Game AI

Ken-Yi Lee

Game Programming, Fall 2020 @ National Taiwan University

FeisStu

Game Programming

- Rendering
- Looping and control
- Math
- Animation
- Physics
- Behaviour and navigation (AI)
- Effects
- Networking

FeisStu

Game Programming

- Rendering
- Looping and control
- Math
- Animation
- Physics
- Behaviour and navigation (AI)
- Effects
- Networking

FeisStu

Artificial Intelligence (AI)

- Based on the current conditions, which actions should an entity take ?

Artificial Intelligence (AI)

- Based on the current conditions, which actions should an entity take ?
- The goal of AI
  - Looks like a real player ?
  - Beat the player ?

FeisStu

Considerations

- Performance
    - 60 FPS ?
    - The scale of quantities ?

FeisStu

Considerations

- Performance
- Explainability
    - AlphaGo ?

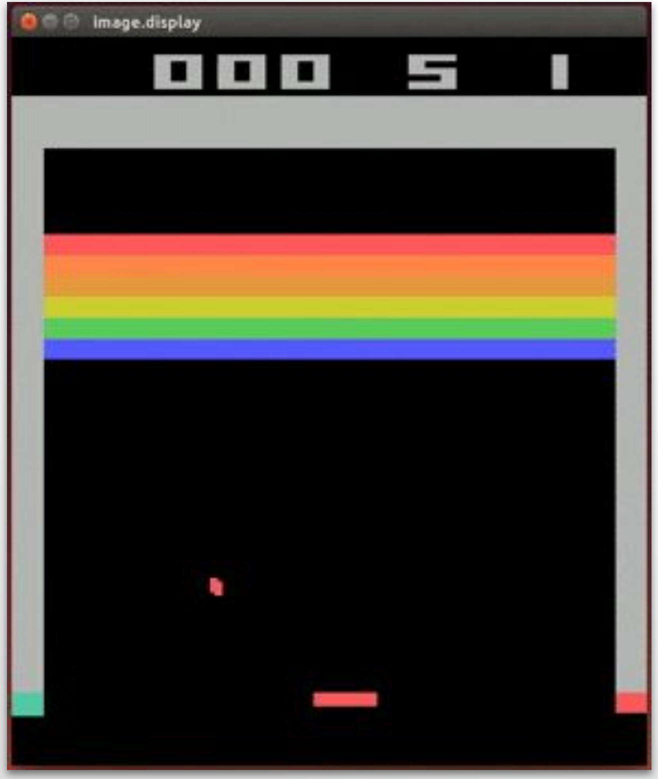
Considerations

- Performance
- Explainability
- Editability
    - ○ Human readable format ?
    - ○ Structured representation ?

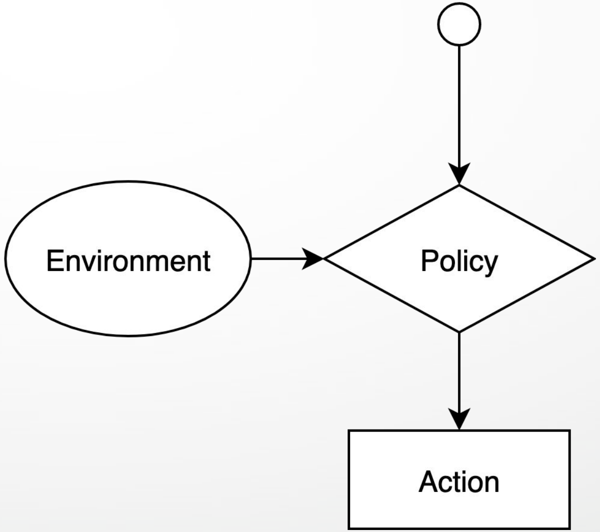
Considerations

- Performance
- Explainability
- Editability
- Flexibility
  - Easy to expand

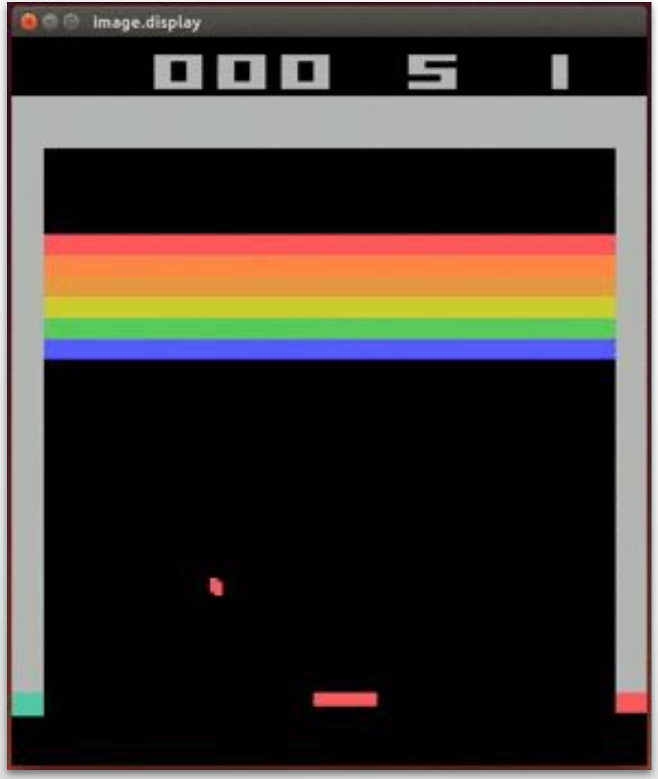
How to design / implement a Breakout AI ?

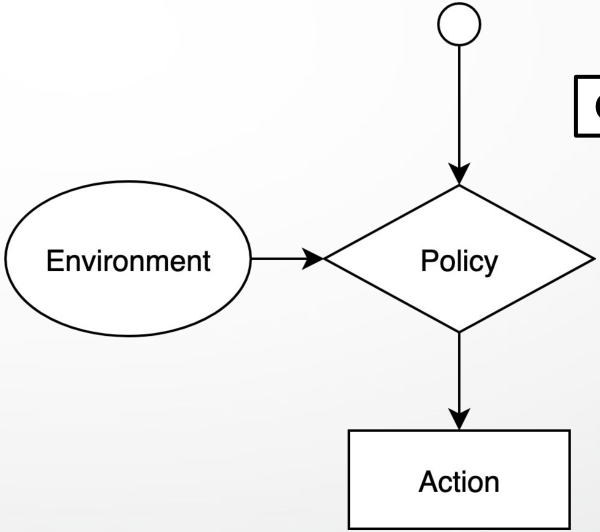


How to design / implement a Breakout AI ?



How to design / implement a Breakout AI ?



