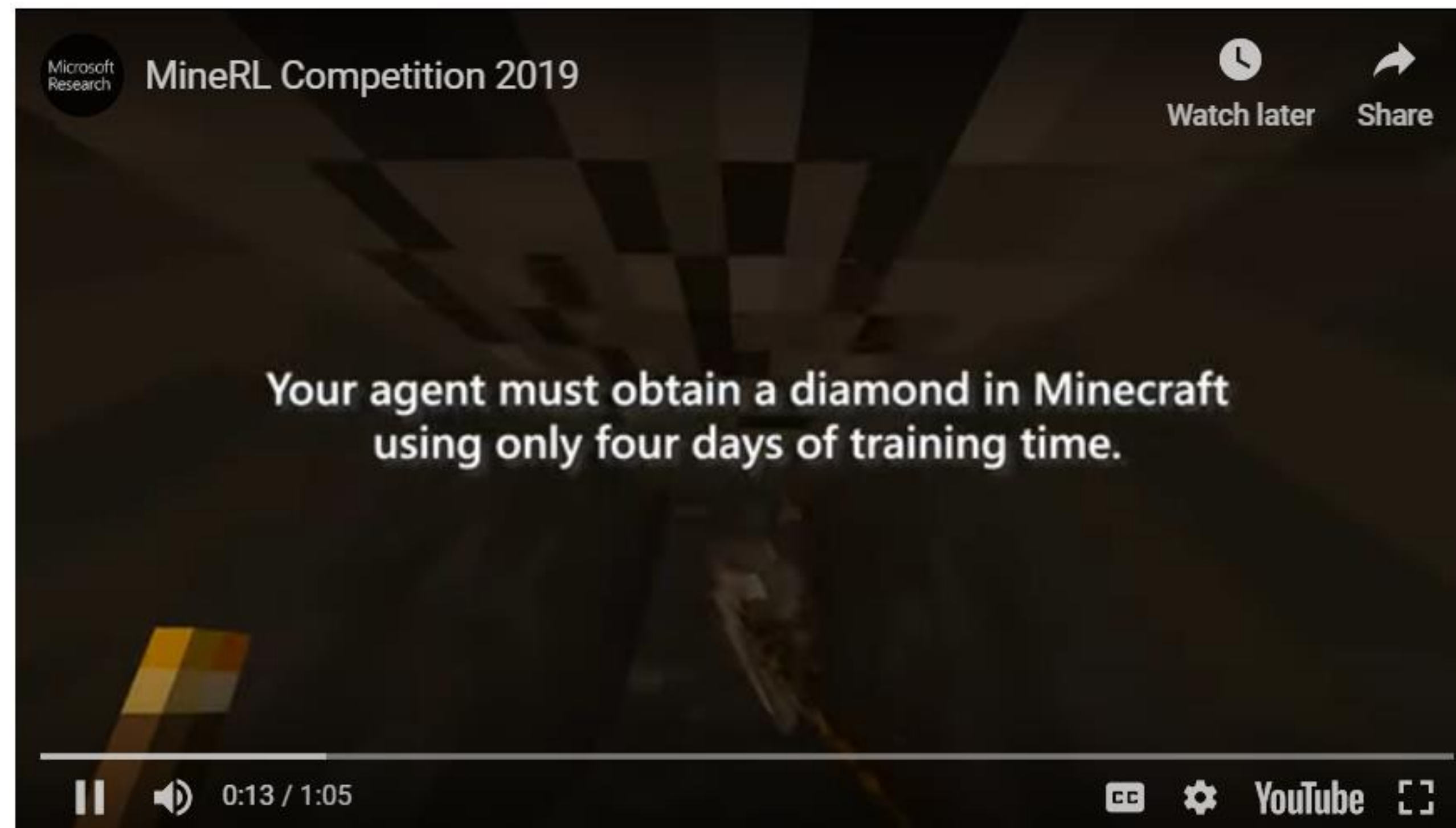
[About](#) [Challenge Rules](#) [FAQs](#) [Challenge Team](#)

A long-standing goal of researchers in **Artificial Intelligence (AI)** is to develop technology that collaborates with humans to achieve their goals. But, research in **Collaborative AI** poses unique challenges:

- How can AI agents learn to recognize someone's intent (that is, what they are trying to achieve)?
- How can AI agents learn what behaviors are helpful when working toward a common goal?
- And, how can they coordinate or communicate with another agent to agree on a shared strategy for problem-solving?

MineRL Competition

Advanced RL Competition, by extending Malmo



<https://www.youtube.com/watch?v=ggo1WAldyq0>

Dataset

Examples of humans doing tasks, and more importantly, **subtasks!**

Navigate:



Navigate Extreme:



Treechop:



Bed:



Diamond:



Meat:



Iron Pickaxe:



Survival:



Evaluation

How would YOU define that your project was a success?



Quantitative
Evaluation

Numerical Metrics:

- Accuracy, F1, AUC, ...
- Time to “run”, time to “train”

Baselines:

- What would be currently used?
- What are reasonable “simpler” methods?

By how much amount?

We hope to improve the METRIC by AMOUNT over BASELINE!

(I won't hold you to it, just want you to think about it)

Evaluation

How would YOU define that your project was a success?



Qualitative
Evaluation

Simple Example Cases:

- What are examples that your idea will “definitely” work on?
- What is the expected output on these?

Error Analysis and Introspection:

- Are there plots/figures to verify the behavior?
- If it doesn't work, how will you improve it?

The Super-Impressive Example

- What is the best example? “awesome if it works”
- E.g. something that perfectly captures your idea!

You will have doubts!

Is it too simple?

Is it too ambitious?

Is there data to train my classifier?

Is there a different algorithm I should use?

Is my evaluation inappropriate?

Can I only use off-the-shelf code?

Every team has to meet me during Week 4.

I am available for more appointments every week

Use
Campuswire!

Previous Projects

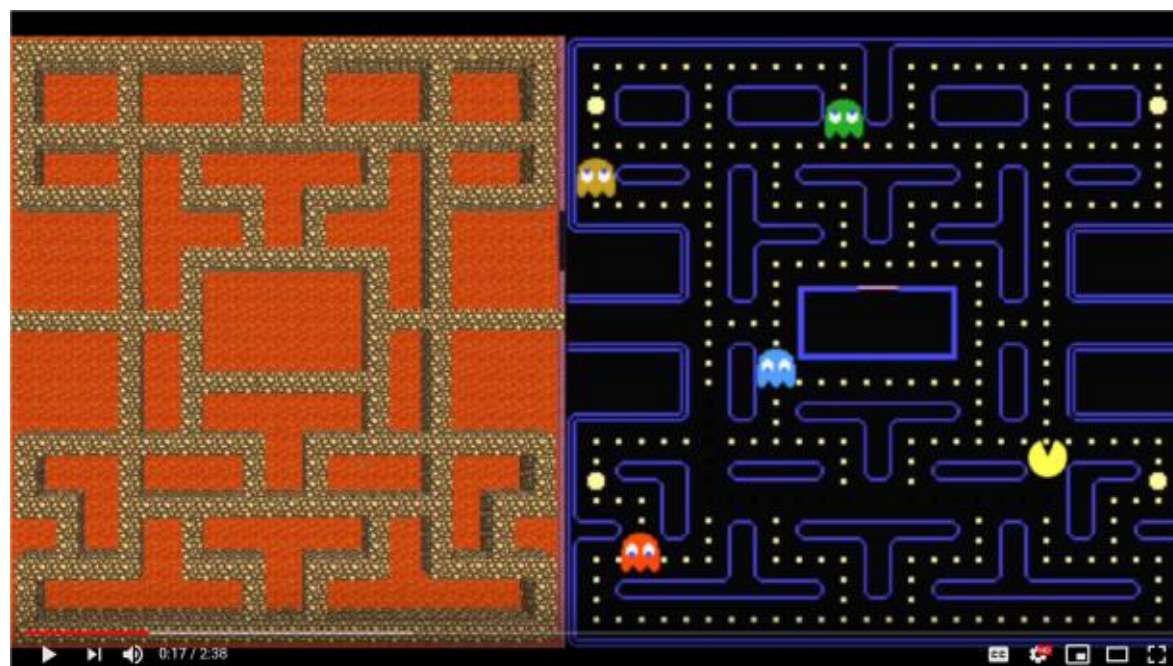
- Full list will be posted on Canvas soon. Meanwhile,



