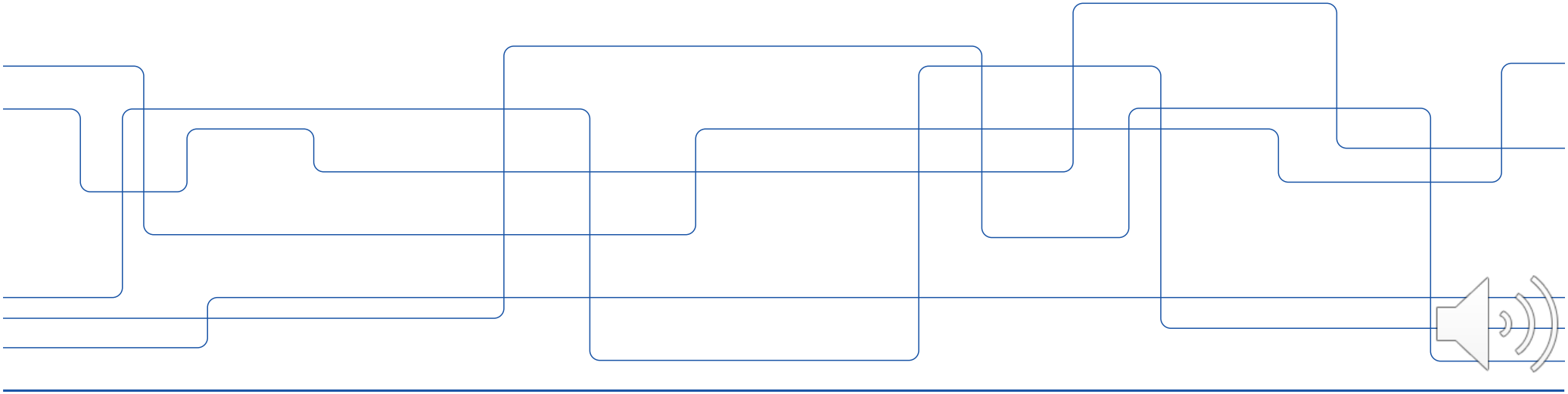
KTH ROYAL INSTITUTE
OF TECHNOLOGY

Artificial Intelligence and Multi Agent Systems DD2438

Lecture 4



Assignment 3

If your partner is slacking, let me know!

Note: a bit shorter than A1 and A2 (you have 28 days)

Module 3 (4 weeks, 100h of work)

w13 Mon 2023-03-27 13:00 - 15:00 in U21
w14 Mon 2023-04-03 13:00 - 15:00 in U21
w15 EASTER
w16 Mon 2023-04-17 13:00 - 15:00 in U21
w17 Mon 2023-04-24 13:00 - 15:00 in U21

Results of Pre-Final Tournament

Results of Final Tournament

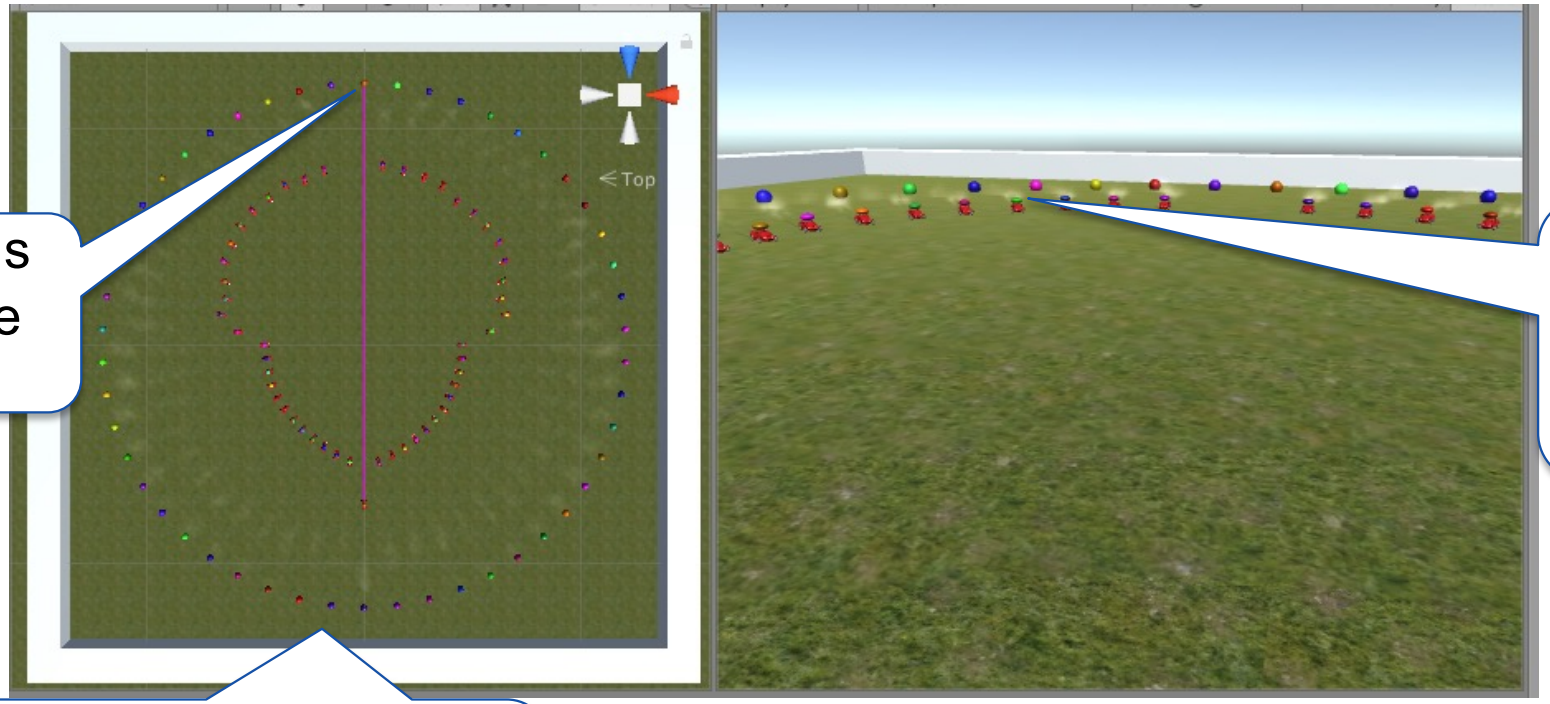
The Problems of Assignment 3

- P1: Multi agent collision avoidance
- P2: Multi agent car/drone-soccer

Details

- You solve both problems with both **cars and drones**
- The drones
  - move like dynamic points (as before)
  - look like blobs/pills (to make pushing balls more predictable)
- The cars
  - same as before, with team color blob above

Problem 1b: Multi agent collision avoidance (autonomous cars in a city)



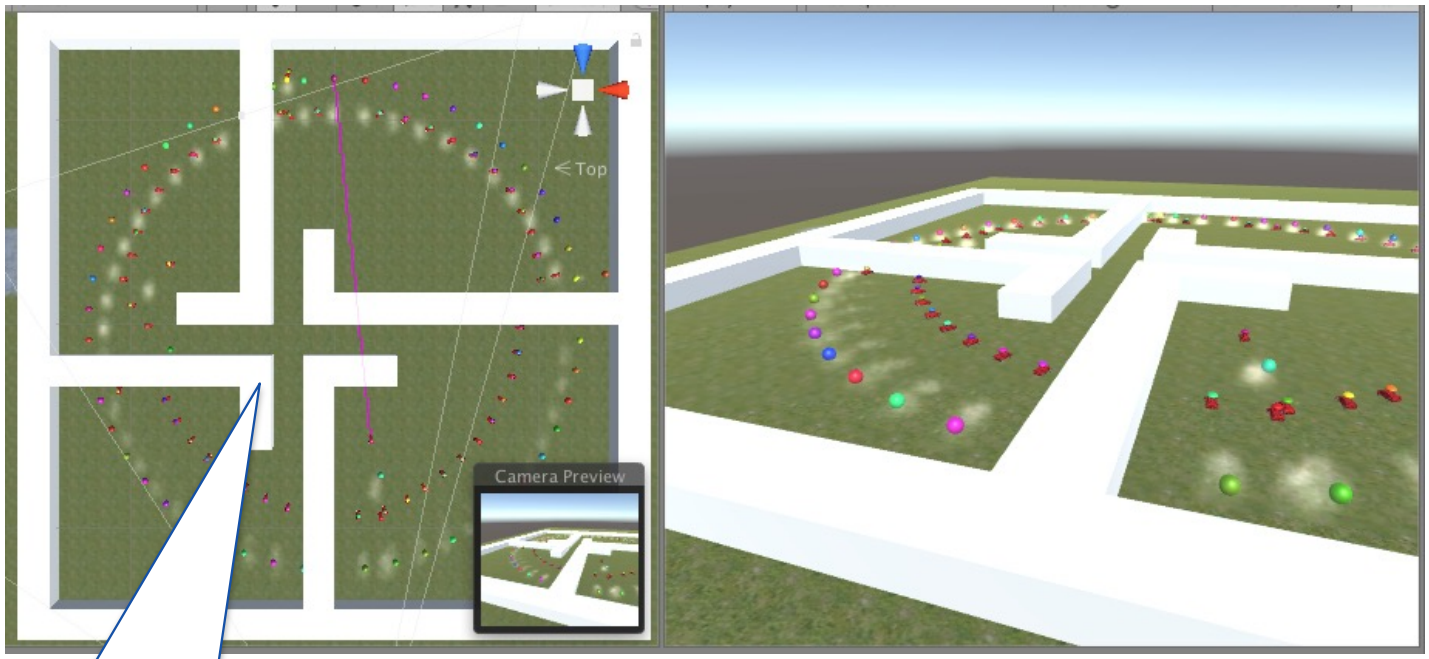
Destinations on opposite sides

Vehicles color coded by M&Ms

Do not identify circular pattern (and apply carousel solution) Same code for all settings

terrainBB.json

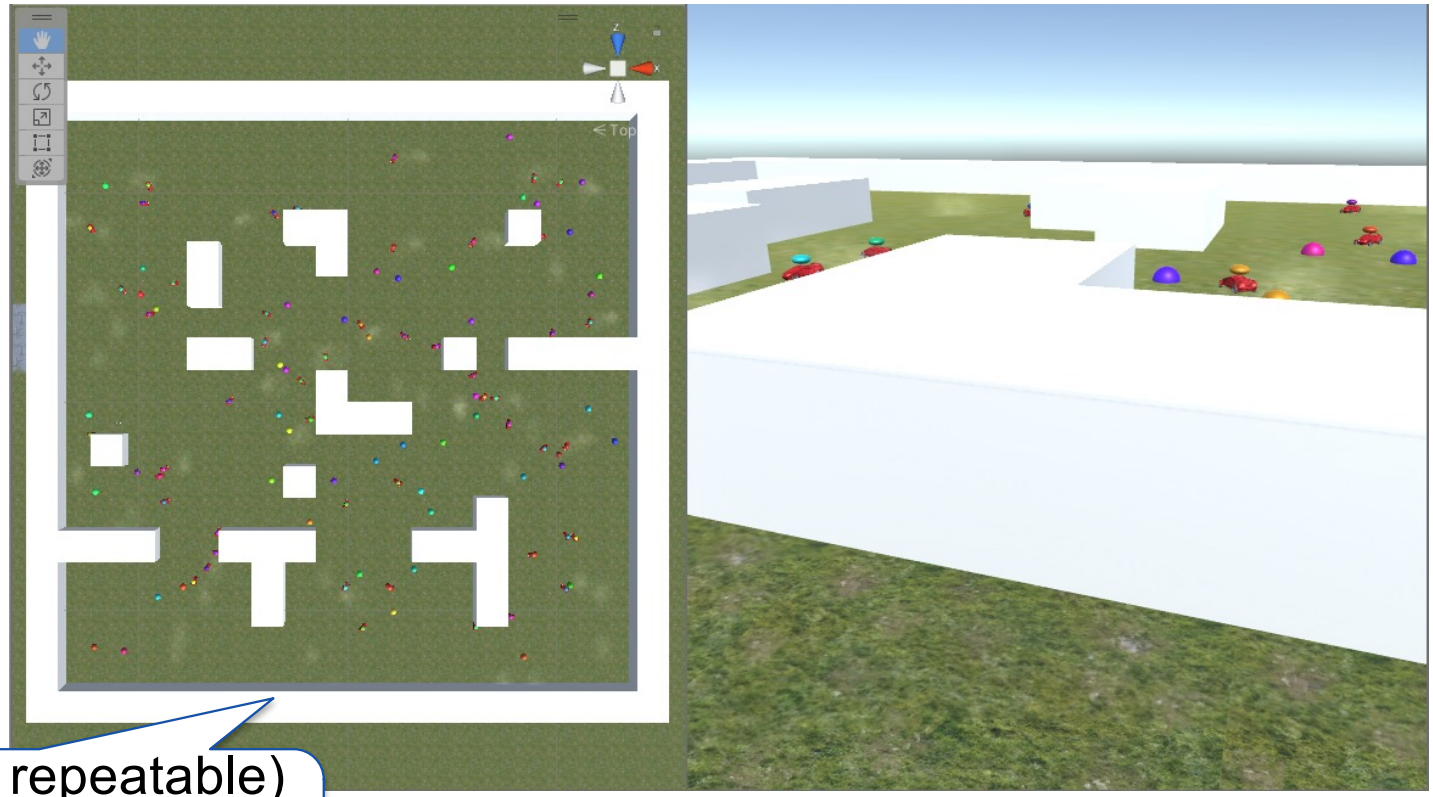
Problem 1a: Multi agent collision avoidance (autonomous cars in a city)