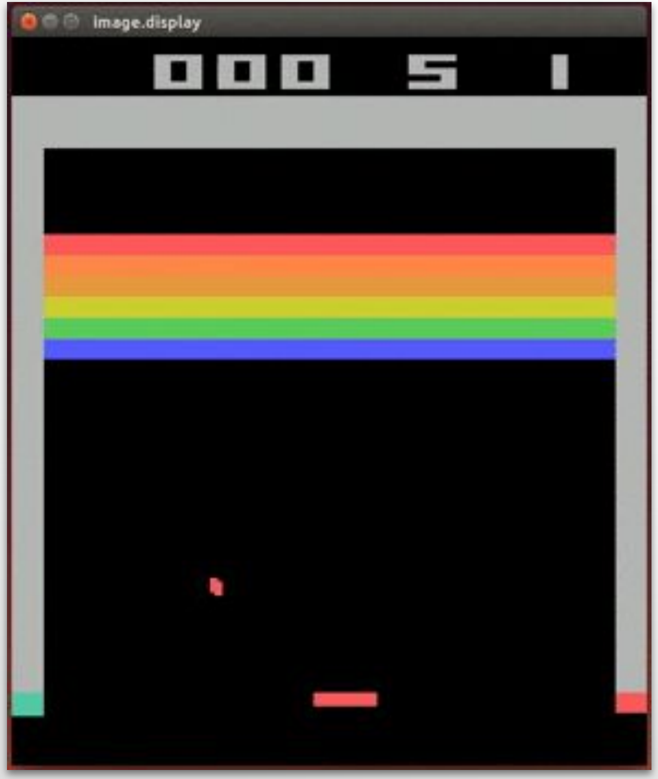


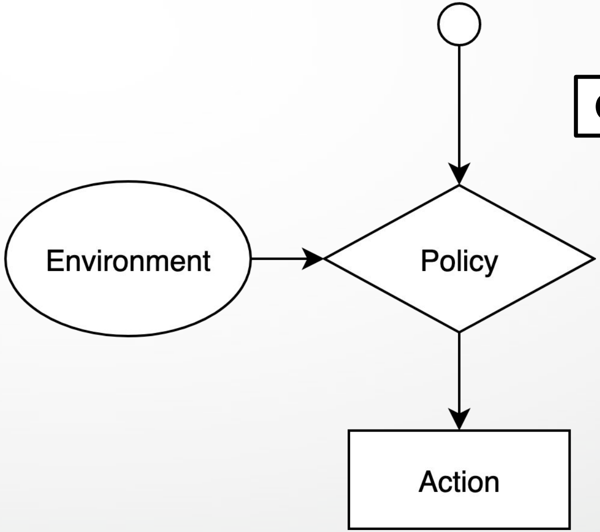
Goal ? (Reward)

DeepMind : DQN Breakout

FeisStu

How to design / implement a Breakout AI ?



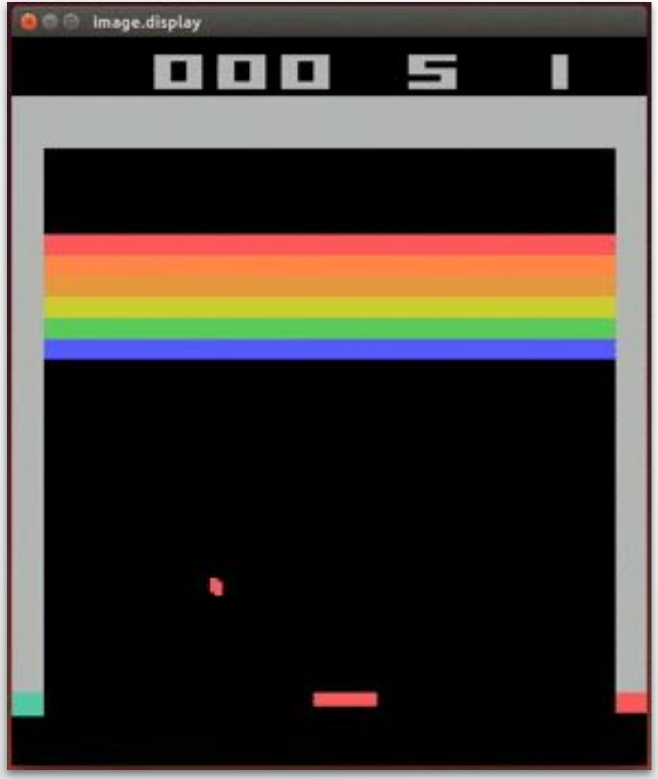


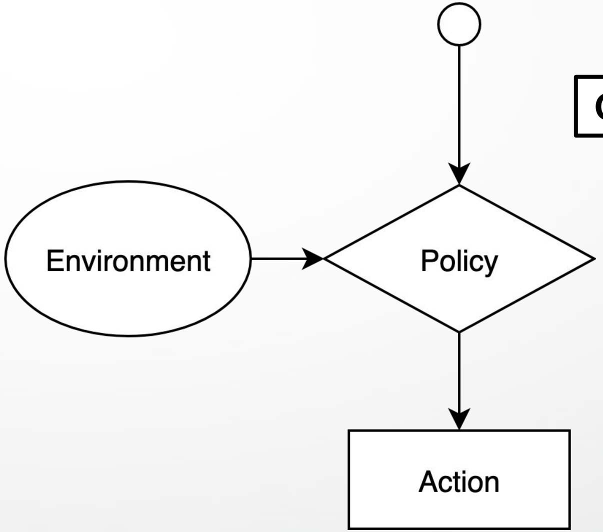
Goal ? (Reward)

Performance ?

DeepMind : DQN Breakout

How to design / implement a Breakout AI ?





Goal ? (Reward)

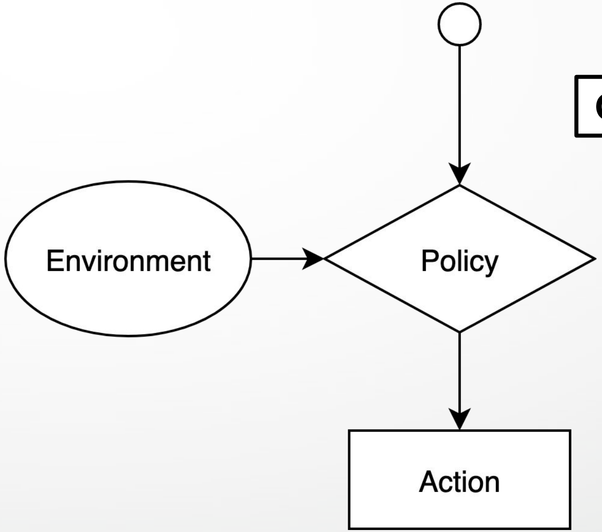
Performance ?

Explainability ?

DeepMind : DQN Breakout

FeisStu

How to design / implement a Breakout AI ?



Goal ? (Reward)

Performance ?