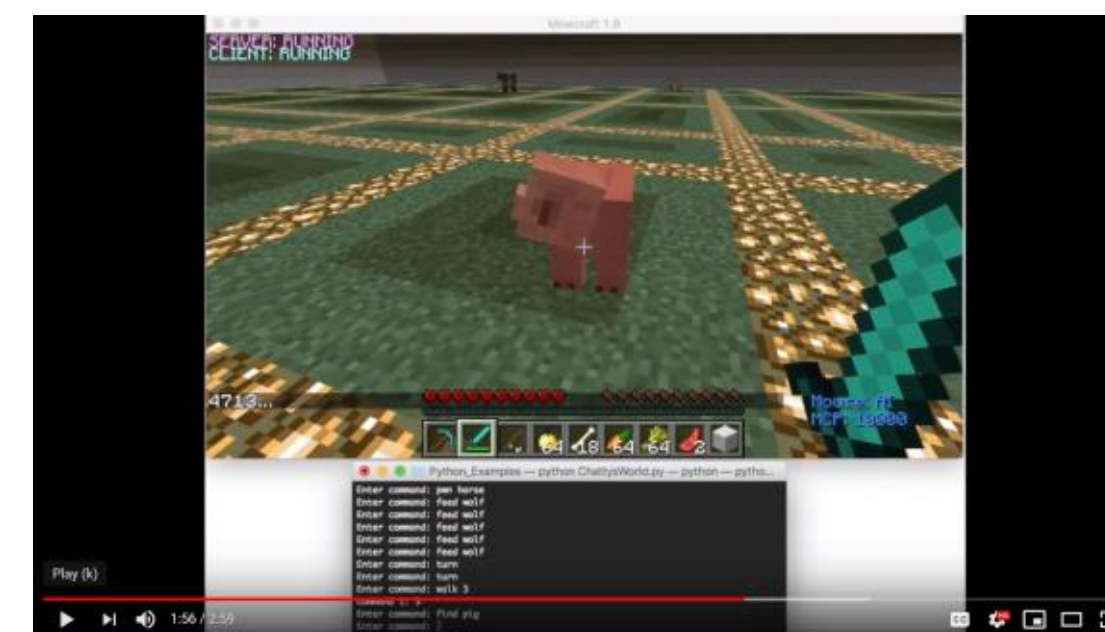
<https://www.youtube.com/watch?v=JkVa7xYHzVA>



https://www.youtube.com/watch?v=QOfay_gvvJ0



<https://www.youtube.com/watch?v=sS253RfbM3s>



<https://www.youtube.com/watch?v=ulUQLIo7MJY>

Reinforcement Learning

Agent learns to do things by trying things, and succeeding/failing

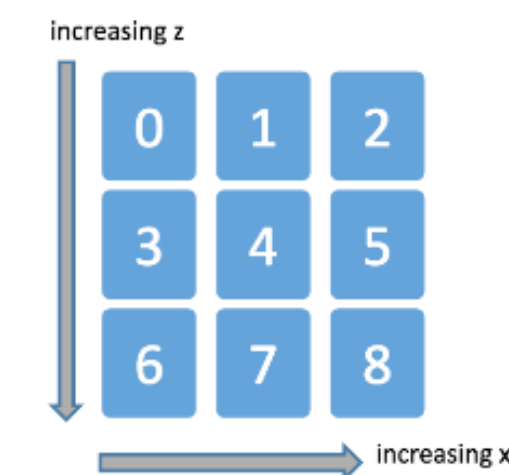
Observation

What the agent sees



Action

What the agent can do



Reward

What the agent likes/dislikes

New Item++
No Item-

Goal++
Died---

Reinforcement Learning

Agent learns to do things by trying things, and succeeding/failing

Navigation

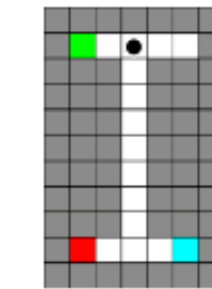
- Explore the map without dying
- Solve mazes
- Learn the best way home from anywhere
- Get to the highest hill in the map

Learn Recipes

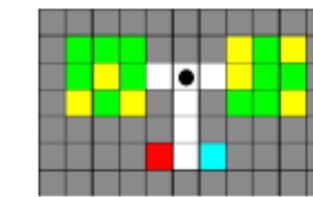
- Figure out best way to make items
- Without any knowledge of the recipes

Combat

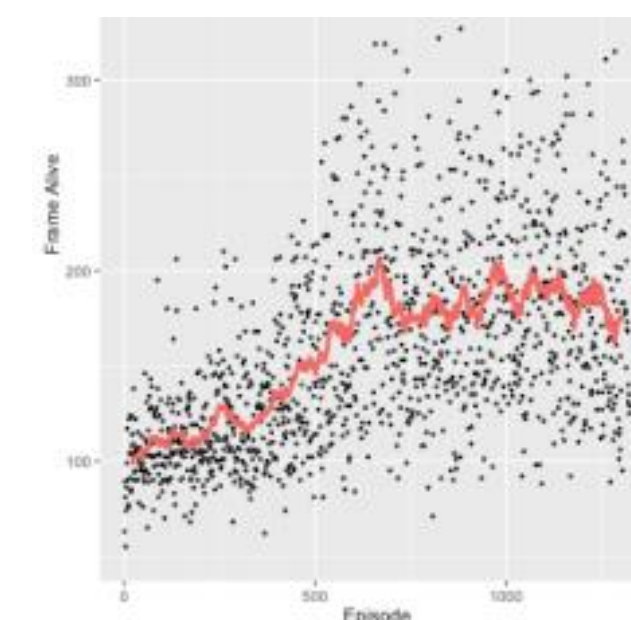
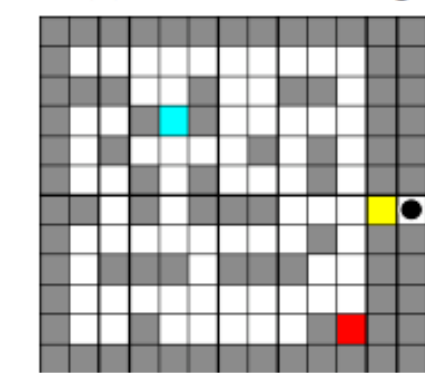
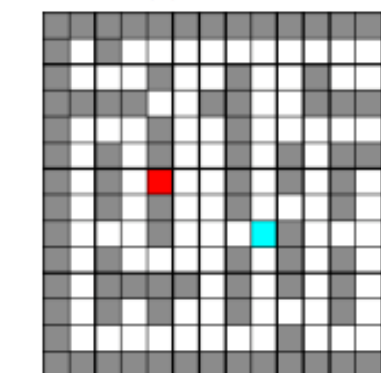
- Learn to hide/find shelter
- Learn to fight, [example paper](#)



(a) I-Maze



(b) Pattern Matching



Describe the Scene



Houses and a pig on a grassy field during the day.

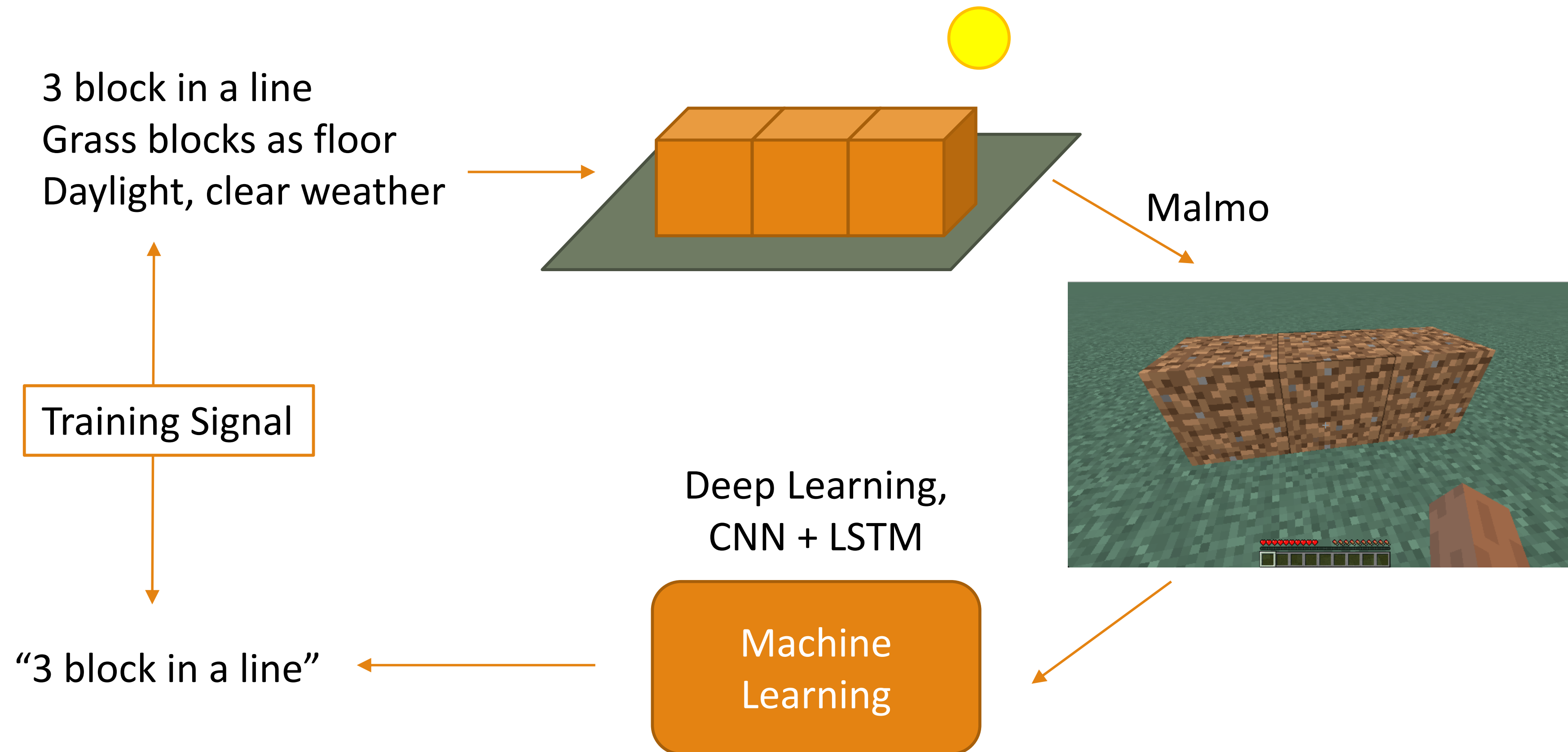
Pig staring at me in a village.