An intersection everyone has to pass

terrainAA.json

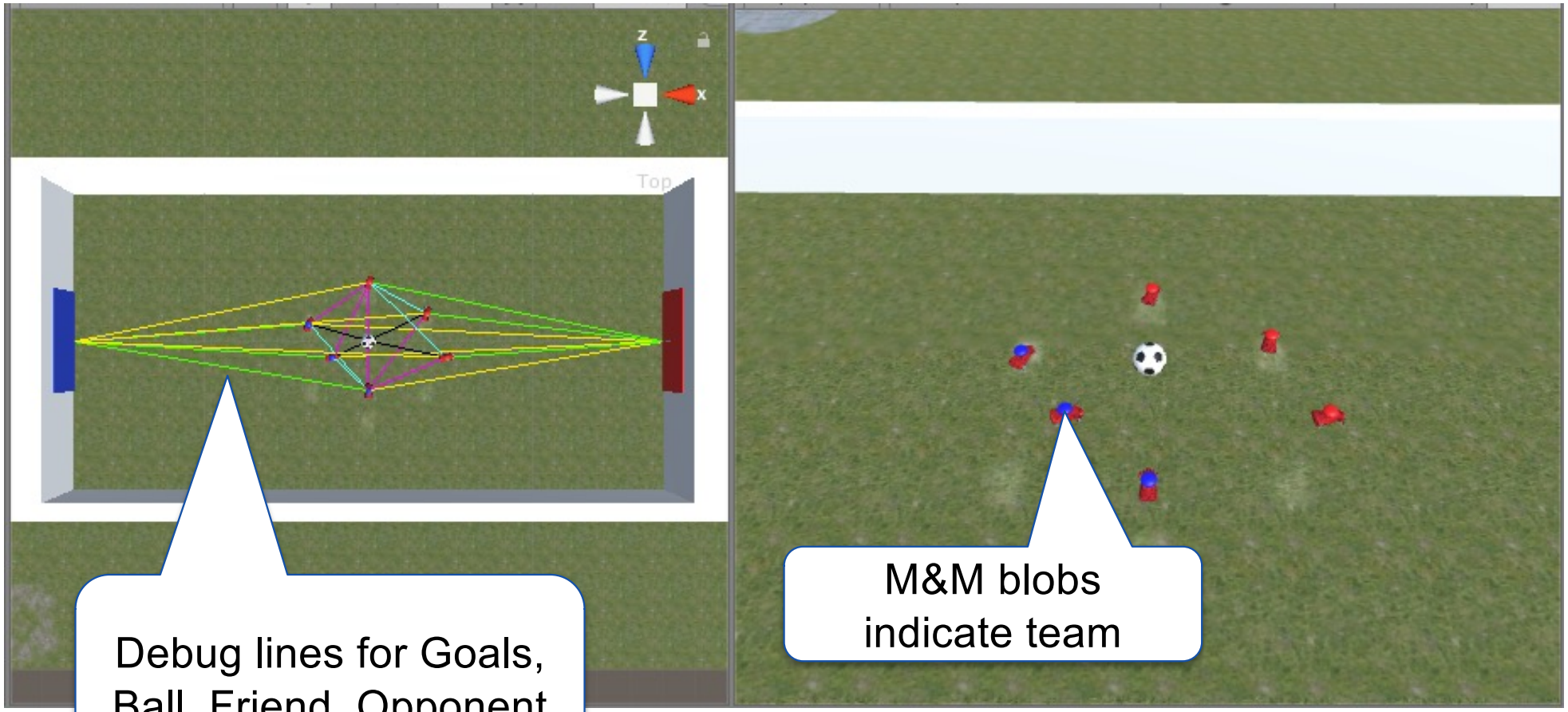
Problem 1c: Multi agent collision avoidance (autonomous cars in a city)



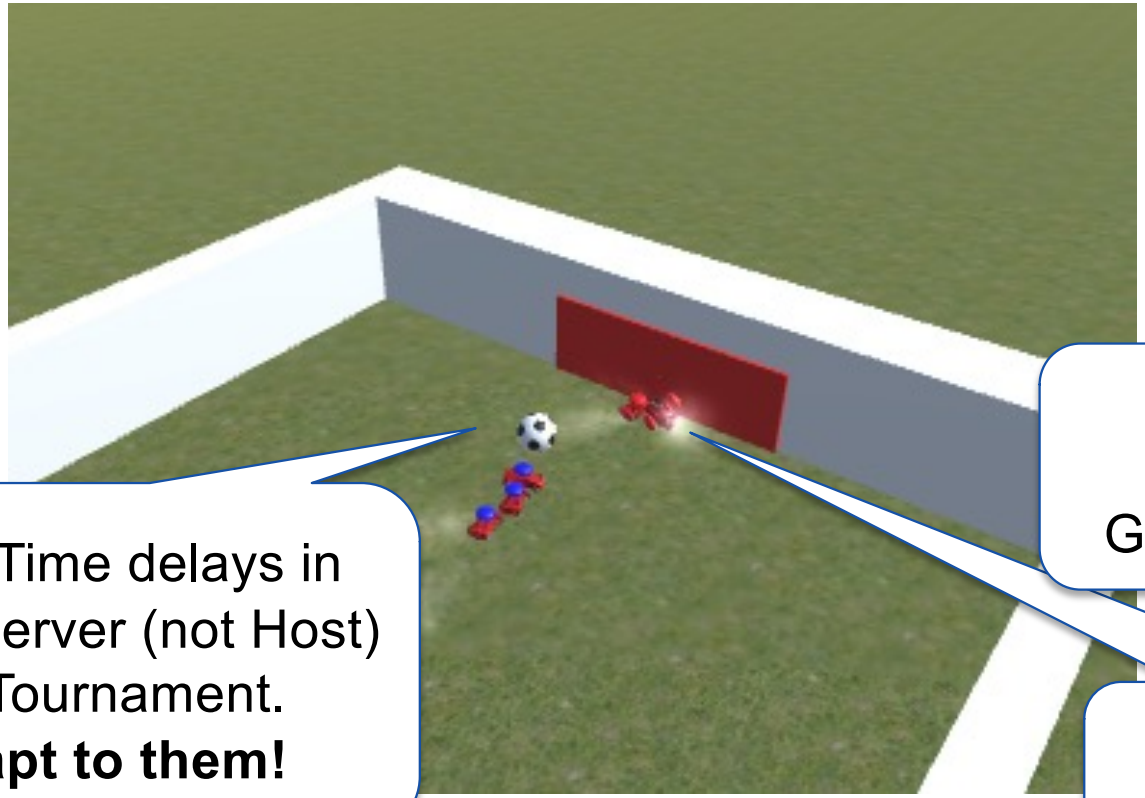
Random (but repeatable) start and goal configurations

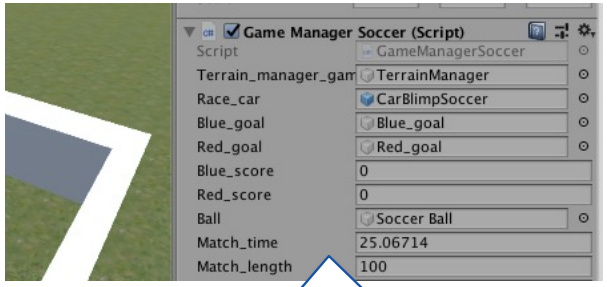
terrainCC.json

Problem 2: Multi-agent car soccer



Problem 2: Multi-agent car soccer





Score keeping and match clock in GameManagerSoccer

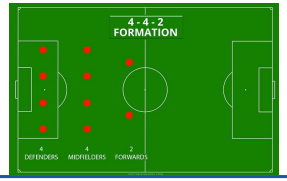
Note: Time delays in Client/Server (not Host) for Tournament. **Adapt to them!**

Default strategies:
-Red: Defend goal
-Blue: Chase ball

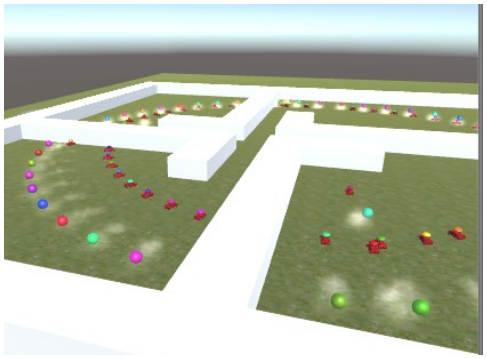
What capabilities are needed?

Soccer AI needs to

- Dribble the ball
- Shoot the ball
- Pass the ball
- Intercept a ball
- Intercept an opponent
- Move to position
  - to defend
  - to get pass
  - to have goal opportunity
  - “to stay in formation”



What capabilities are needed?



Soccer AI needs to

- Dribble the ball
- Shoot the ball
- Pass the ball
- Intercept a ball
- Intercept an opponent
- Move to position
  - to defend
  - to get pass
  - to have goal opportunity
  - “to stay in formation”

Autonomous Car needs to

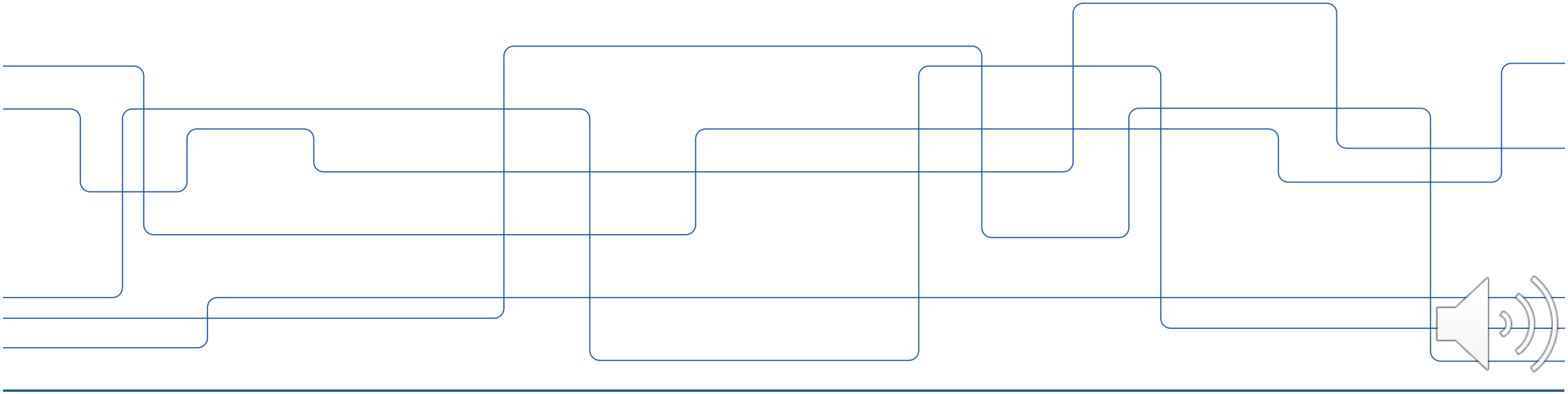
- Get to goal
- Avoid collisions (handle collisions)
- Handle roads
  - Follow car in front
- Handle intersections
  - Avoid car from the side
  - Know when to wait or not