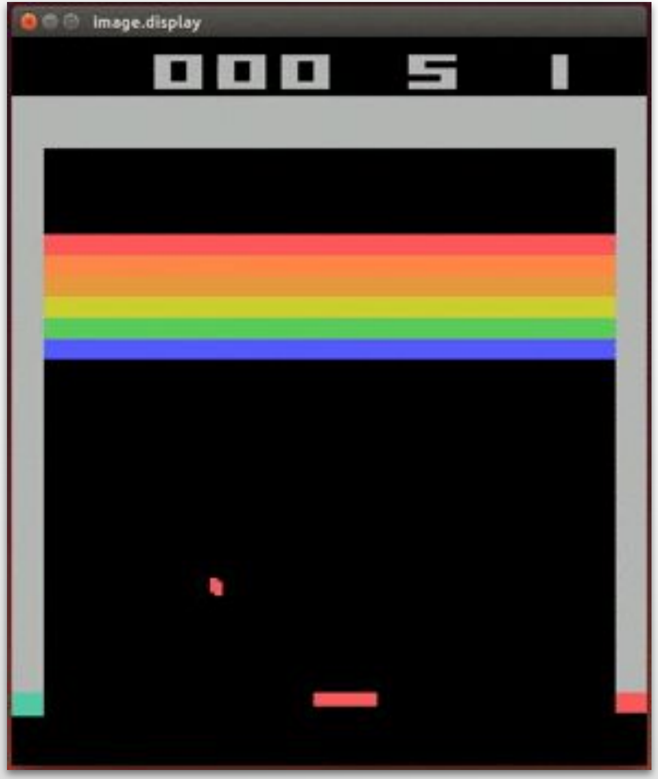
Explainability ?

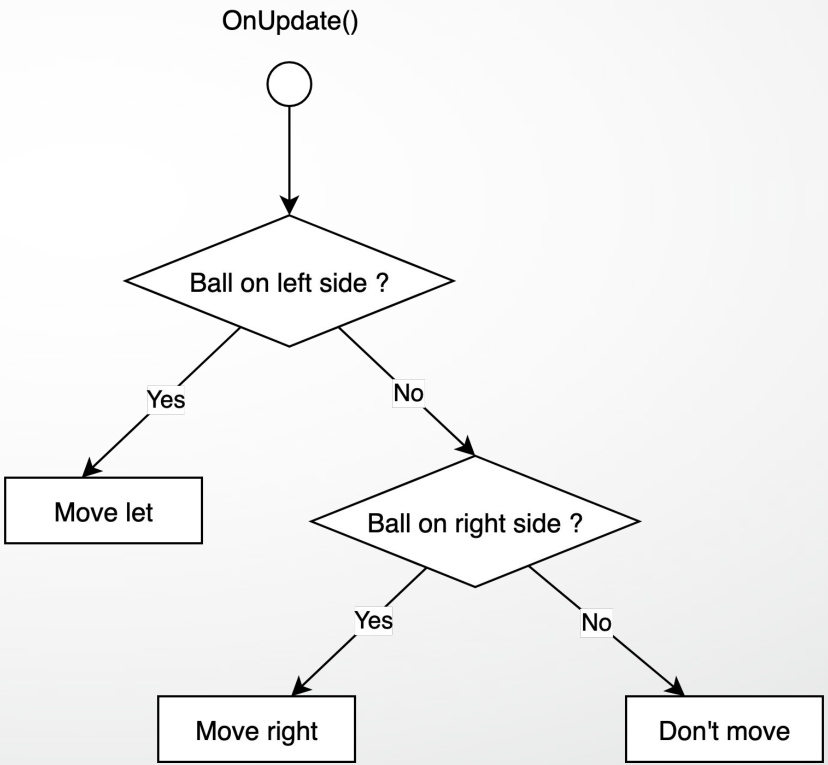
Editability ?

DeepMind : DQN Breakout

FeisStu

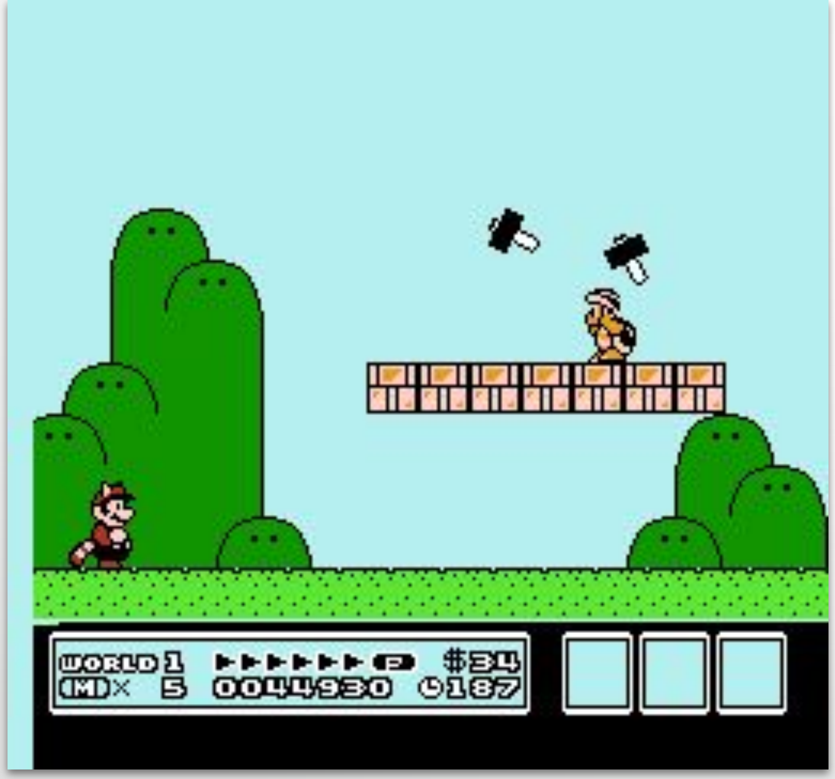
Decision trees



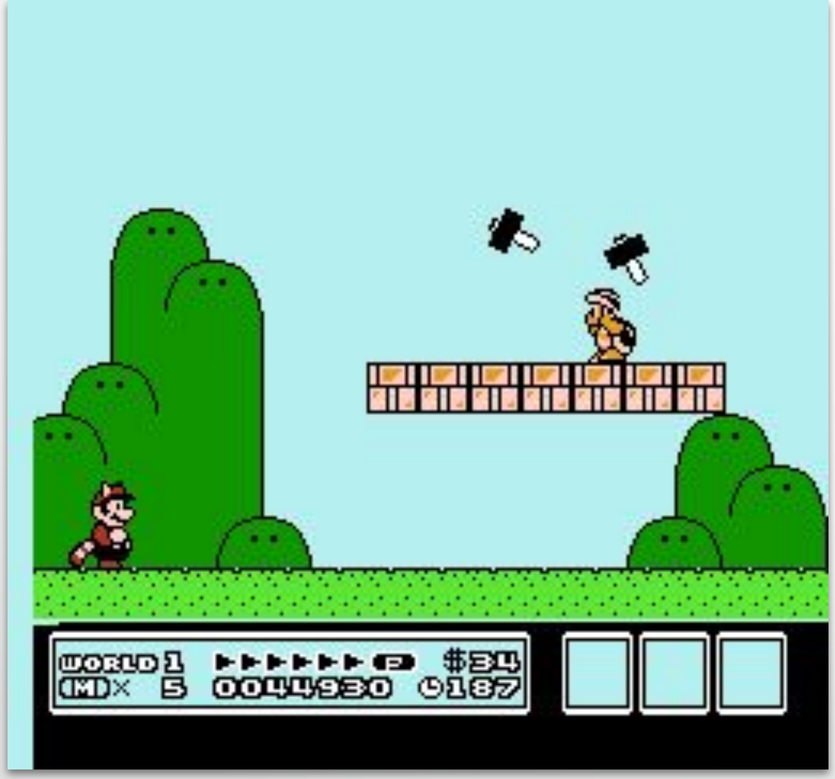


DeepMind : DQN Breakout

How to design / implement the turtle’s AI ?