Live Commentator



“Hit a rabbit”

How is this even possible?



Natural Language Navigation



Quite Difficult!



- > Go forward till you hit a wall
- > Go to the pig
- > Go to the house on the right
- > Go behind the house

trivial
↑
↓
hardest

Recipe Planners

Inventory



“Need”(s)



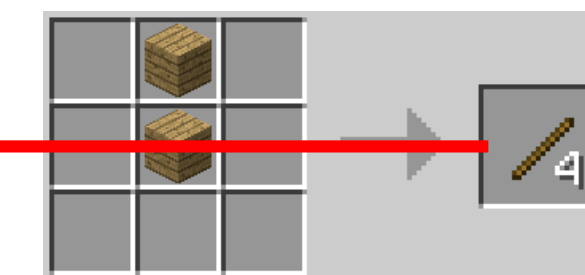
Steps



~~> Get 2 wood planks~~



~~> Make a stick~~



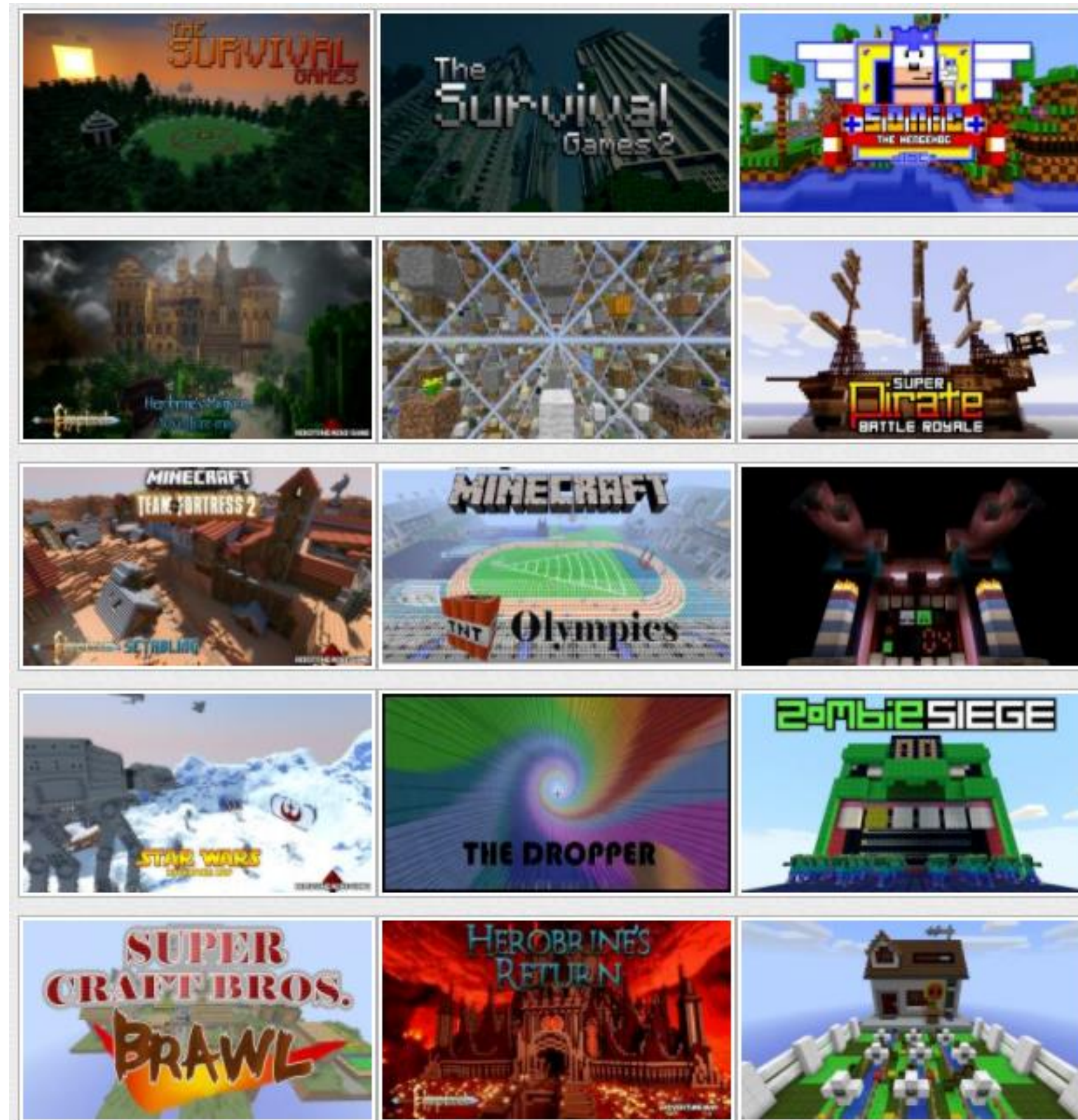
~~> Get 2 diamonds~~



> Make diamond sword



Lots of other possibilities



Many other games in Minecraft

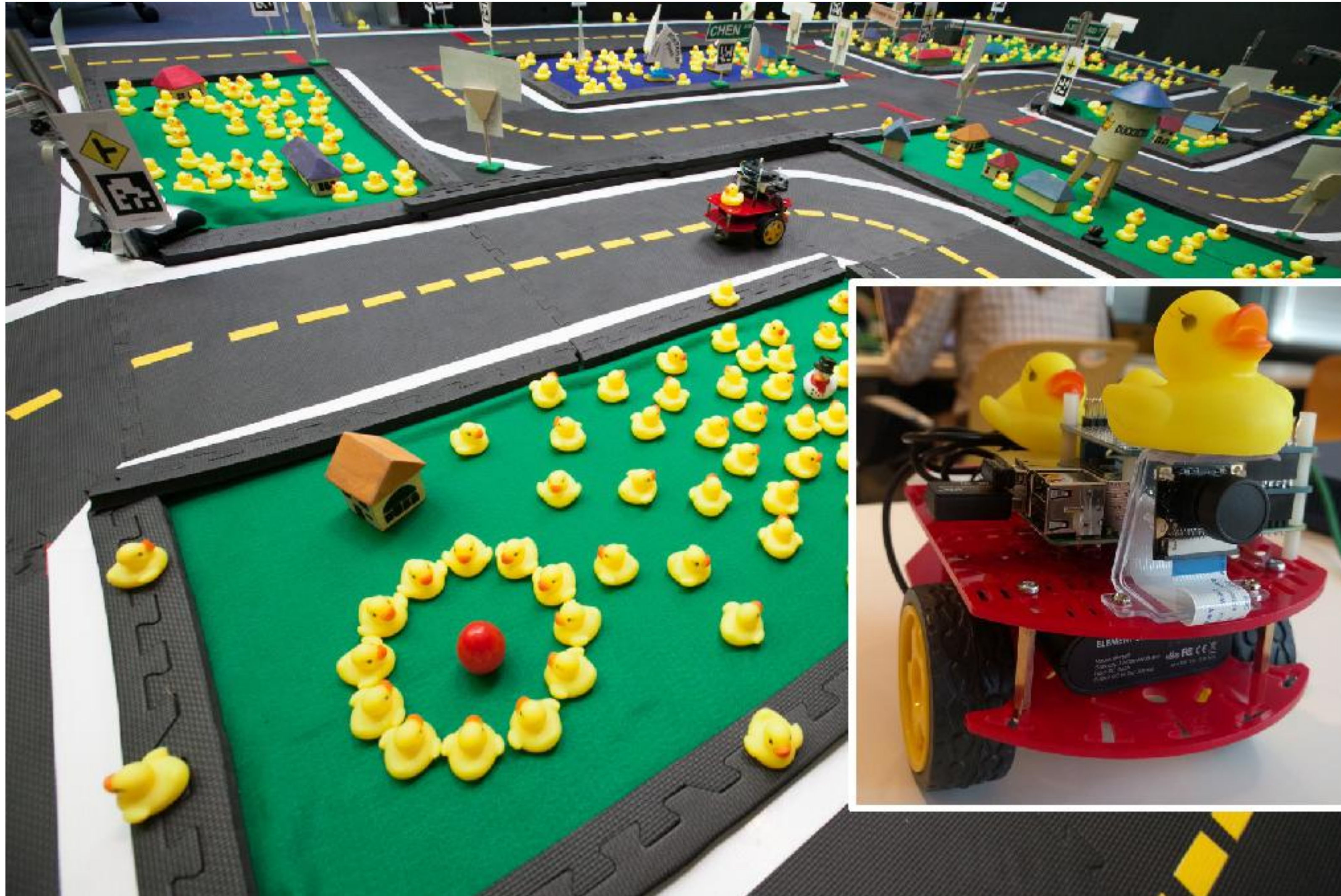
Create AI for those?