Both cases need to

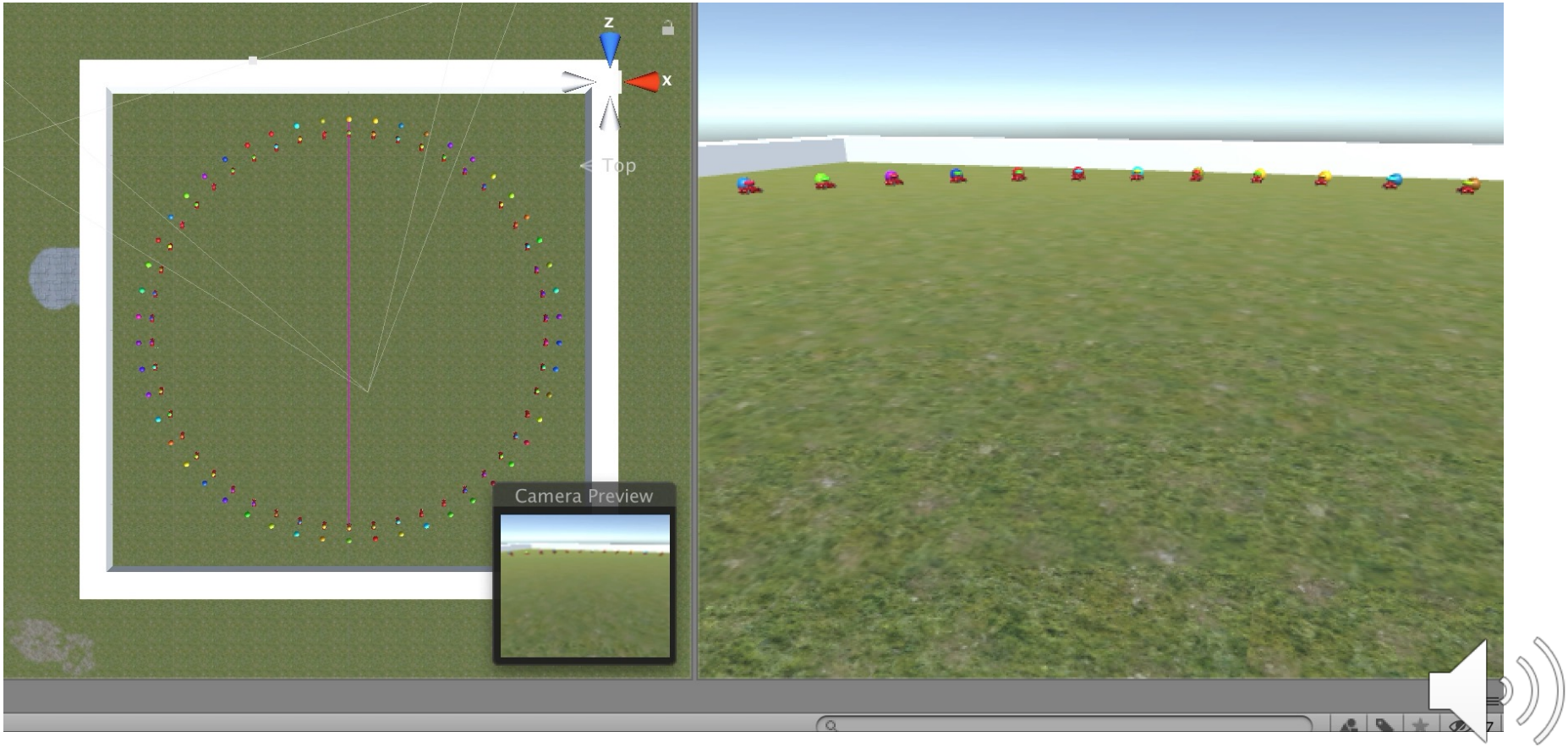
- avoid and Intercept moving obstacles
- do structured switching between controllers/tasks

KTH ROYAL INSTITUTE
OF TECHNOLOGY

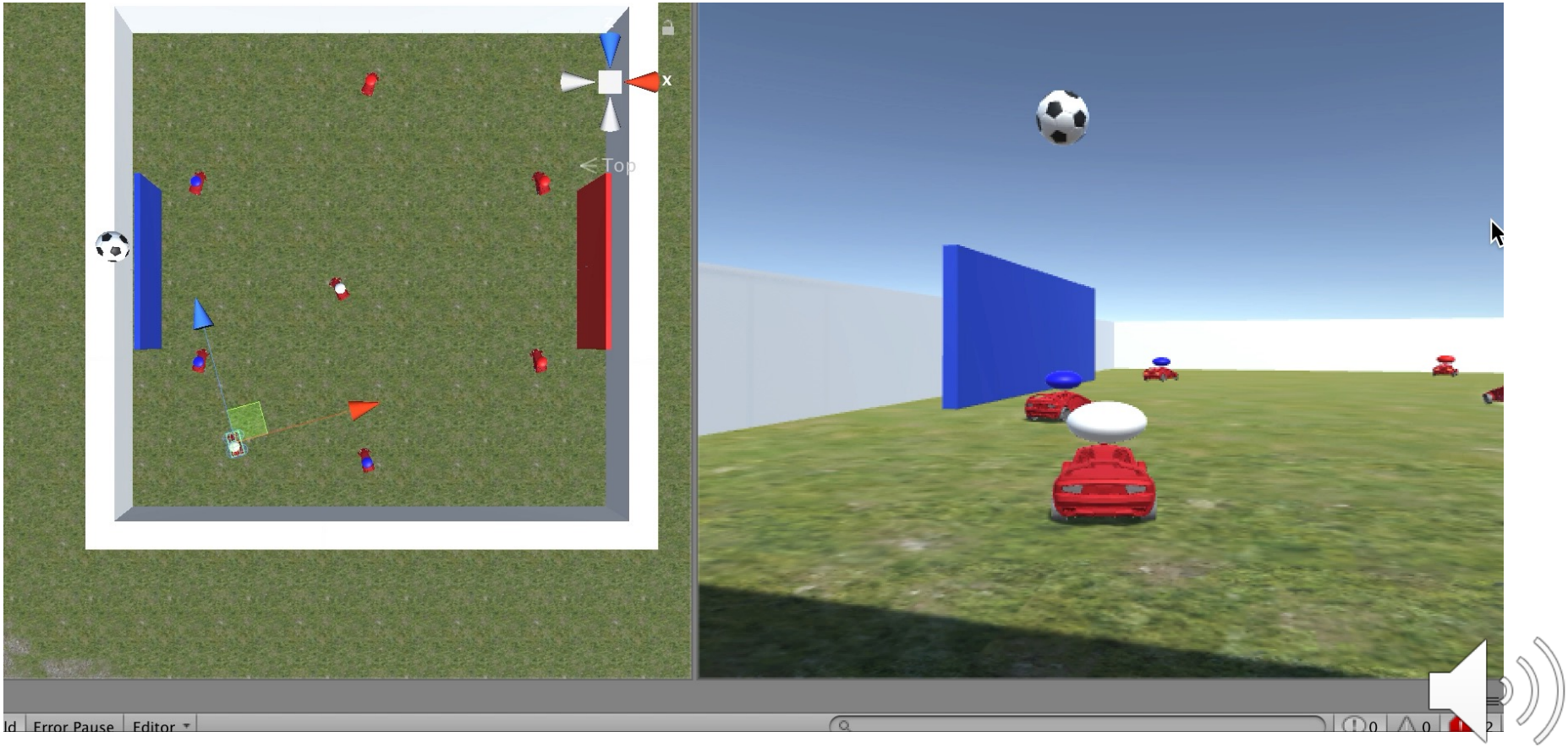
Avoiding and Intercepting Moving Obstacles



Why do we need to avoid and intercept objects?

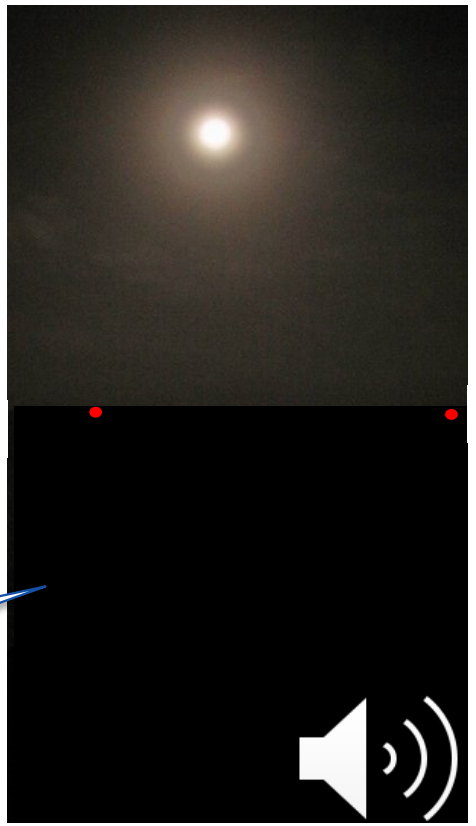


Why do we need to avoid and intercept objects?



Inspiration

Early Ideas on collision avoidance



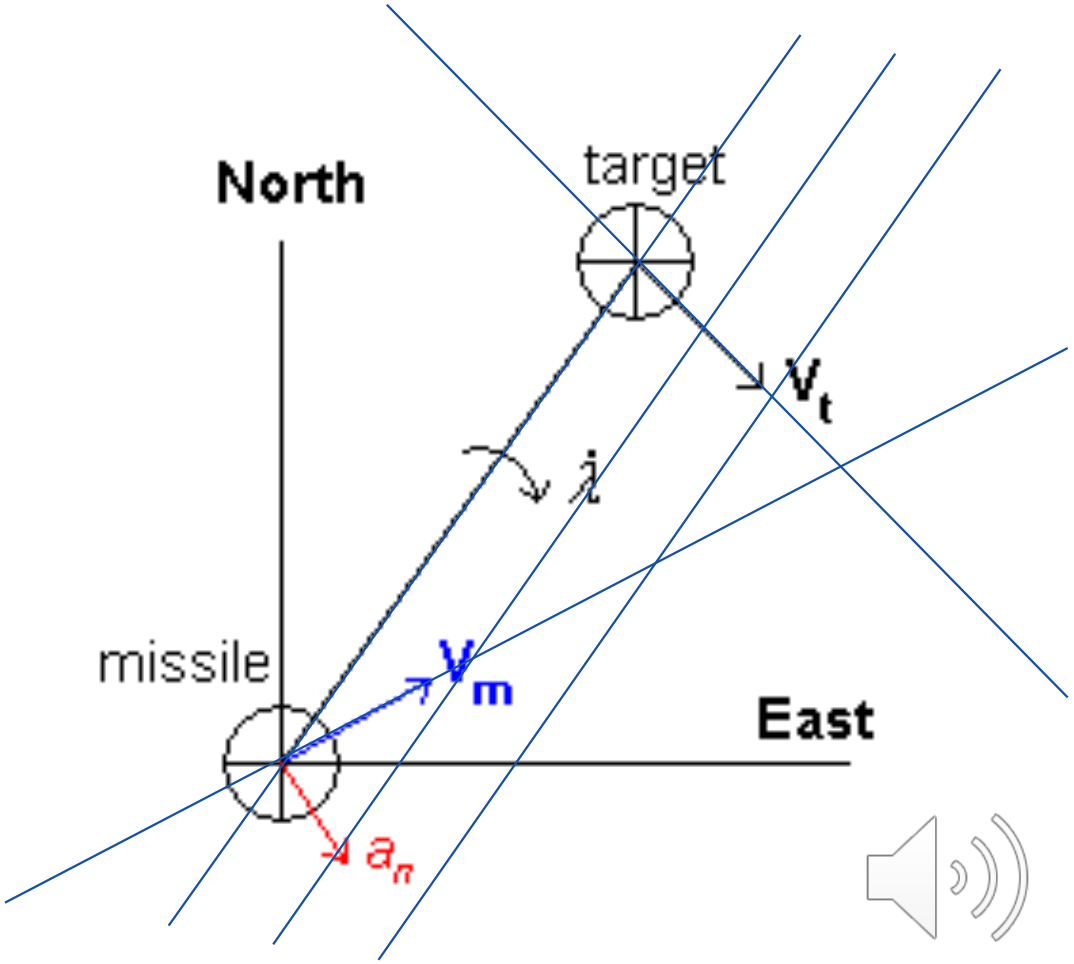
Constant bearing
leads to collisions!