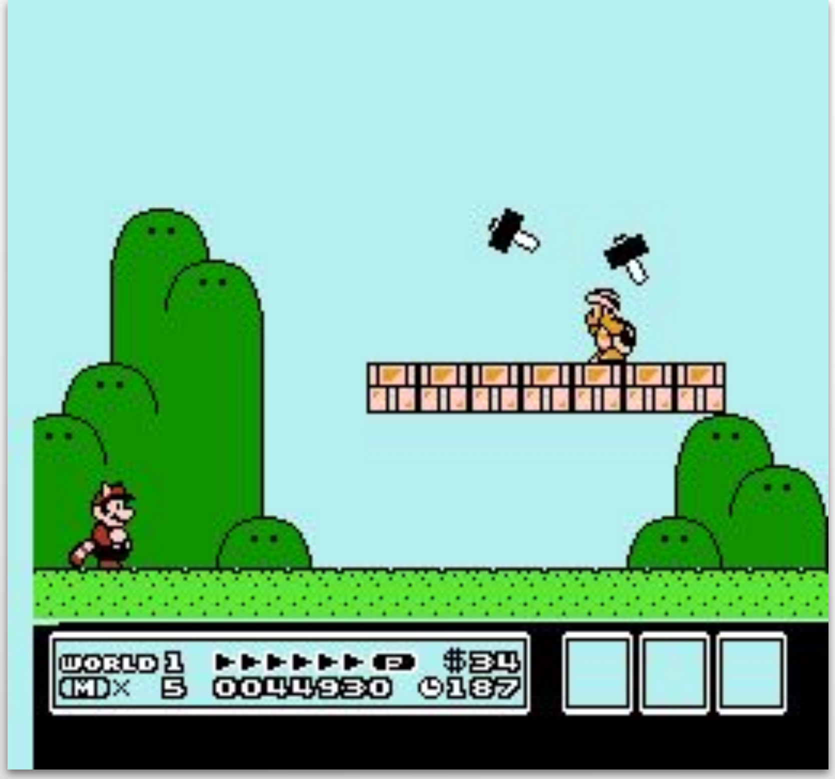


How to design / implement the turtle’s AI ?



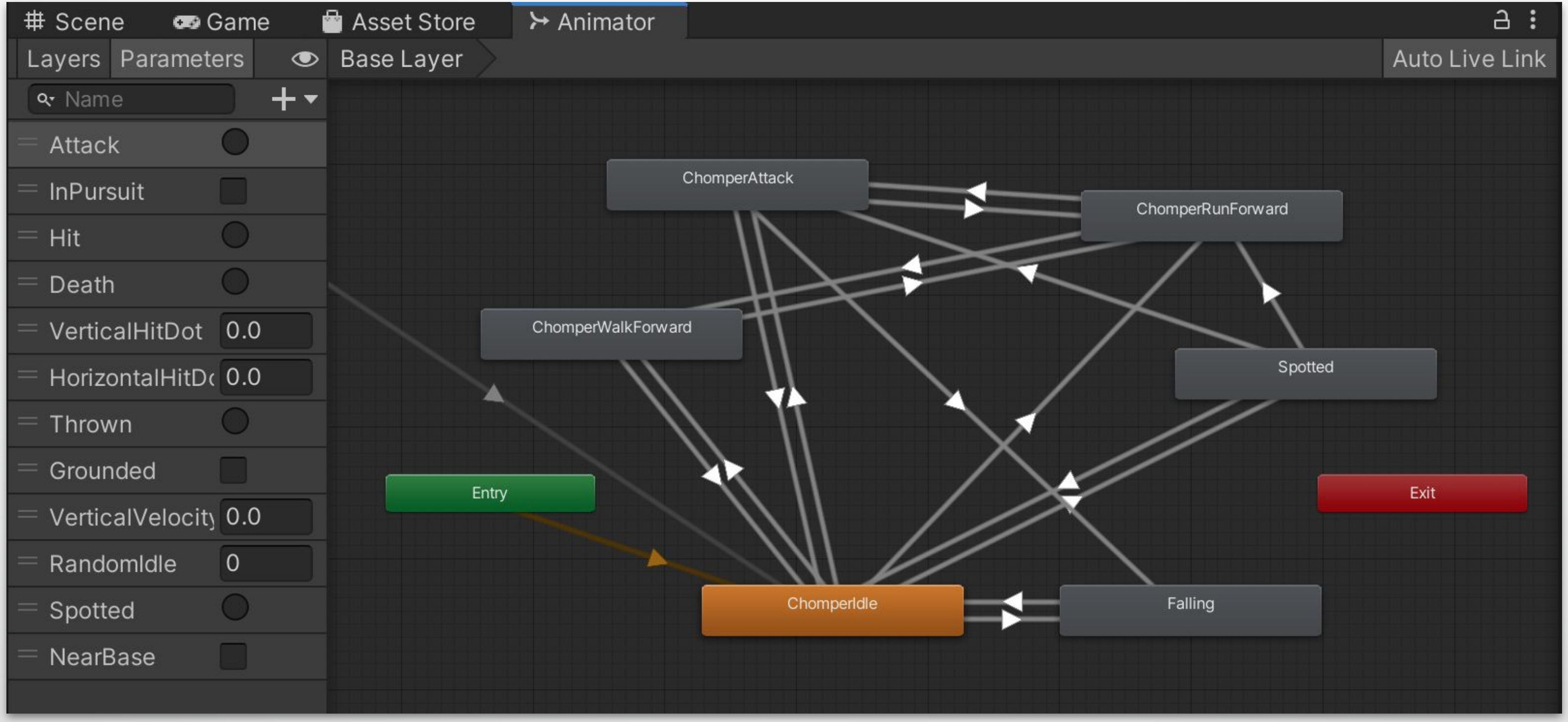
Try to draw the decision tree ?