One AI that works for all of those?

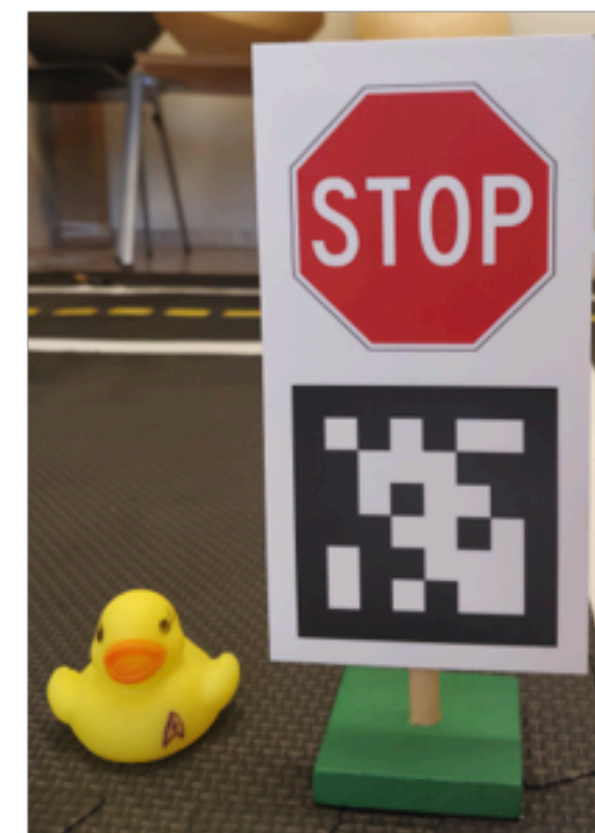
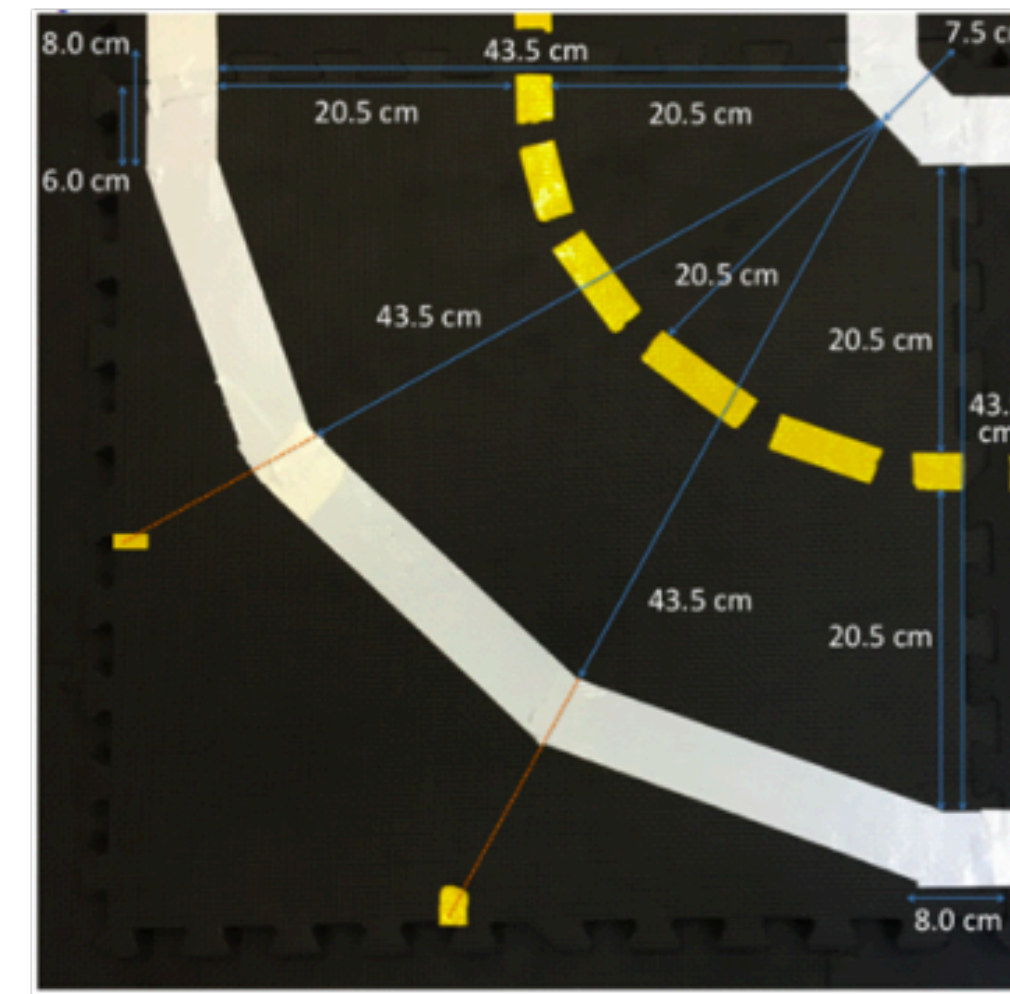
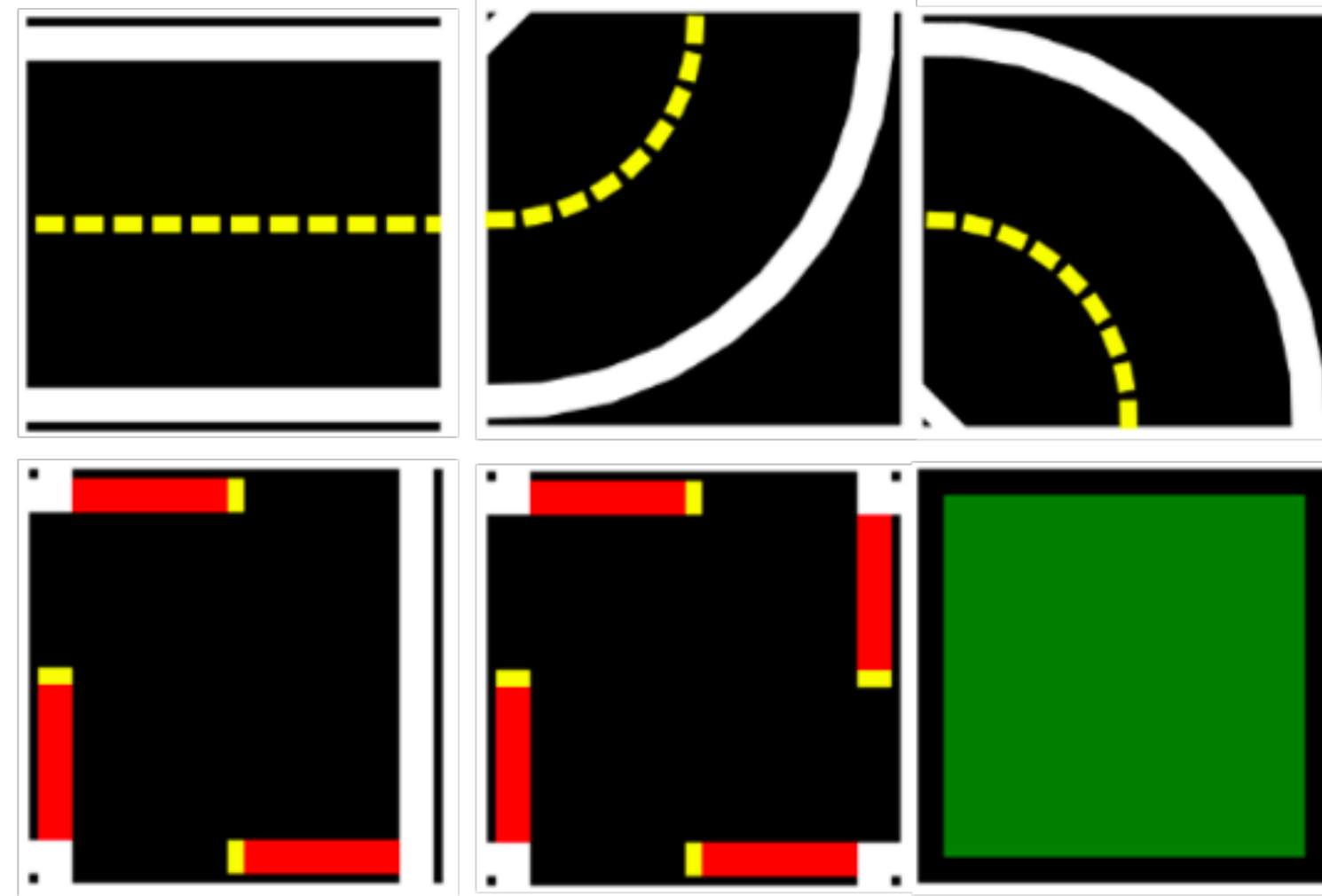
AIDO: AI Driving Olympics