Proportional Navigation (for missiles)

- Observation:
  - Constant bearing leads to collision

- Turn based on bearing change

$$a_n = N\dot{\lambda}V$$

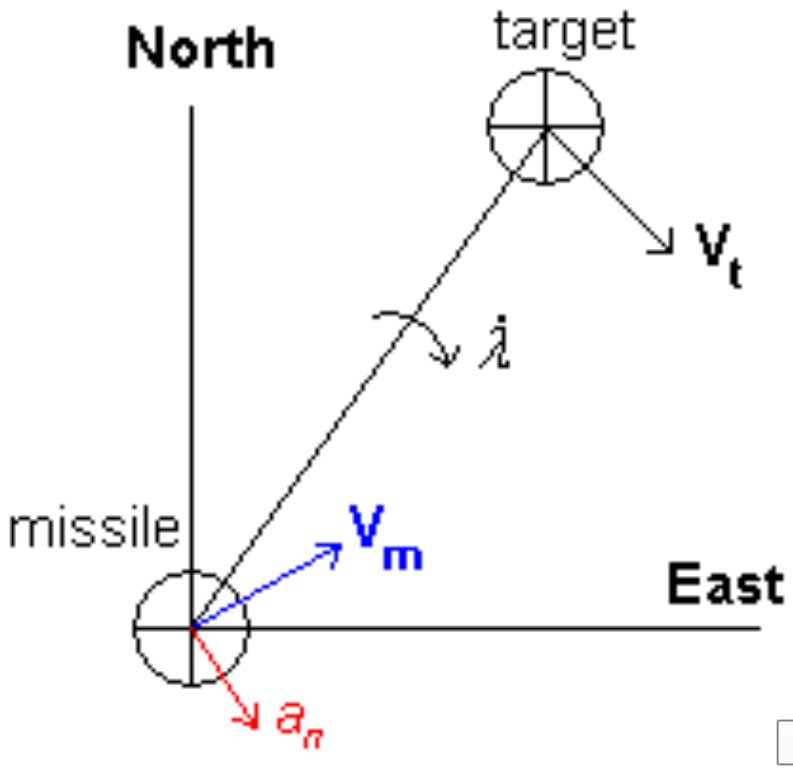


Proportional Navigation (for missiles)

- Observation:
  - Constant bearing leads to collision

- Turn based on bearing change

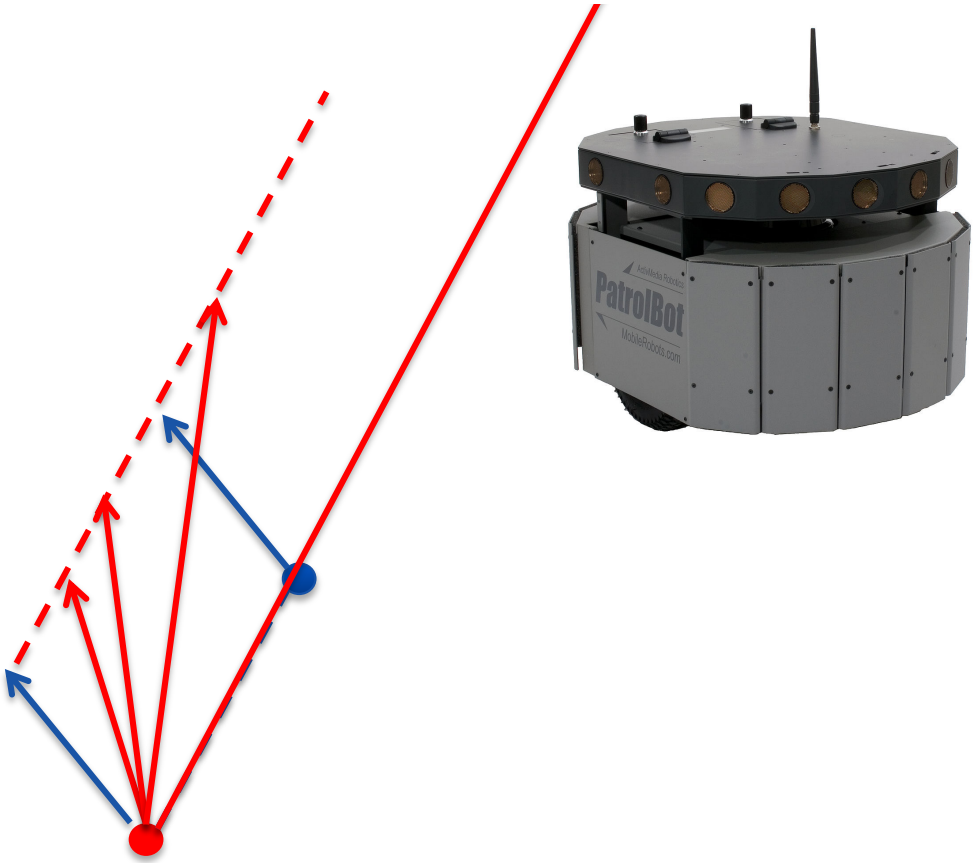
$$a_n = N\dot{\lambda}V$$



What if we have pos & vel of target?

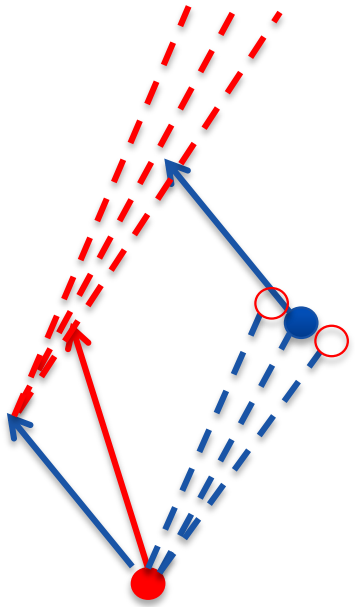
Velocity Obstacles

- What velocities will result in a collision?
- All velocities on the red line!



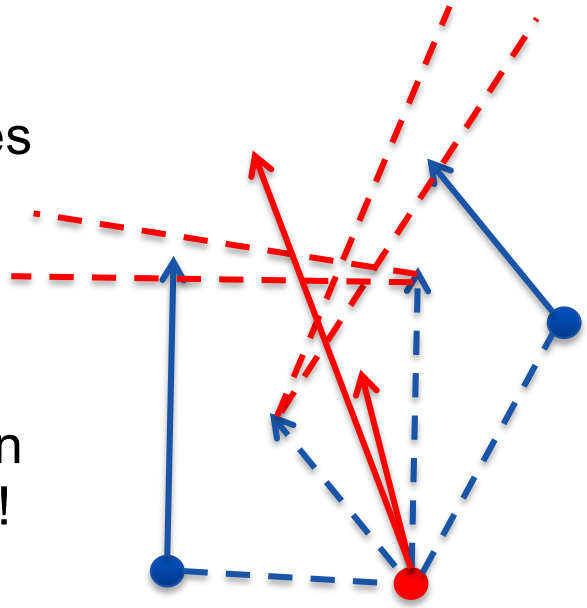
Velocity Obstacles (nonzero size)

- What velocities will result in a collision?
- All velocities in the red cone!



Velocity Obstacles (multiple obstacles)

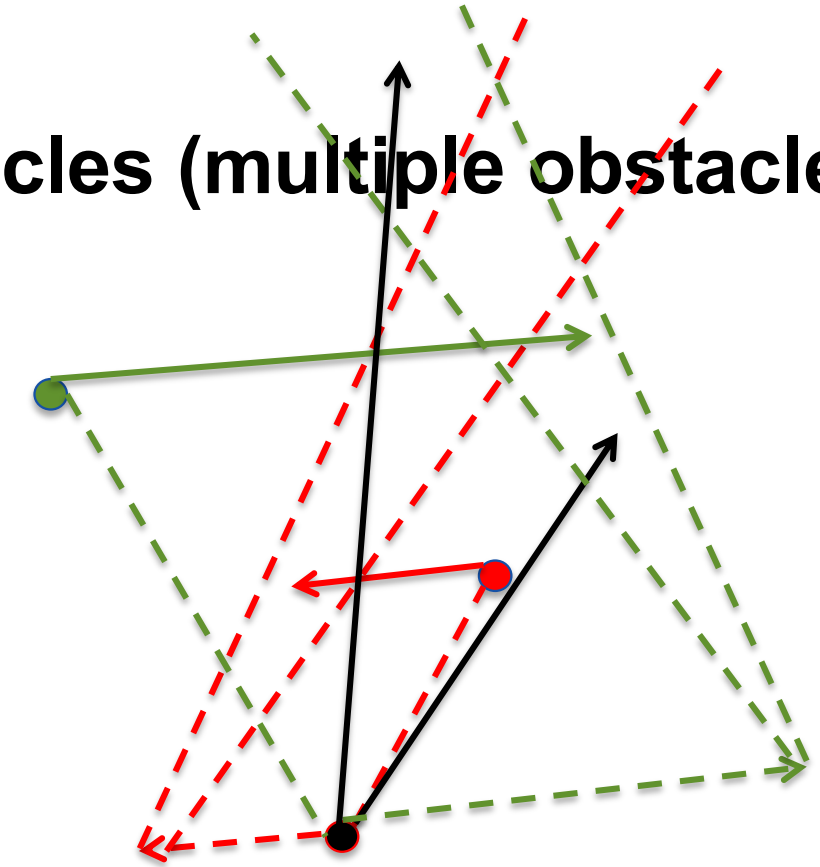
- What velocities will result in a collision?
- All velocities in the red cones!
- Find best velocity vector outside red cones!



Note: this does not take time to collision into account...

Velocity Obstacles (multiple obstacles v2)

- If you want to pass green?
- Note: analysis assumes constant velocity…

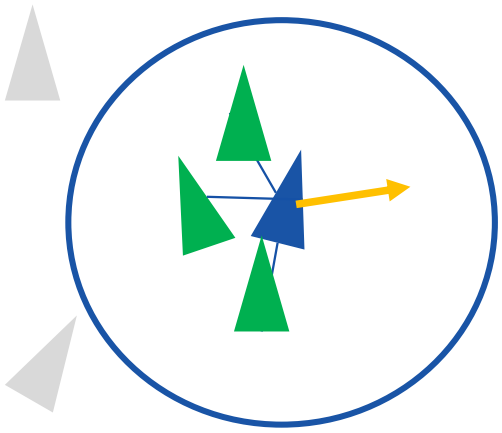


Make passing kick to achieve desired average velocity…

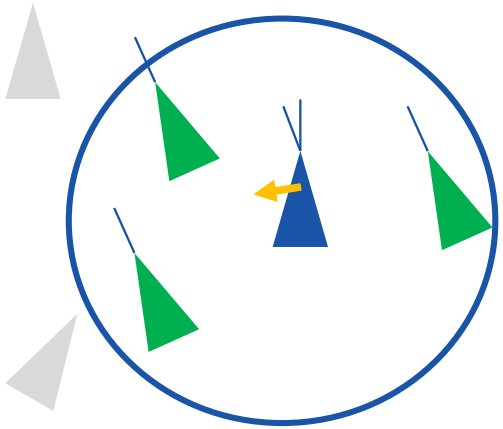
Reynolds Boids

3 simple rules to create flocking behavior

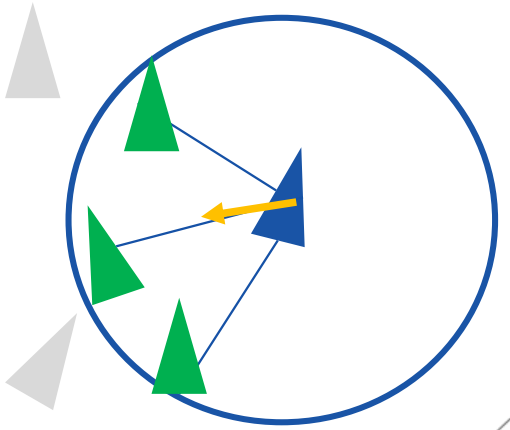
Separation



Alignment

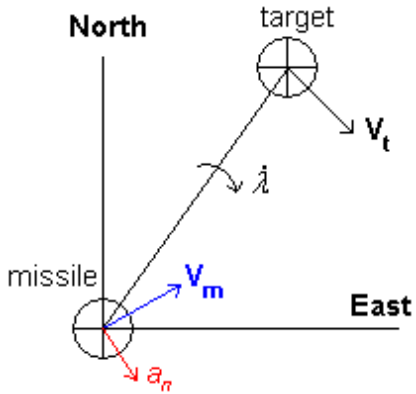


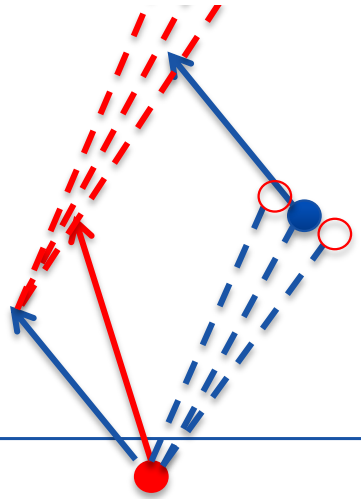
Cohesion

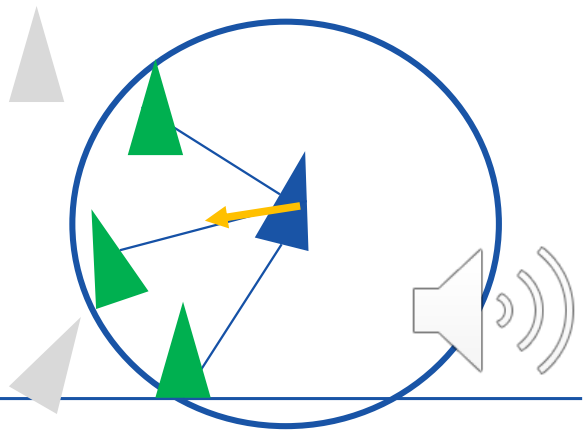


(nice videos on youtube)