Finite state machines



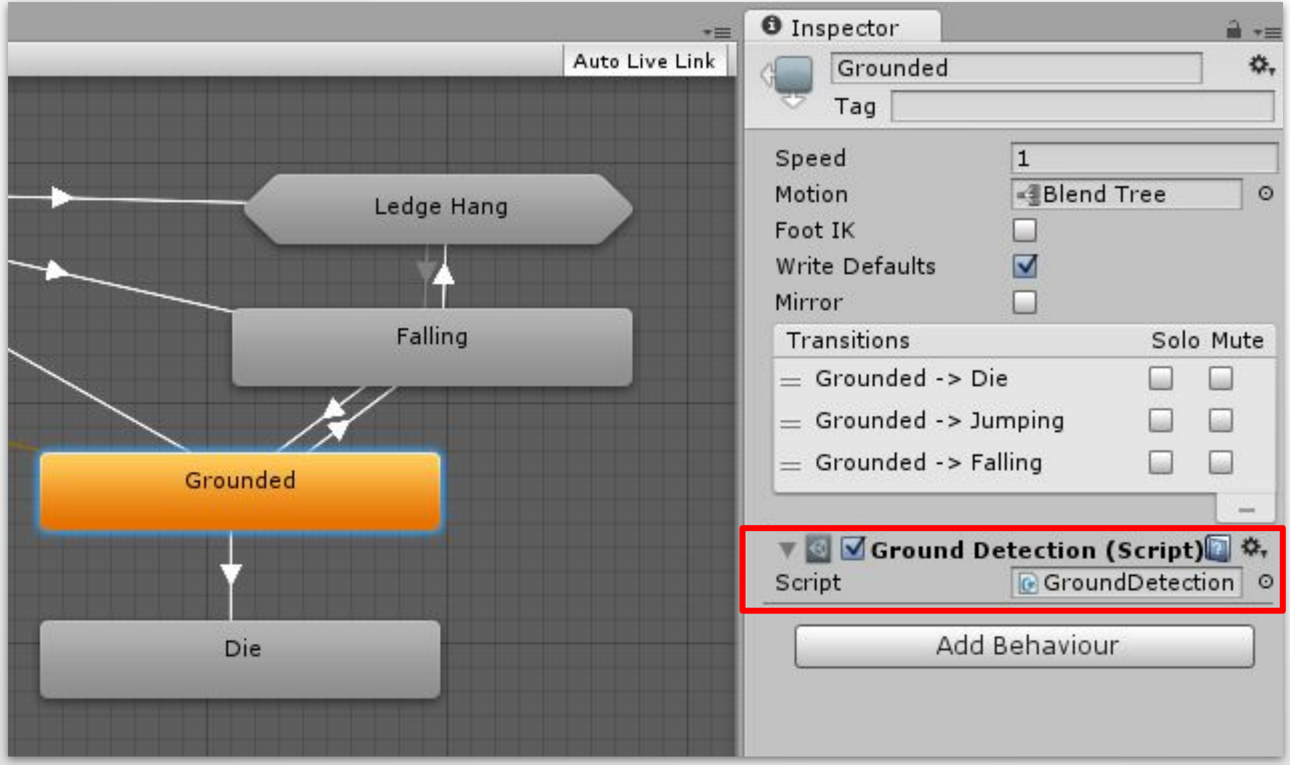
Try to draw the FSM ?

Animator controller



Attack	○
InPursuit	□
Hit	○
Death	○
VerticalHitDot	0.0
HorizontalHitD(	0.0
Thrown	○
Grounded	□
VerticalVelocity	0.0
RandomIdle	0
Spotted	○
NearBase	□

State machine behaviours



Grounded

Tag

Speed 1

Motion Blend Tree

Foot IK □

Write Defaults ☑

Mirror □

Transitions	Solo	Mute
Grounded -> Die	□	□
Grounded -> Jumping	□	□
Grounded -> Falling	□	□

☑ **Ground Detection (Script)**

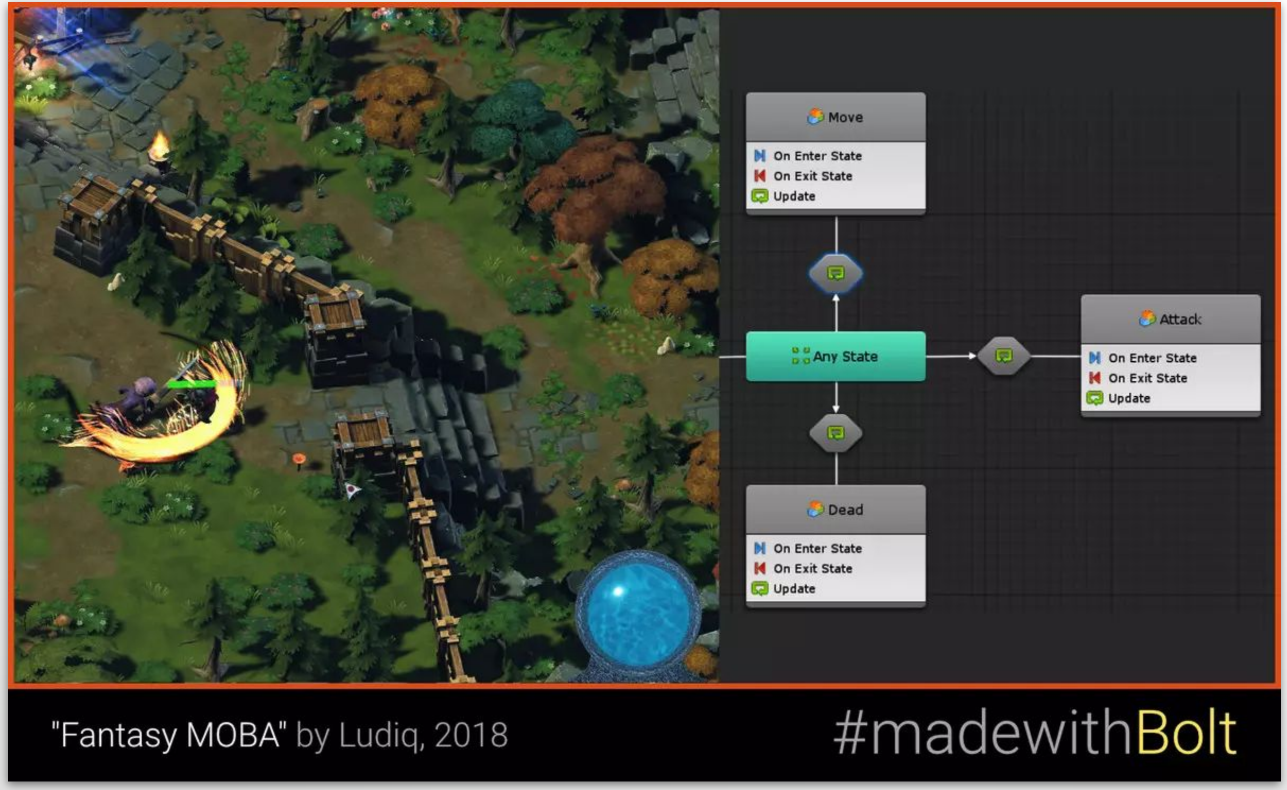
Script GroundDetection

Add Behaviour

DEMO: 3D Game Kit

- Steps :
  - Open scene “**Level2**”
  - Show the Animator of “**Chomper**”
  - Explain the **State machine behaviour** system of 3D Game kit
  - Explain “**ChomperBehaviour**”
  - Explain “**AttackBegin**” event