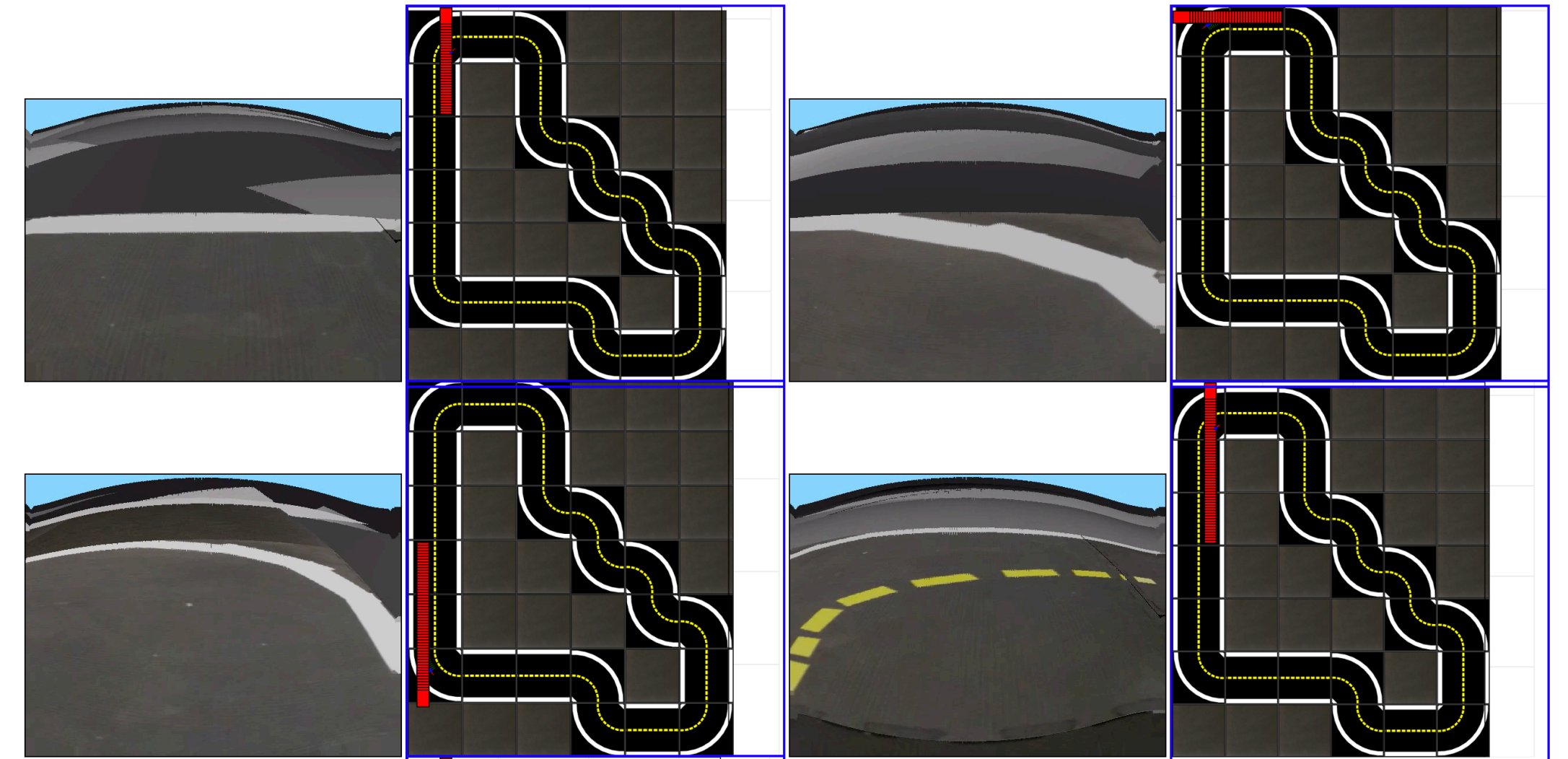
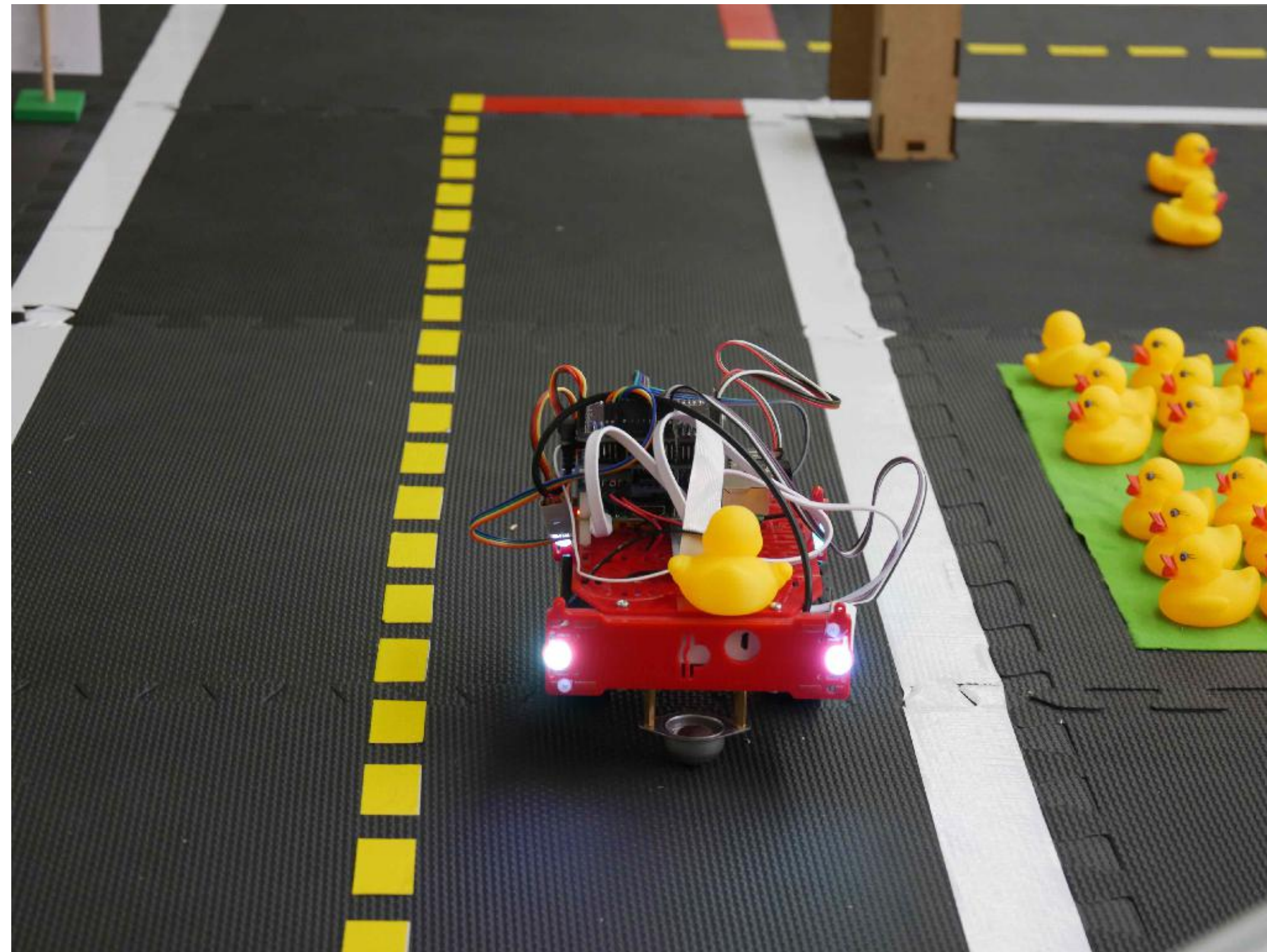
Duckietown Urban Event