Inspiration





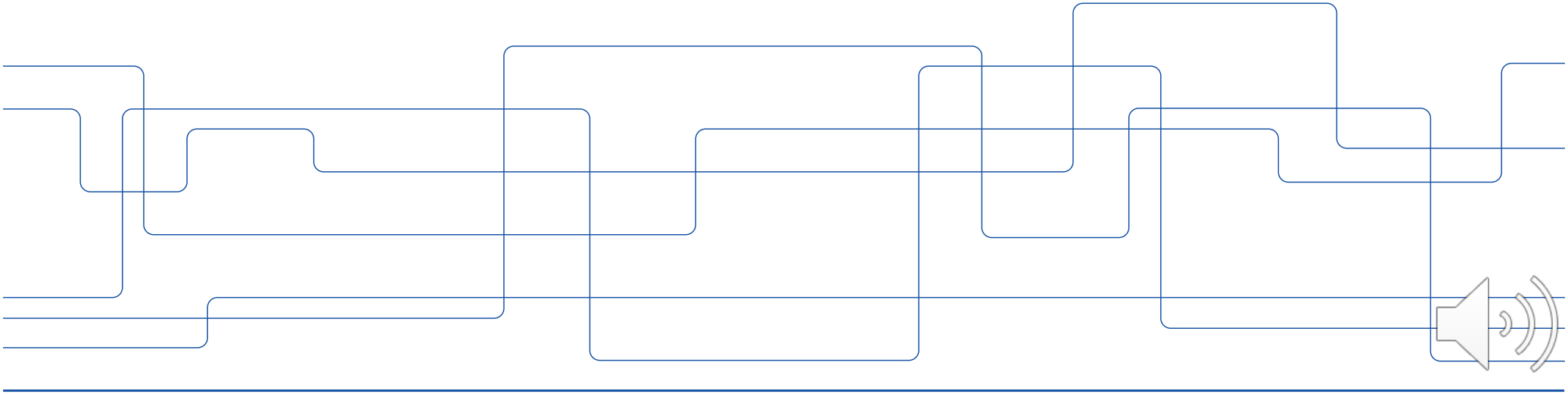


References

- Tan, R., & Kumar, M. (2013). Proportional navigation (PN) based tracking of ground targets by quadrotor UAVs. In *ASME 2013 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers Digital Collection.
- Reynolds, C. W. (1987, August). Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*(pp. 25-34).
- Van den Berg, J., Lin, M., & Manocha, D. (2008, May). Reciprocal velocity obstacles for real-time multi-agent navigation. In *2008 IEEE International Conference on Robotics and Automation* (pp. 1928-1935). IEEE.
- Snape, J., Van Den Berg, J., Guy, S. J., & Manocha, D. (2011). The hybrid reciprocal velocity obstacle. *IEEE Transactions on Robotics*, *27*(4), 696-706.

KTH ROYAL INSTITUTE
OF TECHNOLOGY

Structured switching between controllers/tasks



Many roles/tasks on a soccer field

Looking for pass

Taking the ball

Getting open in good pos

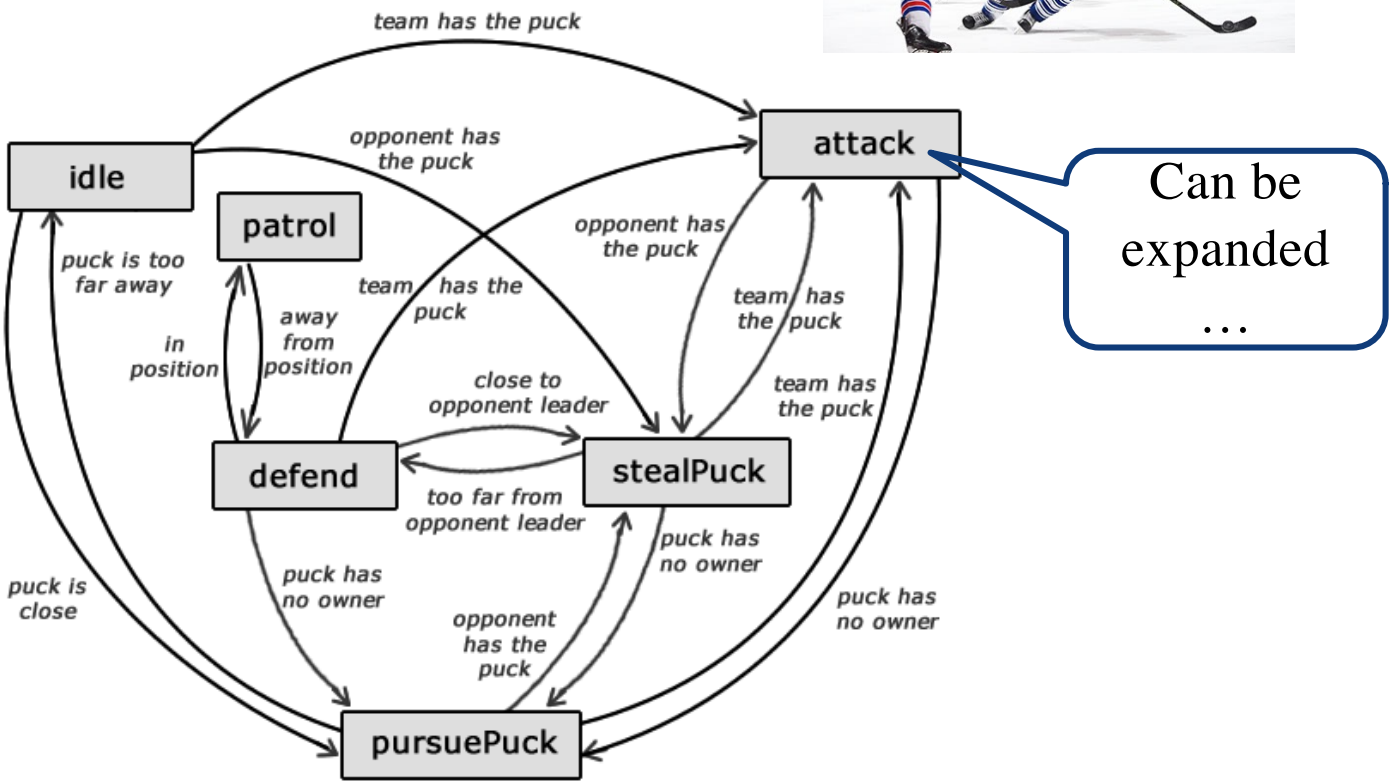
Moving the ball

Getting open

Covering attacker

Getting open in good pos

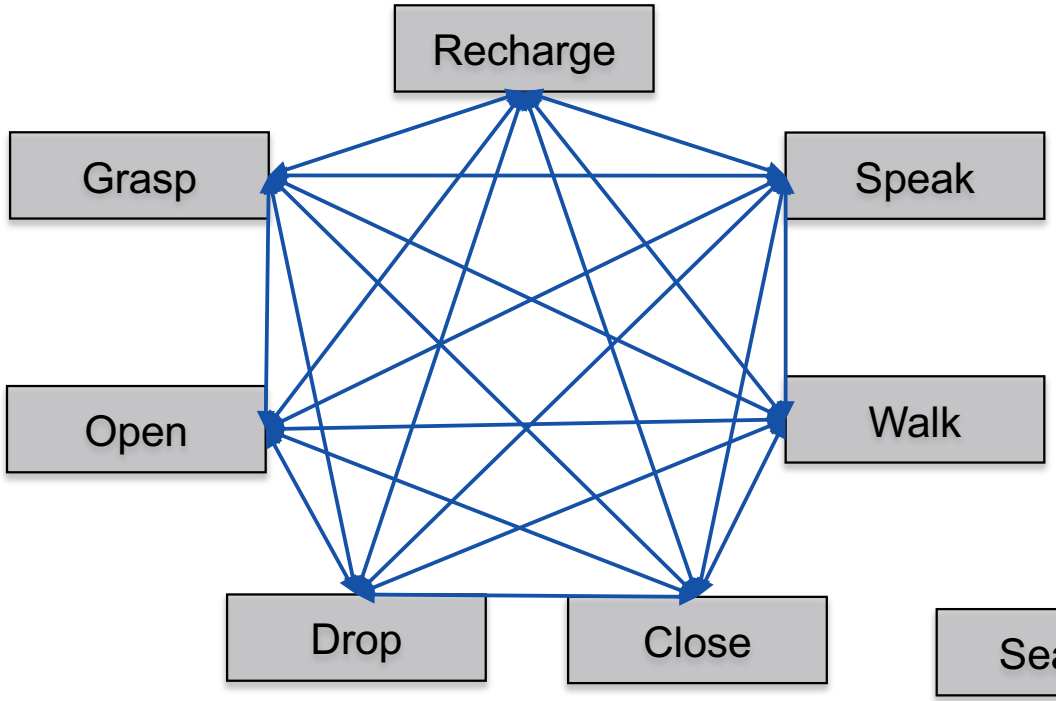
Can you spot the Bug?



What to do next?

Wait	Drive
Pass	Shoot
Prep. pass	Prep. shot
Dribble	Chase ball
Cover opponent	Attack opponent
Cover goal	Get passable
…	Speak
Grasp	Drop
Run	Throw

What to do next?

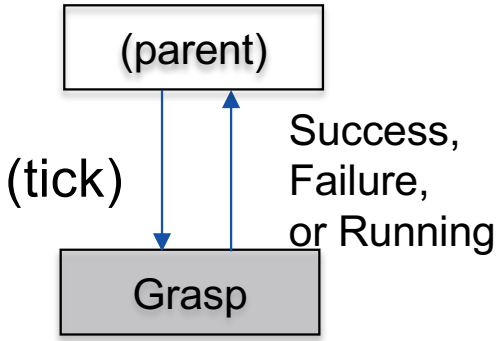


Finite State Machine

Lock
Unlock
Pull
Push
Kick
Search
Speak
Clean
Record

Each action needs to know “What to do next”…

What to do next?



Behavior Tree

Each action needs to know
“Did I Succeed or Fail?”

Grasp | Recharge
Drop | Lock
Walk | Unlock
Open | Pull
Close | Push
Search | Kick
Clean | Speak
Speak | Record
Run | Th…w

Ancestors decide “What do to next?”

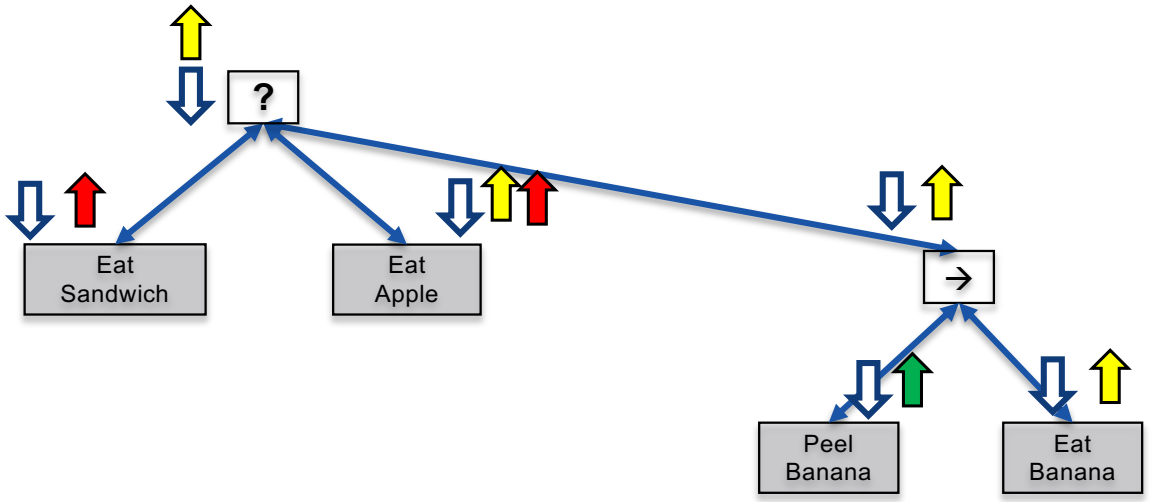
Two Fundamental Compositions of Actions

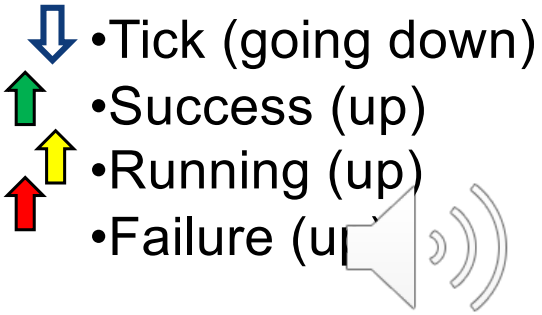
- **Fallback** (?)(or)
  - (Eat Sandwitch ? Eat Apple)

IF Failure then Tick Next
else Return “same as child”