Unity Asset Store : Bolt [Free]

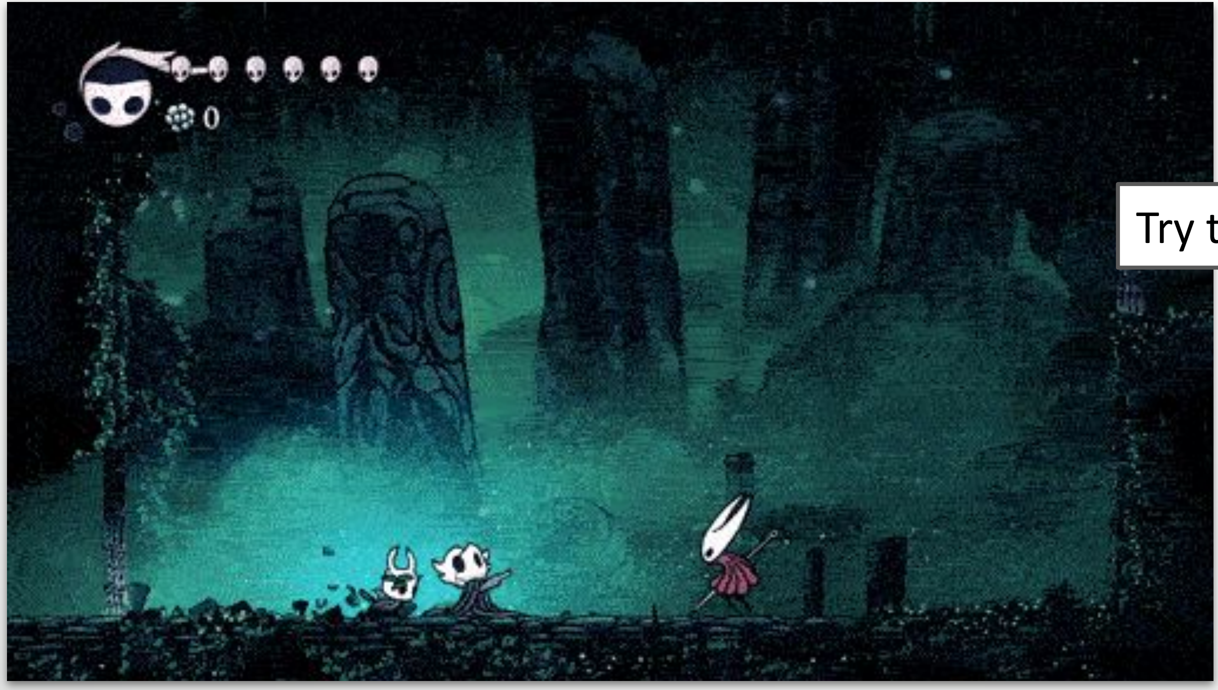


"Fantasy MOBA" by Ludiq, 2018

#madewithBolt

How to design / implement Hornet’s AI ?

How to design / implement Hornet’s AI ?



Try to draw the decision tree ?

How to design / implement Hornet’s AI ?

Try to draw the decision tree ?

Try to draw the FSM ?

Hierarchical state machines

Try to draw the decision tree ?

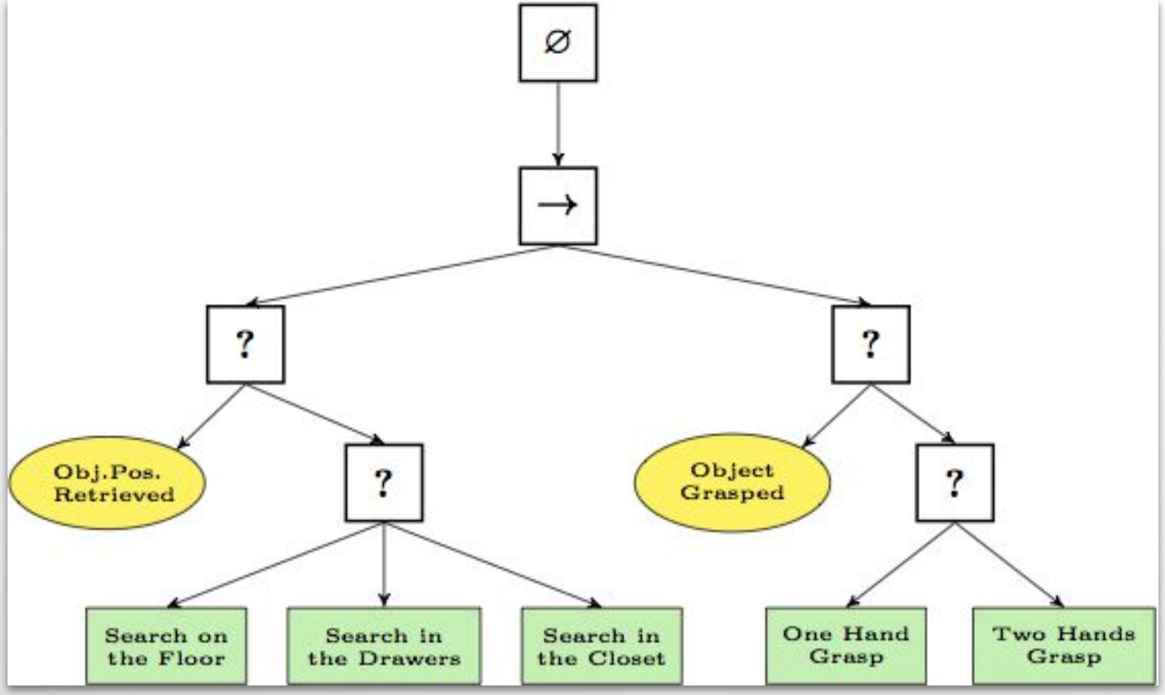
Try to draw the FSM ?

Try to draw the HFSM ?

DEMO: 3D Game Kit

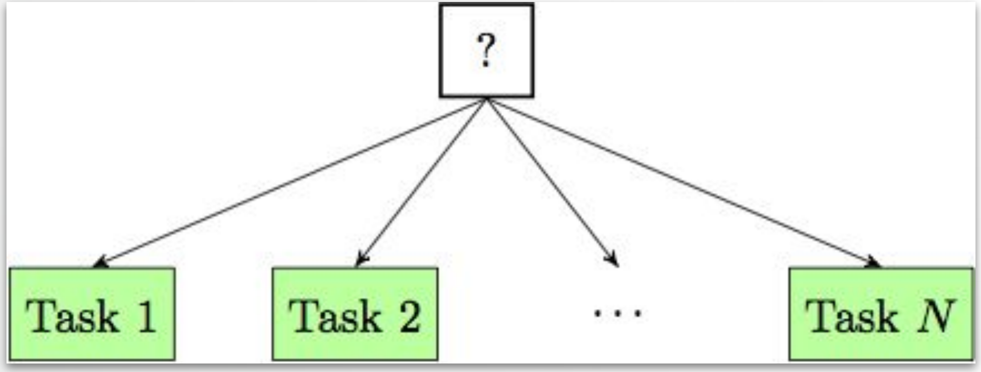
- Steps :
  - Open scene “**Level2”**
  - Show the Animator of “**Ellen**”
  - Explain the **Hierarchical state machines**
  - Explain the **“Any”** state