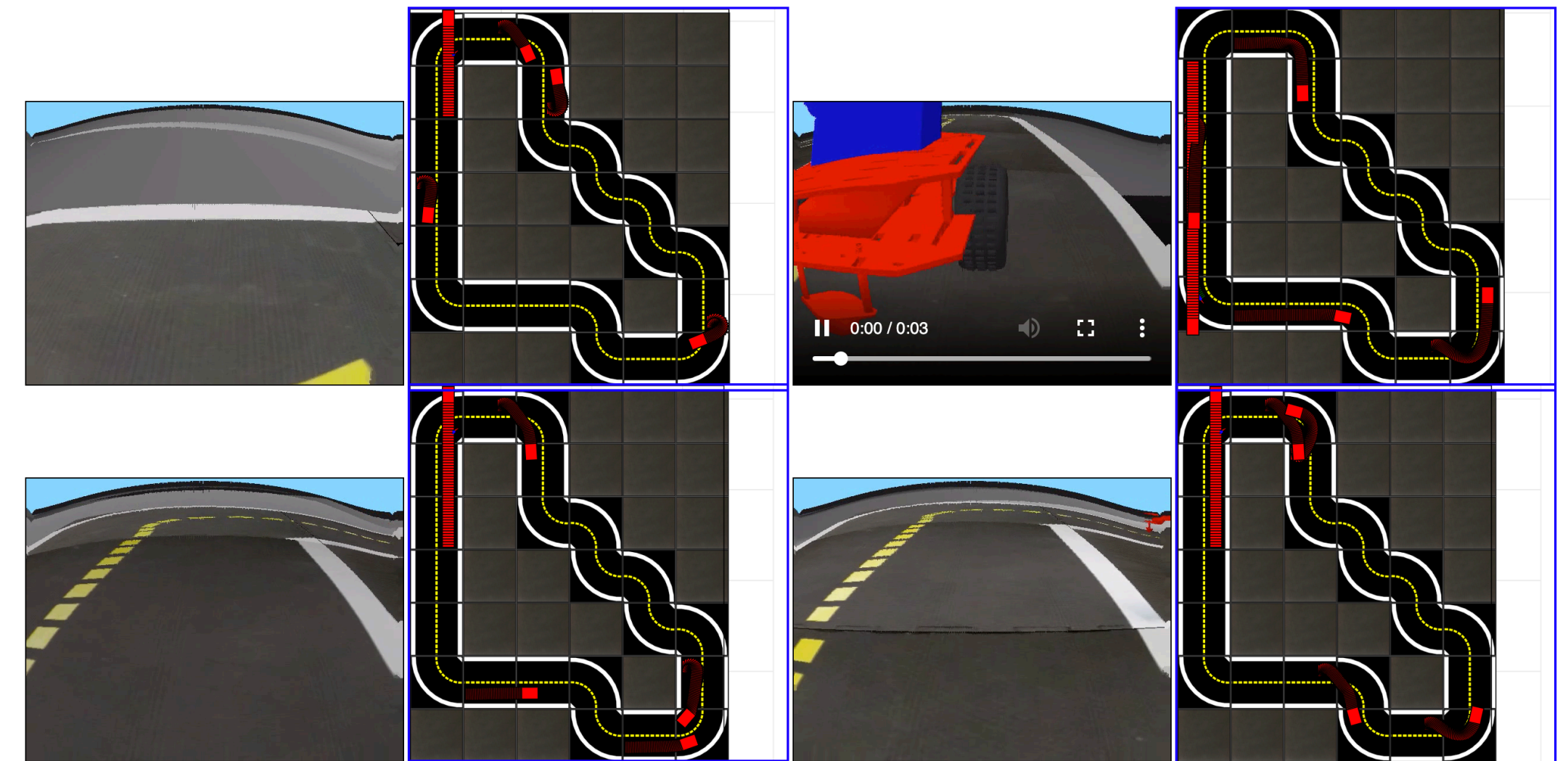
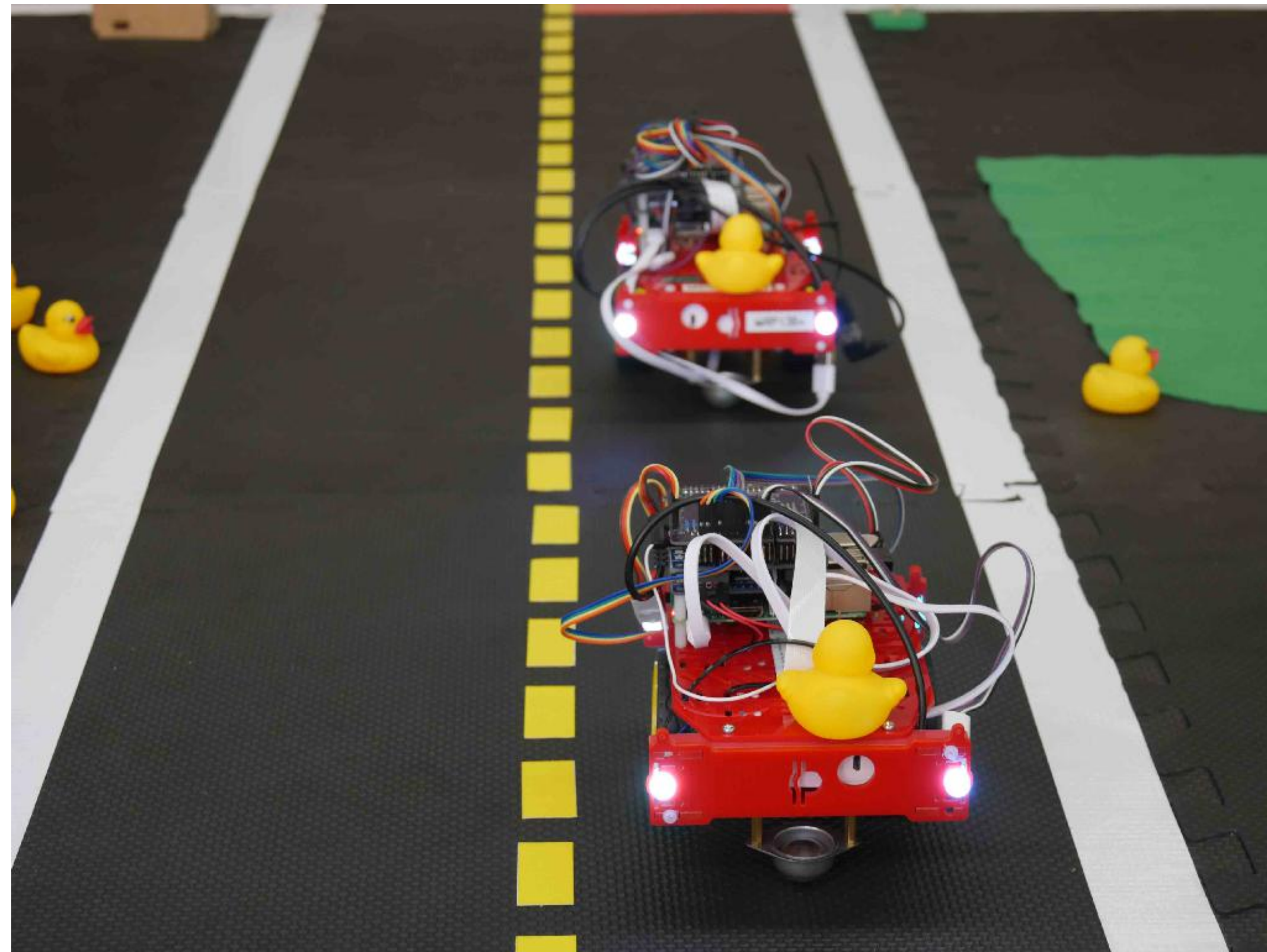
Specification