- **Sequence** (→)(and)
  - (Peel Banana → Eat Banana)

IF Success then Tick Next
else Return “same as child”



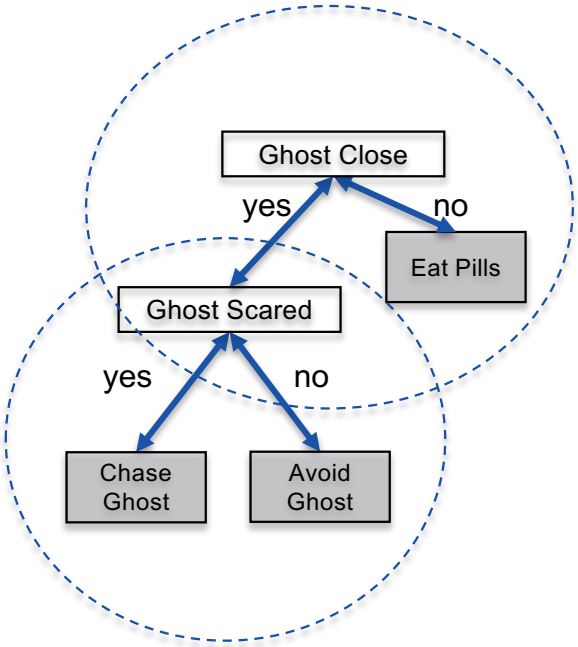


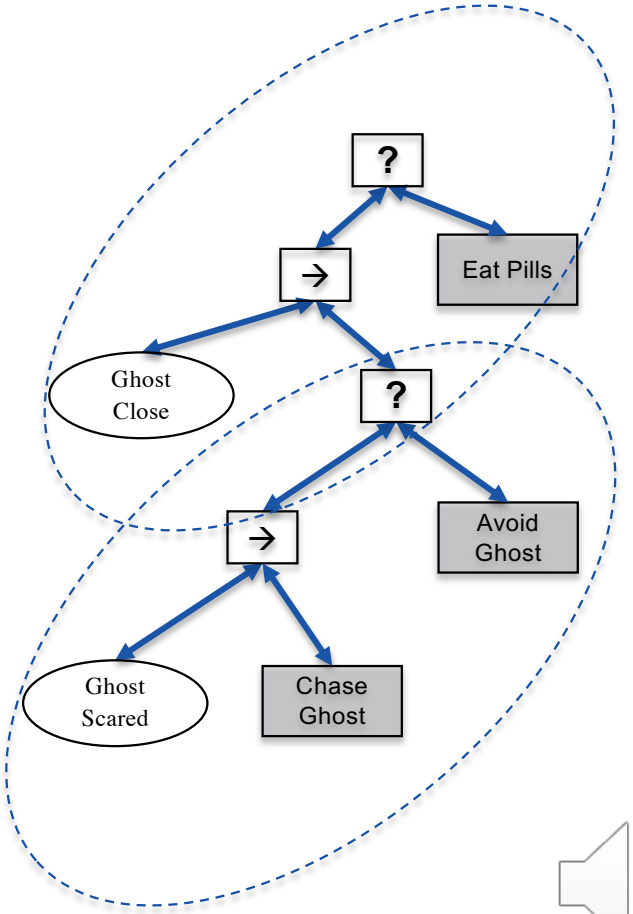
Using Decision Tree design

- Behavior Trees (BTs) **Generalize** Decision Trees (DT)
  - every DT can be written in the form of a BT
- If a DT design is appropriate - We can easily transfer it into a BT
- But, since DTs do not allow return status from Actions/Subtrees we assume that **all "leaves" of the DT always return "Running"**

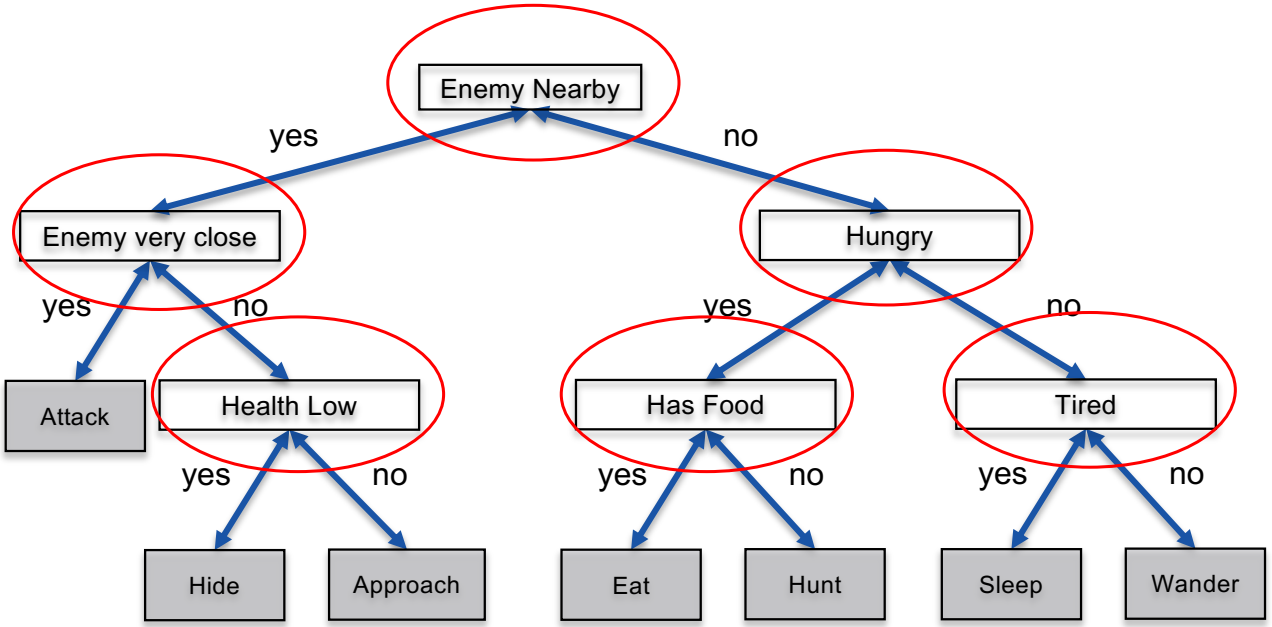
- Why not just use a DT?
  - Lack of return status reduces modularity
  - A DT design might be used as a sub-tree of a larger BT, using other design principles (such as backward chaining)
- Let's see some examples…