Behaviour trees



Wikipedia

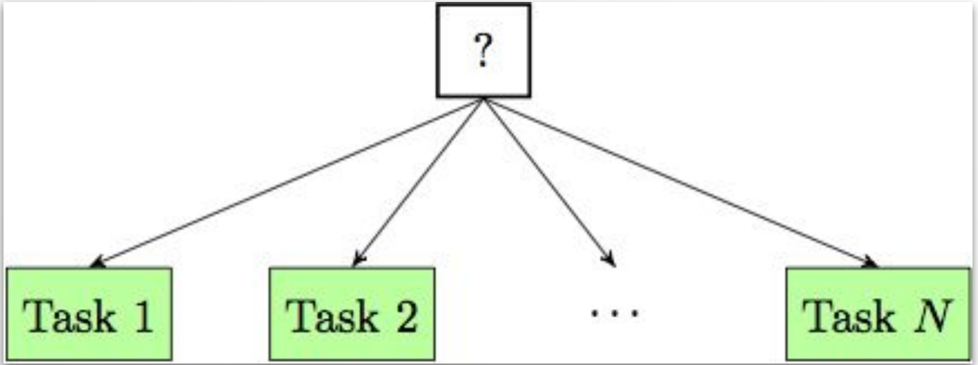
Selector (fallback) nodes

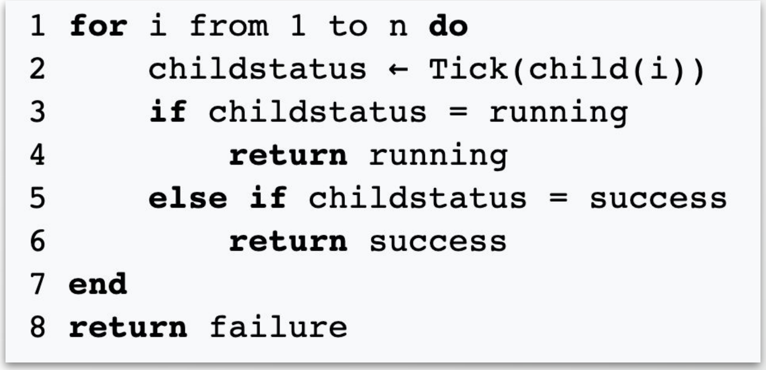


Wikipedia

FeisStu

Selector (fallback) nodes

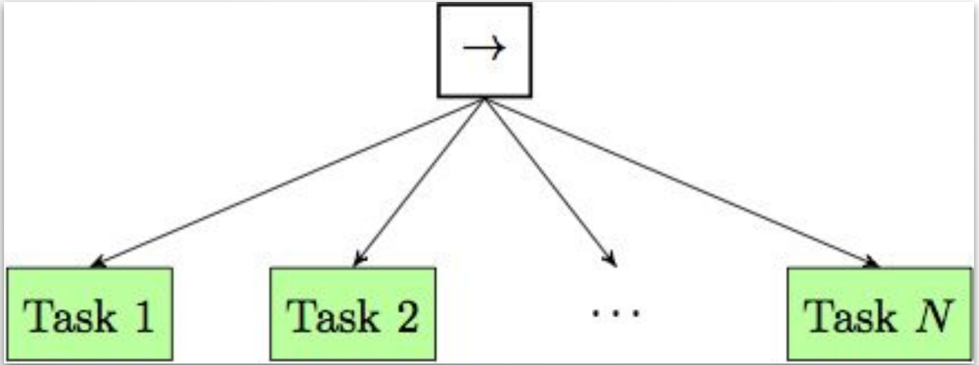


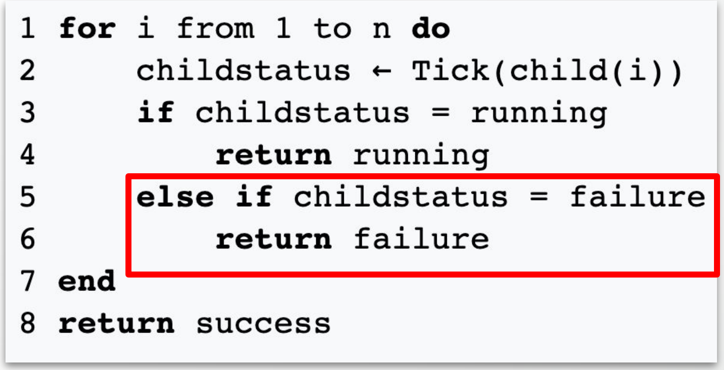


```
for i from 1 to n do
    childstatus ← Tick(child(i))
    if childstatus = running
        return running
    else if childstatus = success
        return success
end
return failure
```

Wikipedia

Sequence node



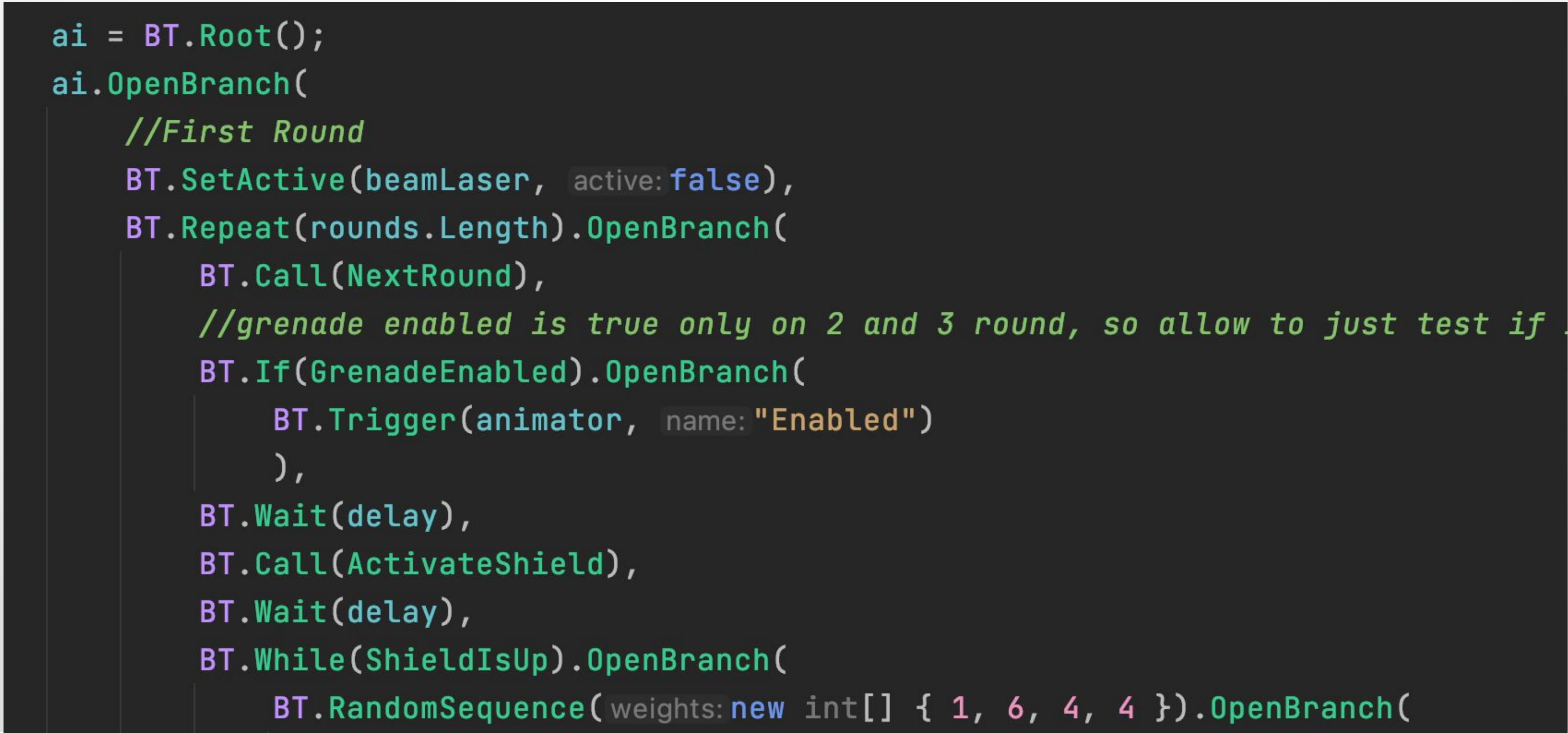


```
for i from 1 to n do
    childstatus ← Tick(child(i))
    if childstatus = running
        return running
    else if childstatus = failure
        return failure
end
return success
```