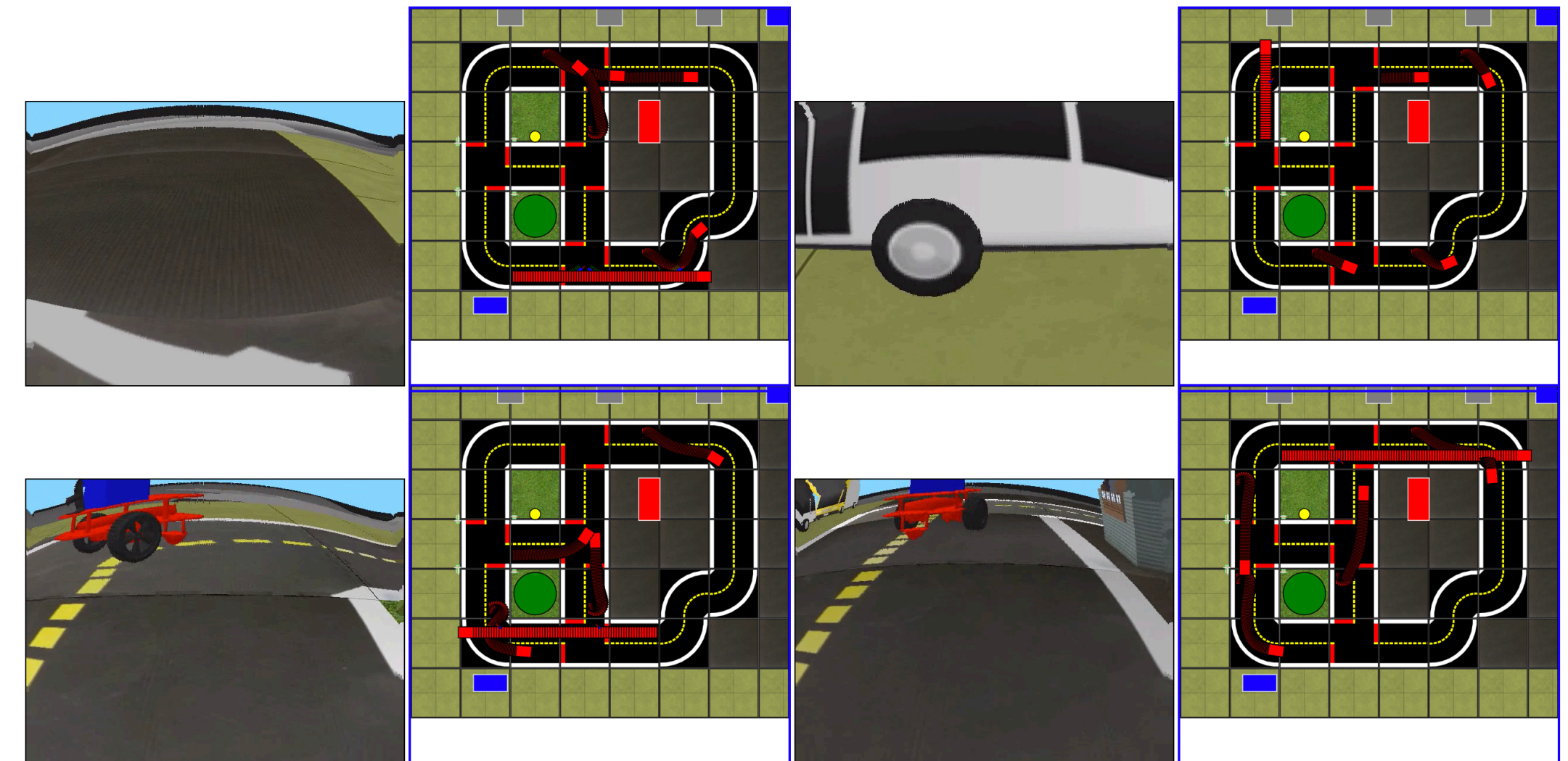
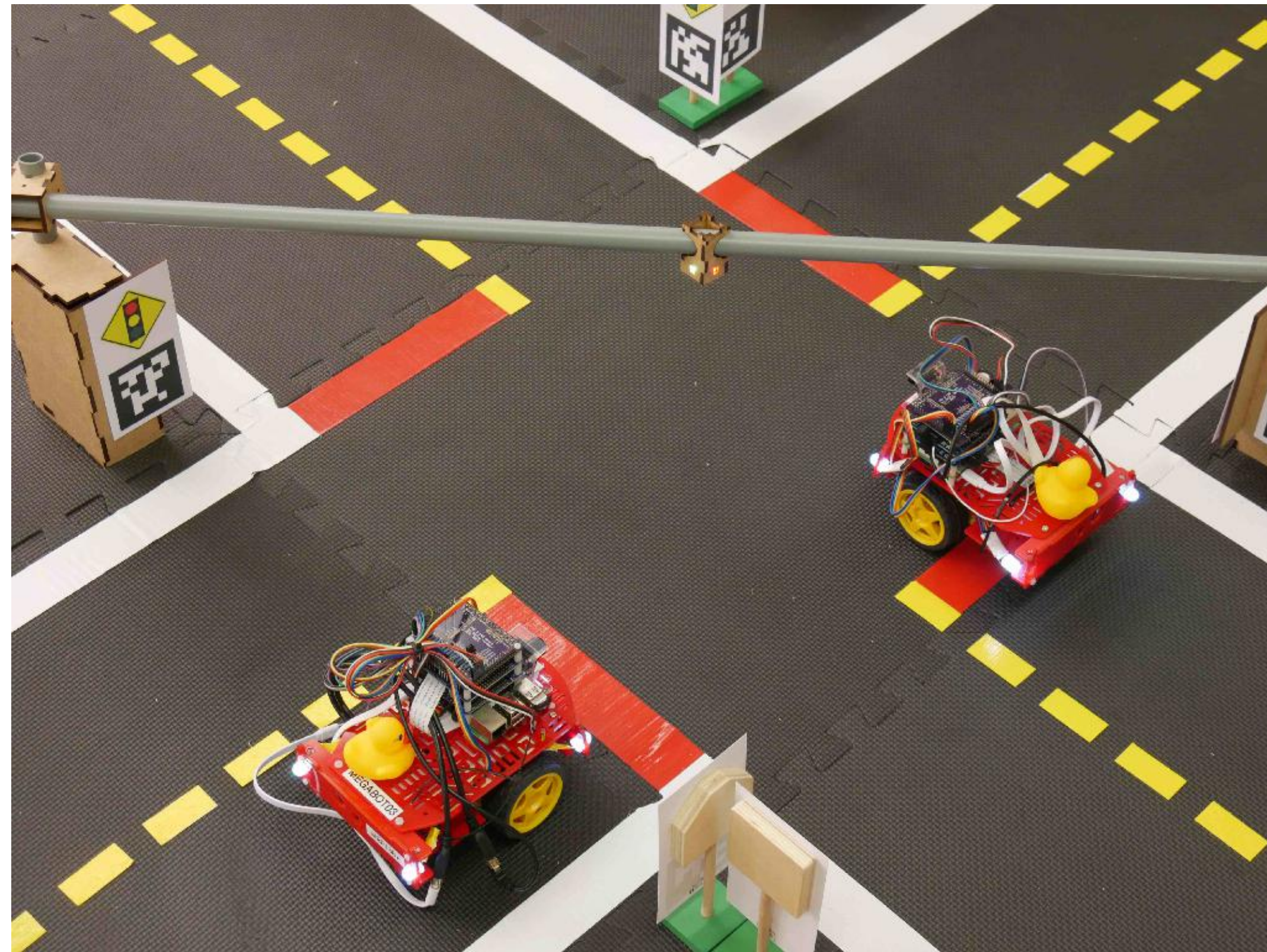
Challenge 1: Lane Following (LF)