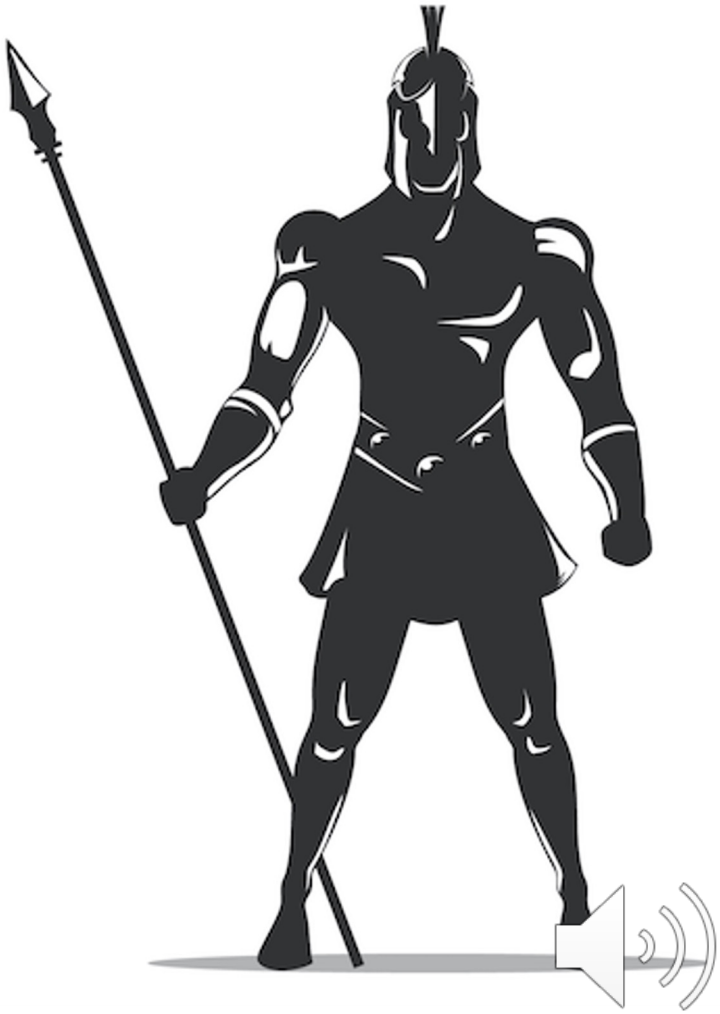
Packman DT design



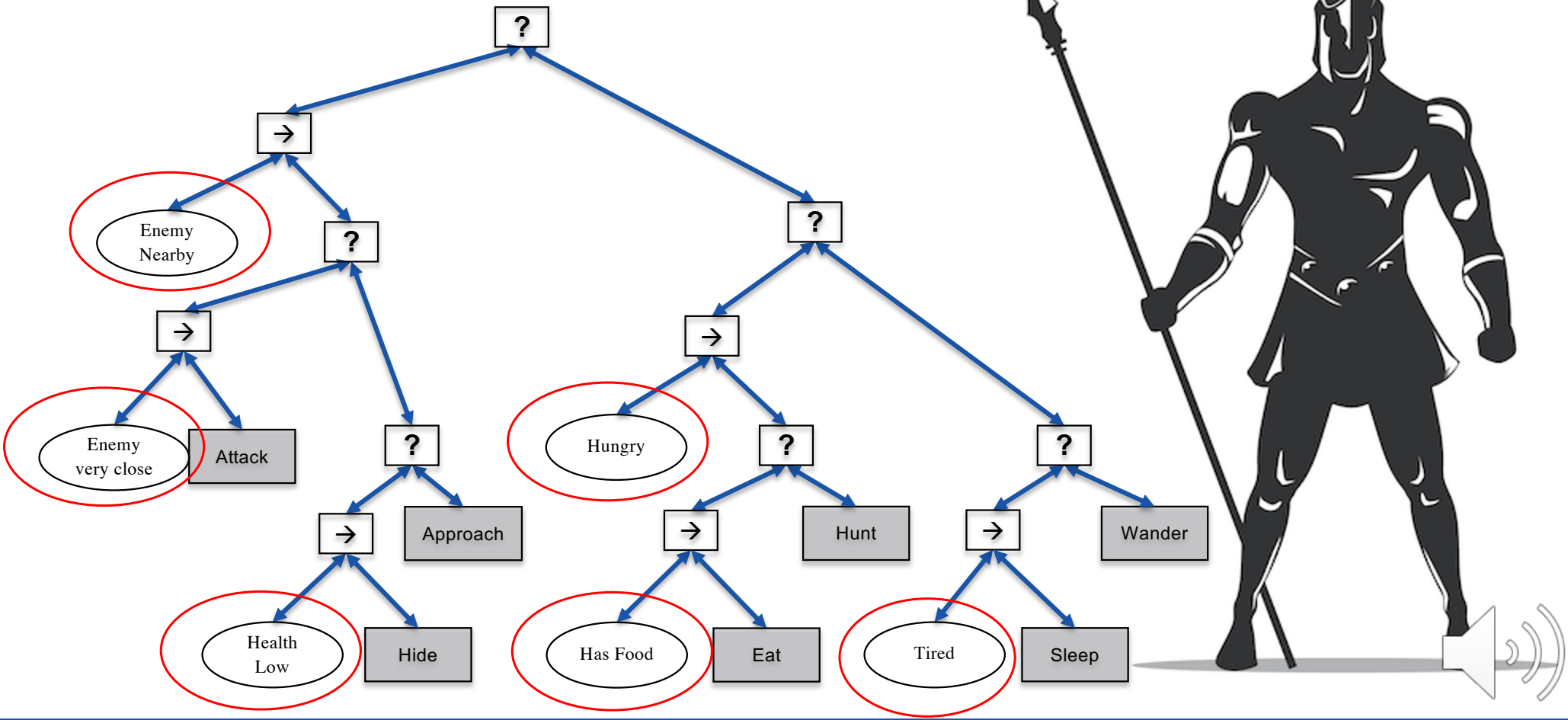


Warrior DT design

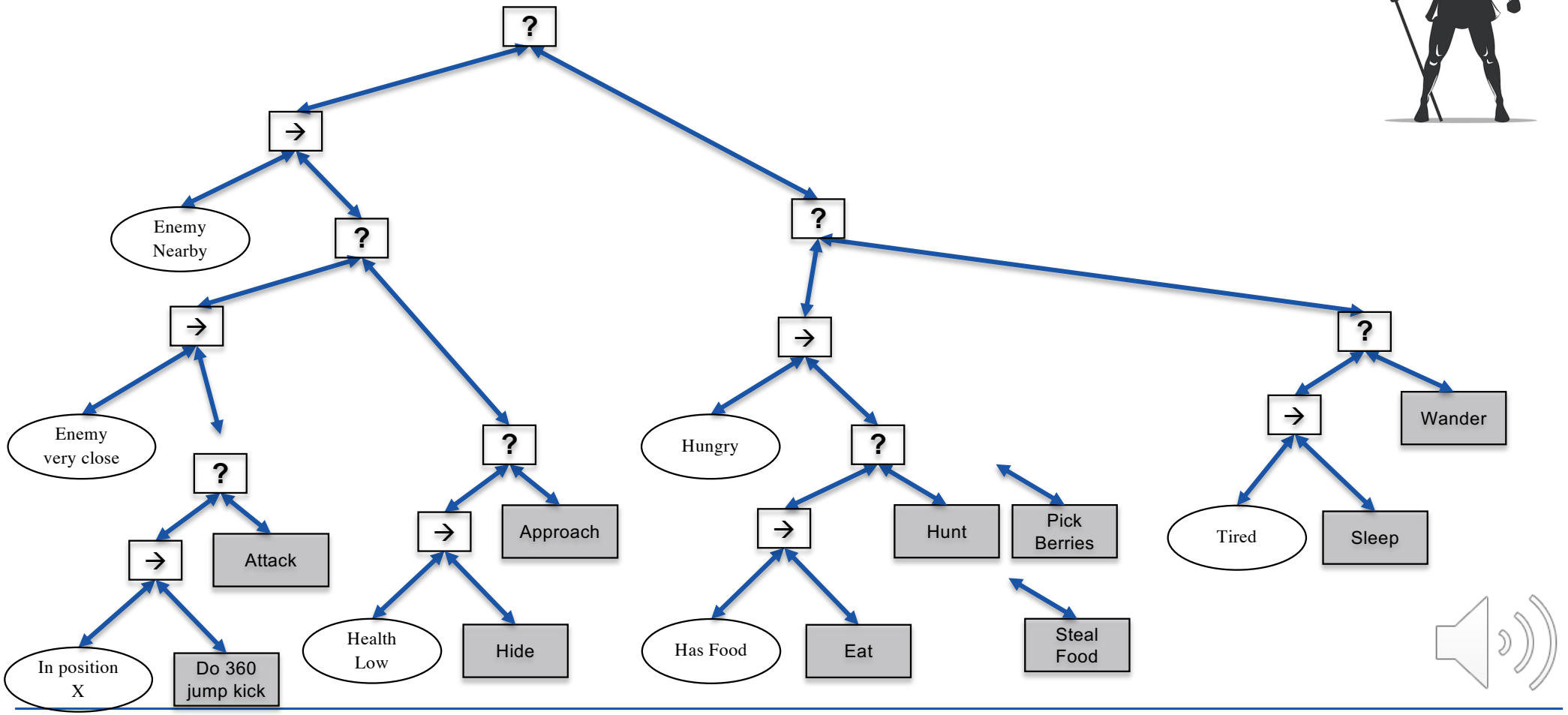




Warrior DT design of BT



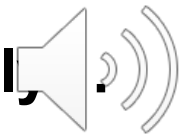
Warrior DT design of BT



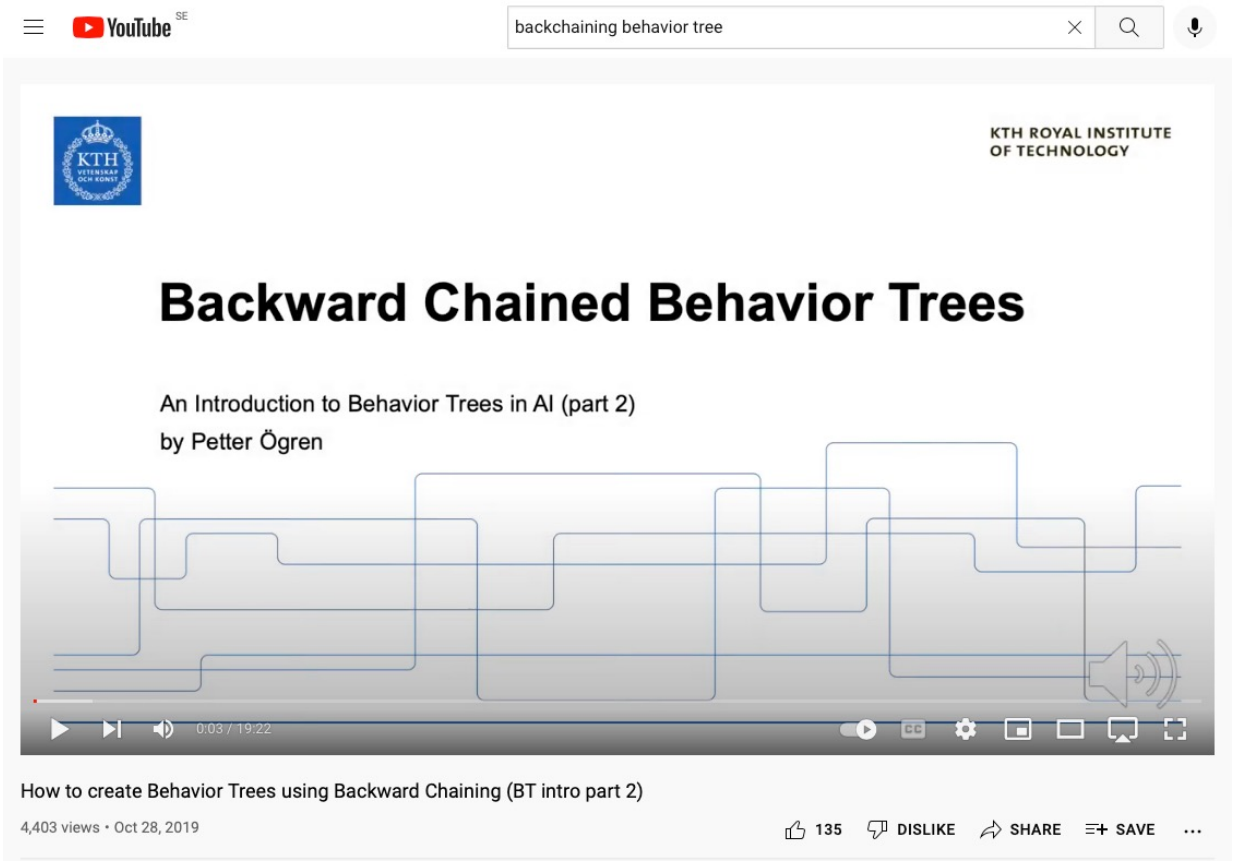
What is Backward Chaining?

- Solving an AI Planning Problem by working **backwards from the goal**
- Example:
  - Goal: *Leave the room*
  - To leave I need to **pass through the door**
  - To pass the door I need to **open the door**
  - To open the door I need to **grasp the handle**
  - To grasp the handle I need to **extend my arm**
  - **Plan:**
    - > ***Extend arm***
    - > ***Grasp handle***
    - > ***Open door***
    - > ***Pass through the door***

BTs can do this reactively!



More details on YouTube if you want…



Assignment 3 summary

- P1: Multi agent collision avoidance
- P2: Multi agent car/drone-soccer

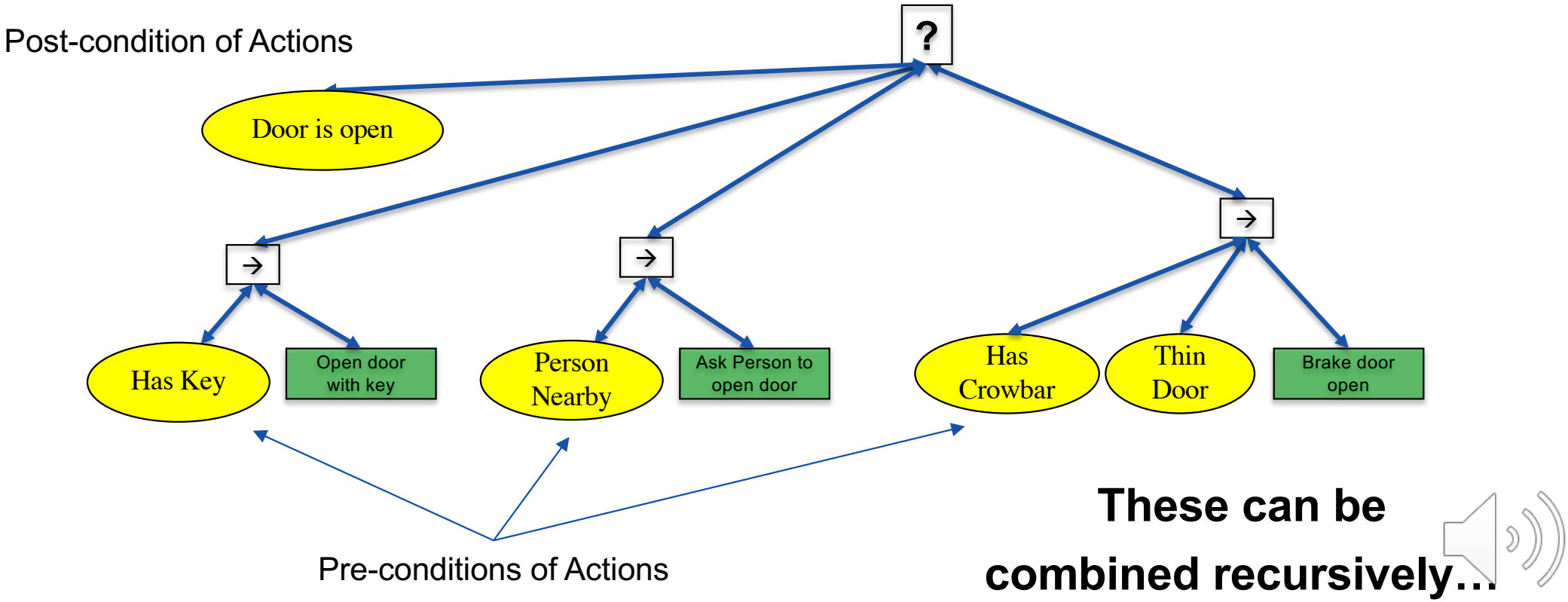
Module 3 (4 weeks, 100h of work)

w13 Mon 2023-03-27 13:00 - 15:00 in U21
w14 Mon 2023-04-03 13:00 - 15:00 in U21
w15 EASTER
w16 Mon 2023-04-17 13:00 - 15:00 in U21
w17 Mon 2023-04-24 13:00 - 15:00 in U21

Results of Pre-Final Tournament

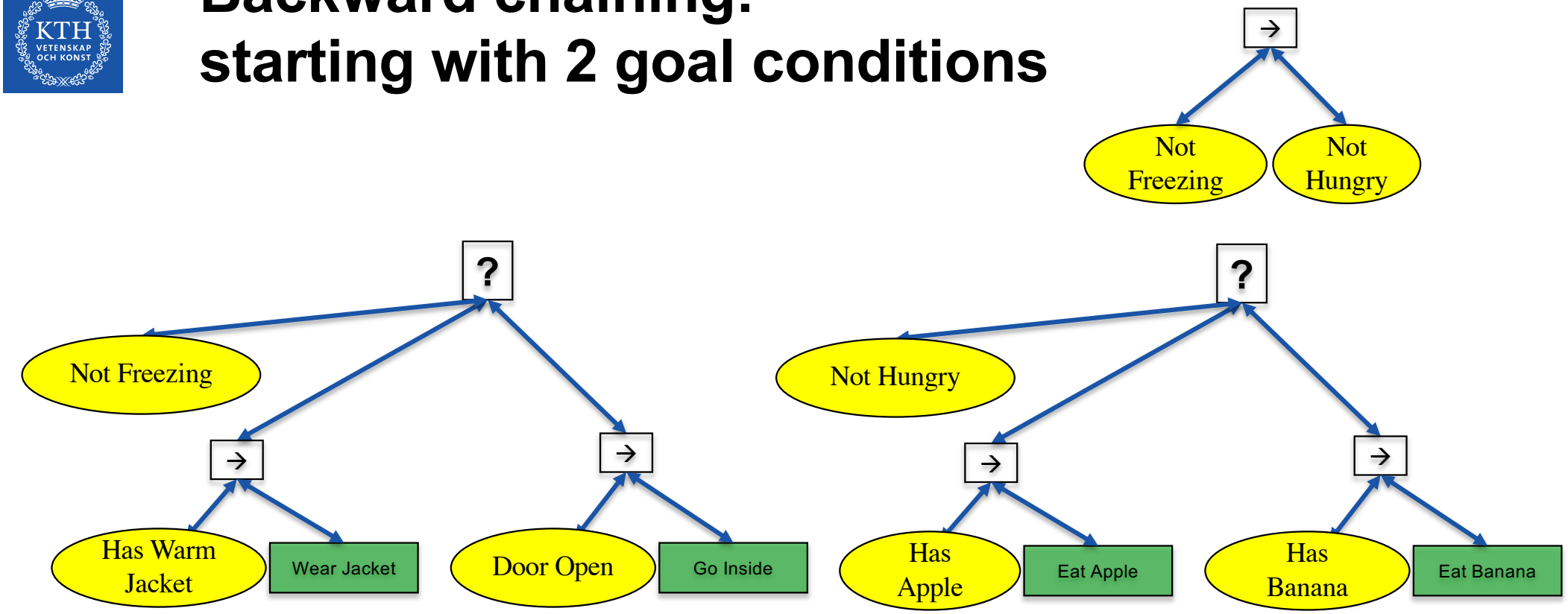
Results of Final Tournament

A BT that achieves a single goal (using feedback)