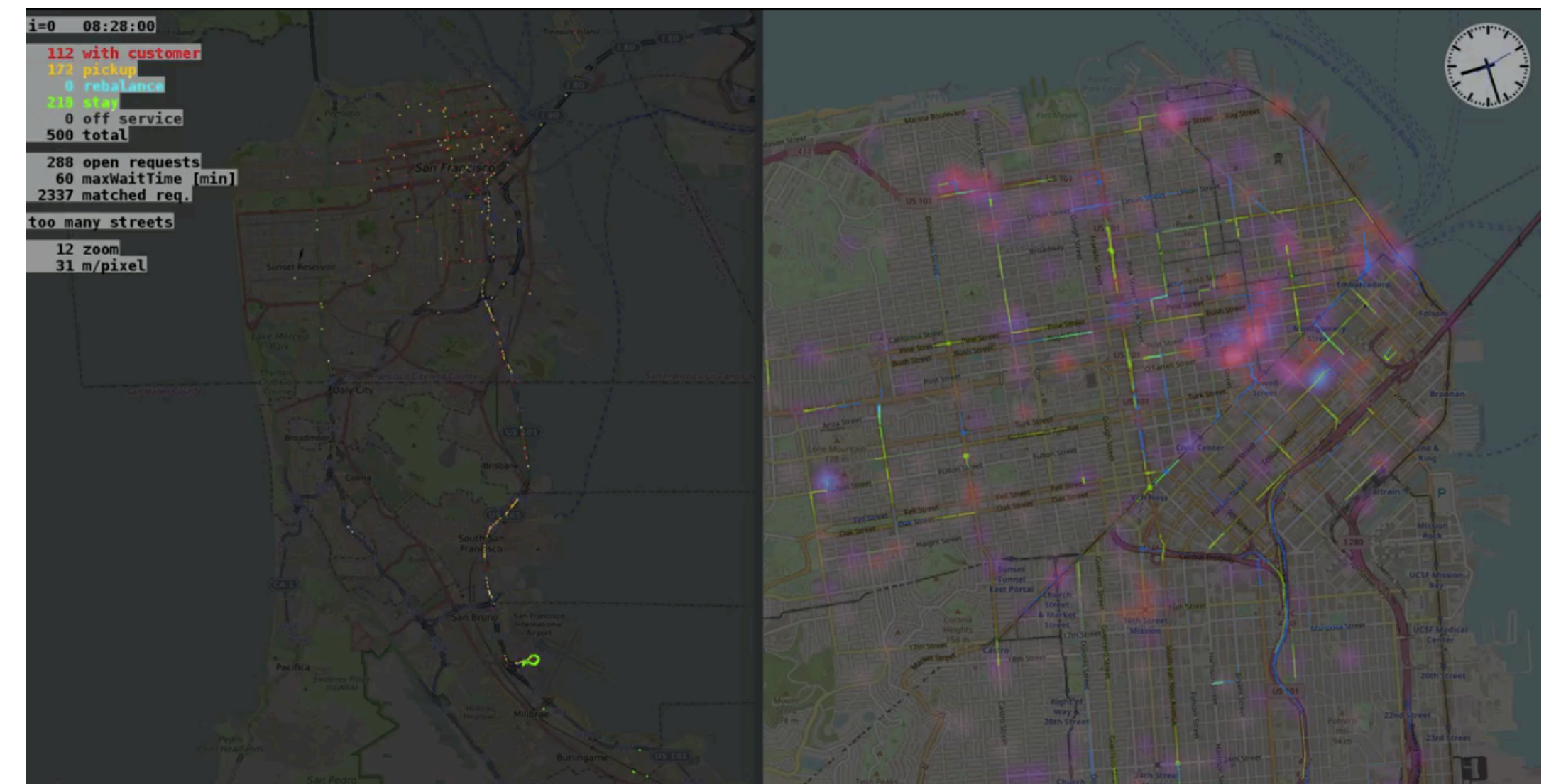
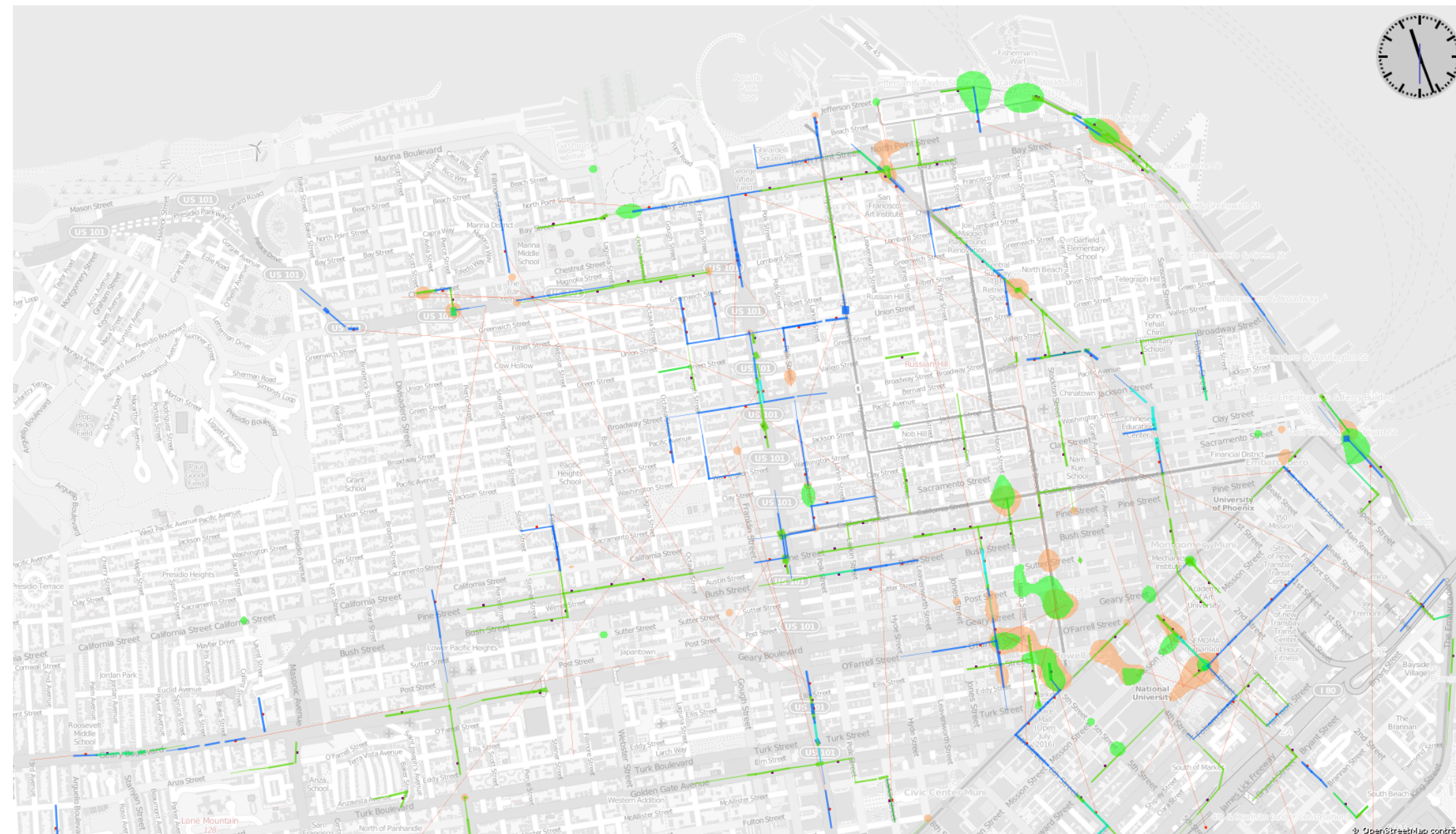
Challenge 2: LF with other Vehicles (LFV)



Challenge 3: LFV with Intersections (LFVI)



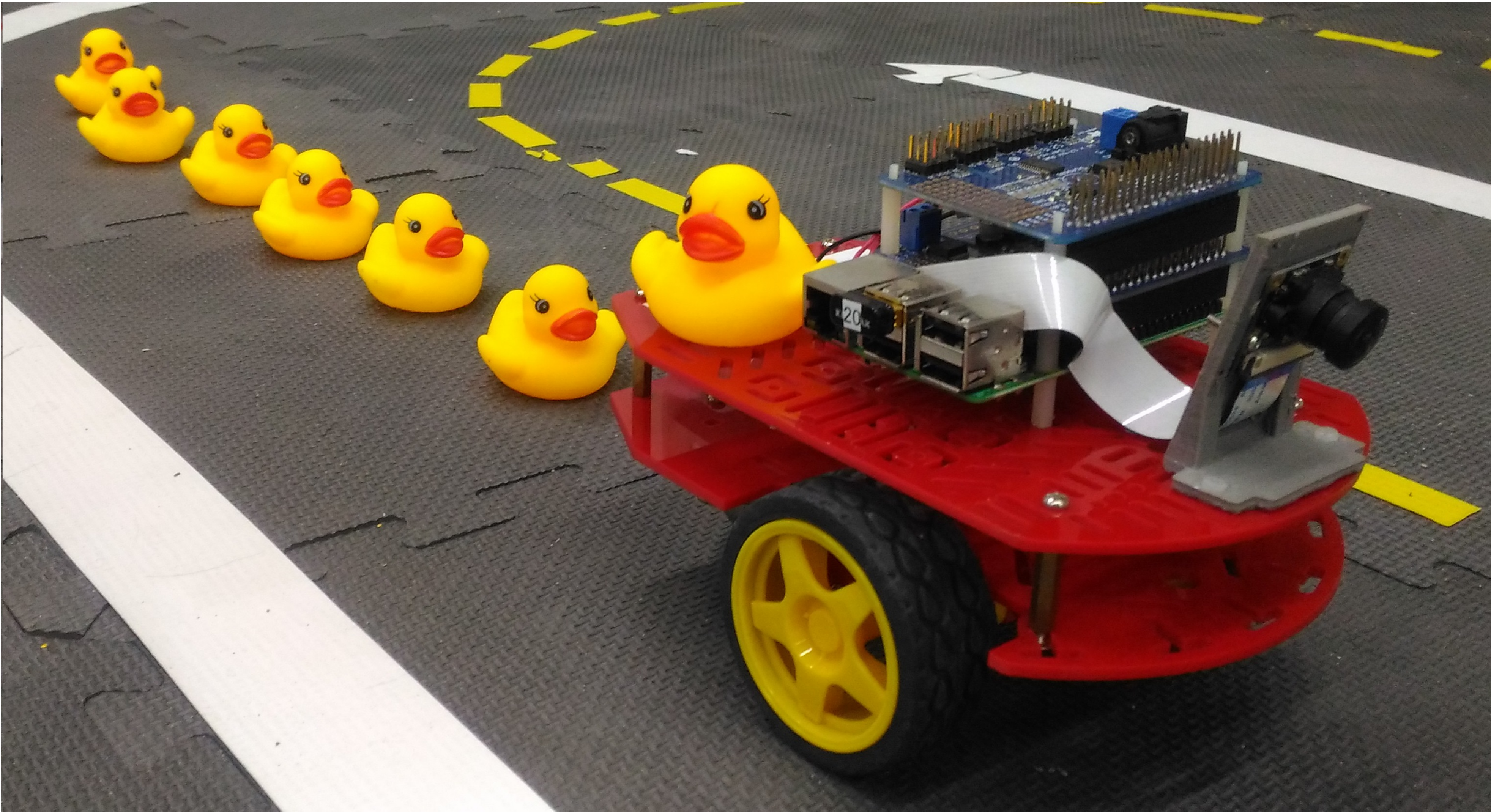
Challenge 4: Autonomous Mobility on Demand (AMoD)



Simulator



Real Duckiebots!



ColosseumRL

ColosseumRL

[Home](#) [Documentation](#) [GitHub](#) [Discord](#)



ColosseumRL

Multi-Agent Reinforcement Learning Environment Framework

Train your own reinforcement learning agent to compete against others in multiplayer games. Designed for the UC Irvine reinforcement learning competition.