These can be combined recursively…

Backward chaining: starting with 2 goal conditions

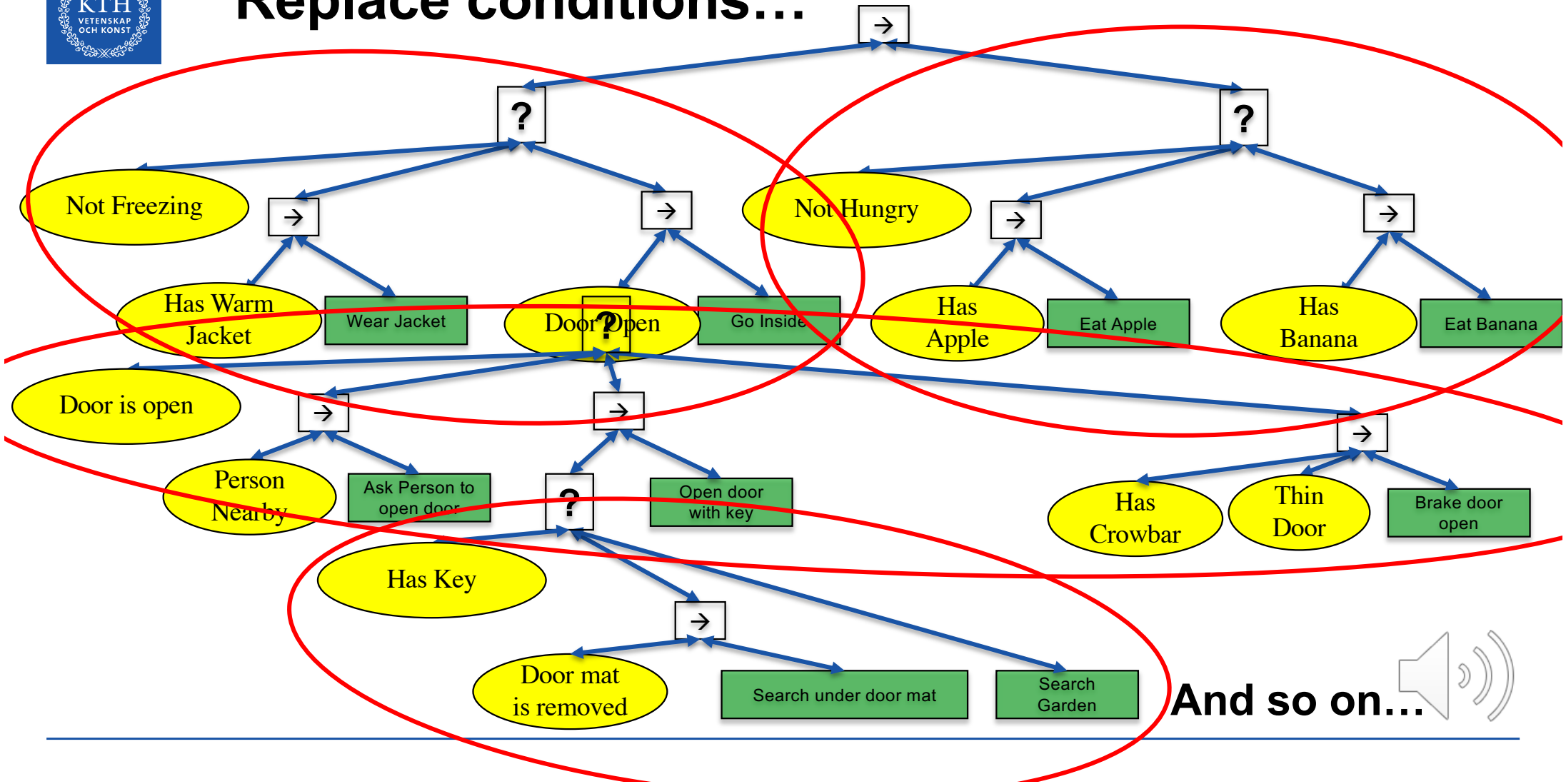


Find BTs that achieve e ch

Replace condition (with Sequence parent) with new subtree

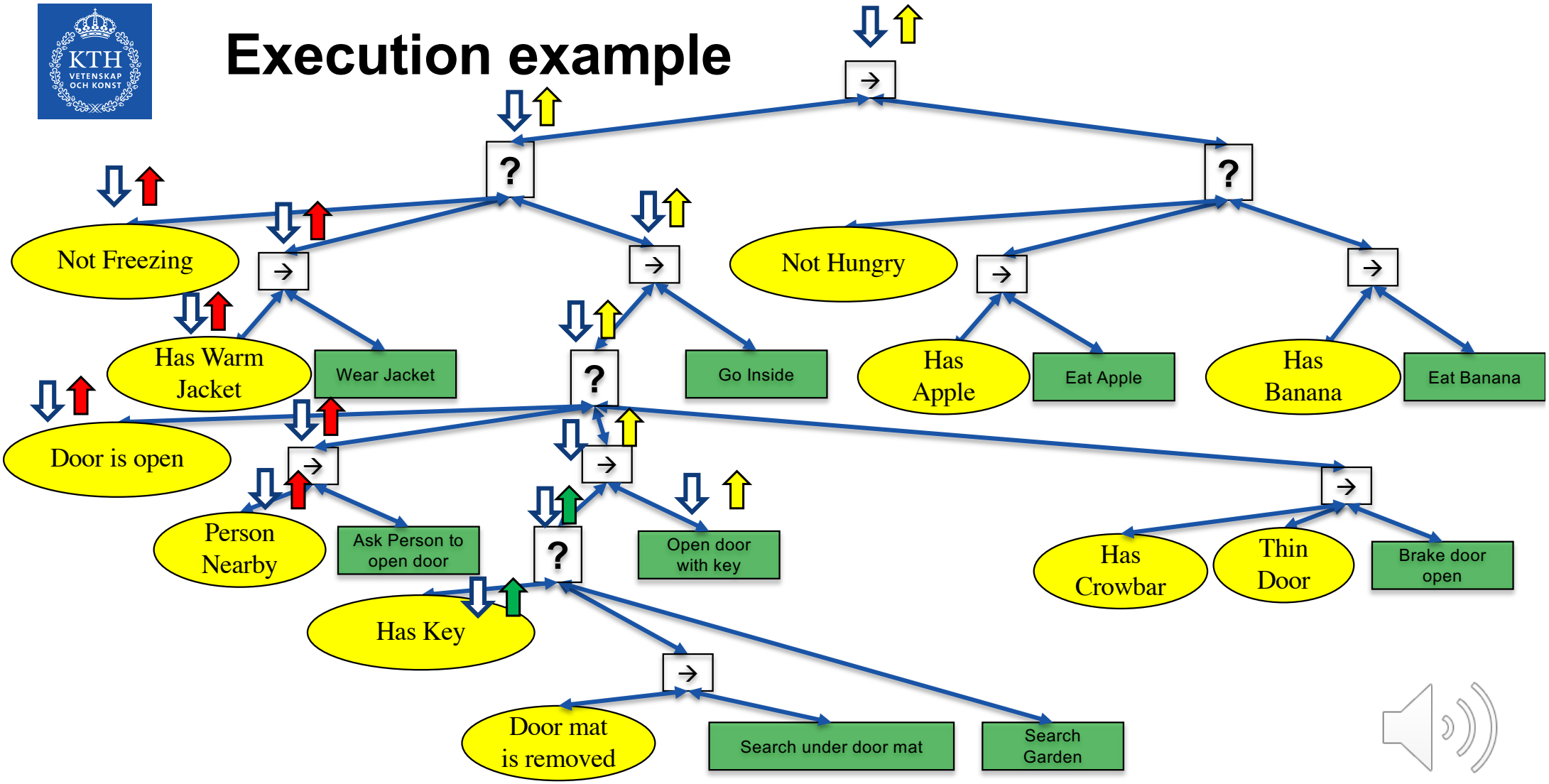
Iterate this…

Replace conditions…

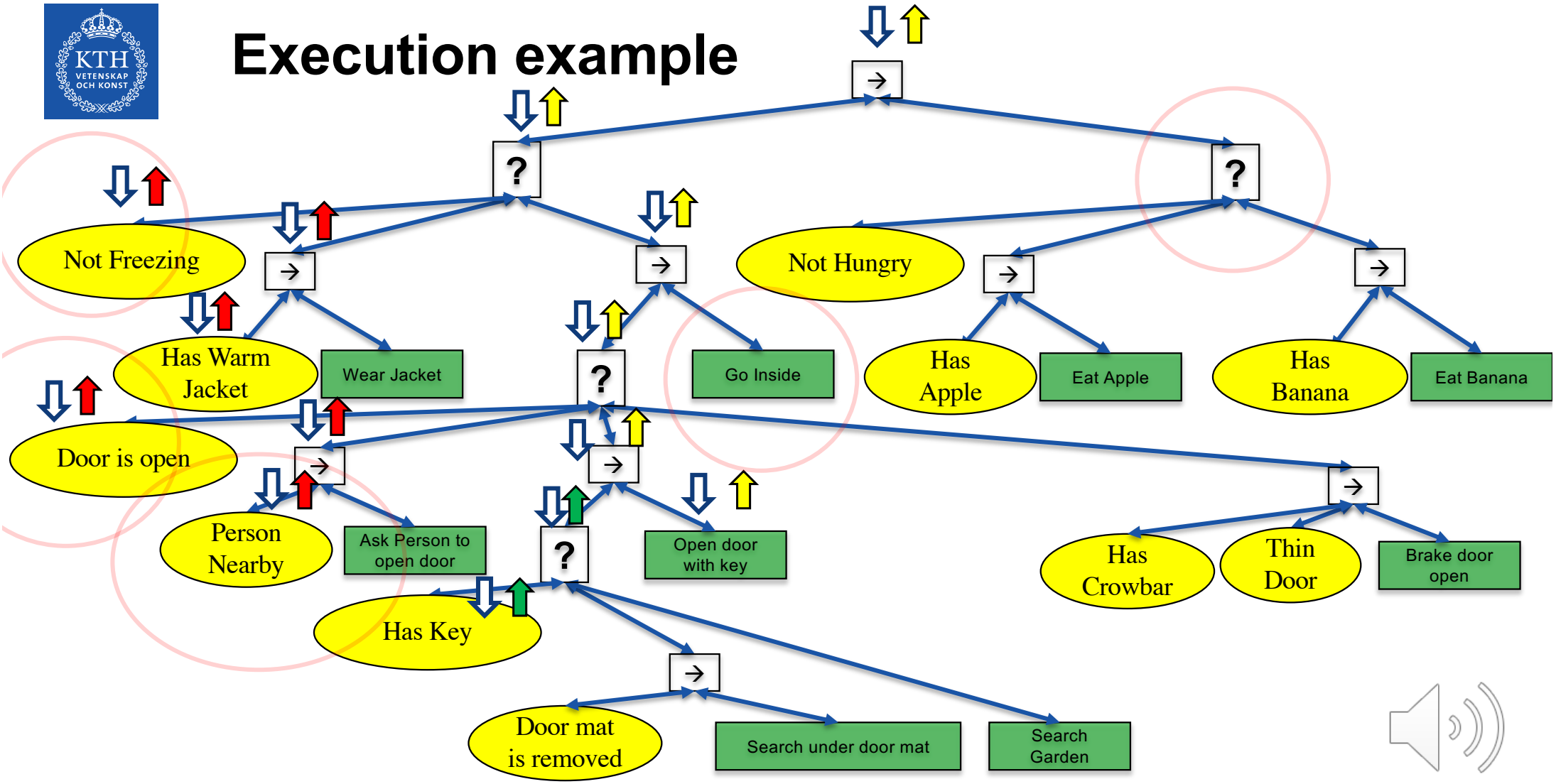


And so on…

Execution example



Execution example



Key idea:
Replace conditions with BT that achieve them!