DEMO: 2D Game Kit

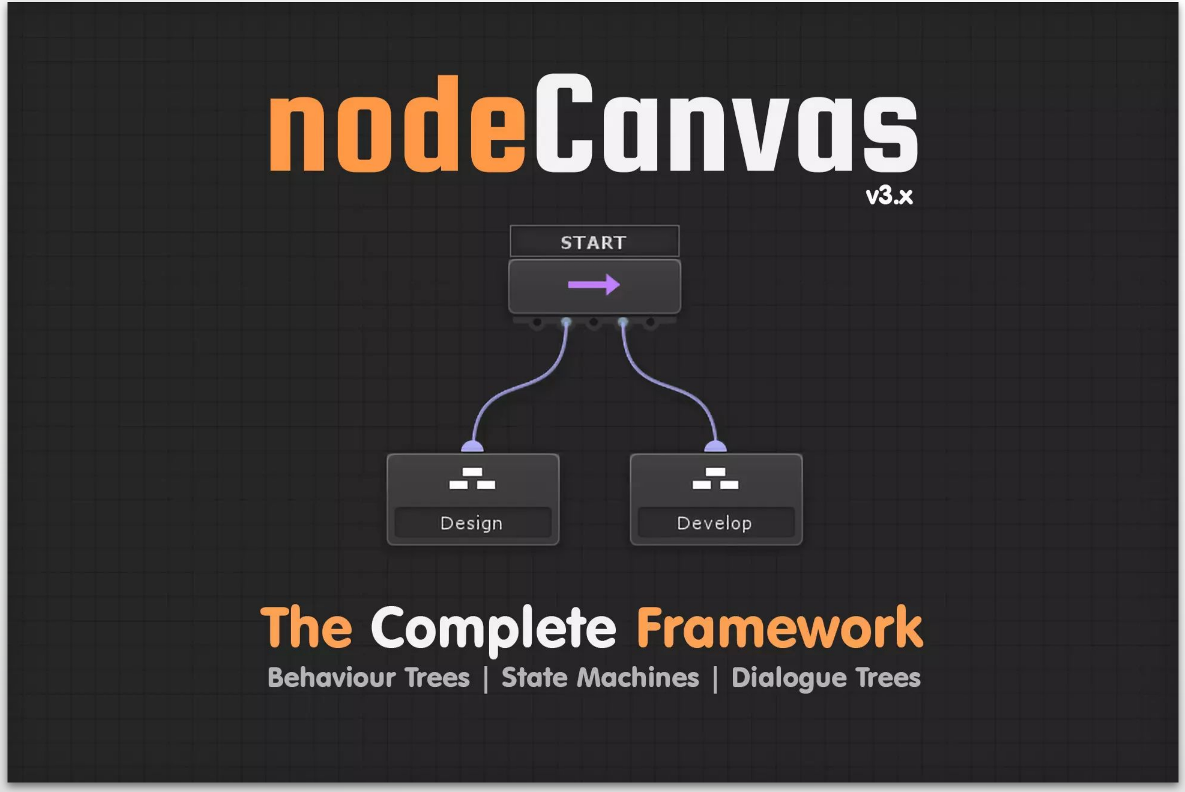
- Steps:
  - Open scene **“Zone 5”**
  - Show the animator of **Gunner**
  - Explain the state topology
  - Explain “**Missle Golem**” component
    - **OnEnable**

DEMO: 2D Game Kit

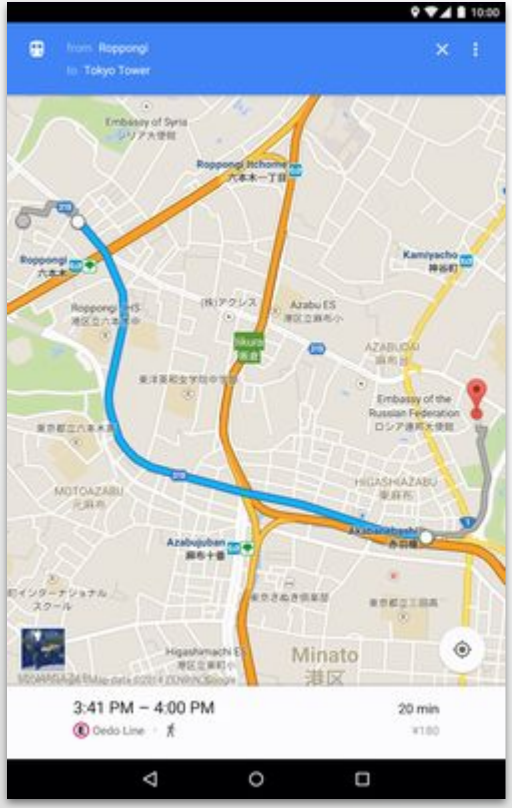


```
ai = BT.Root();
ai.OpenBranch(
    //First Round
    BT.SetActive(beamLaser, active: false),
    BT.Repeat(rounds.Length).OpenBranch(
        BT.Call(NextRound),
        //grenade enabled is true only on 2 and 3 round, so allow to just test if
        BT.If(GrenadeEnabled).OpenBranch(
            BT.Trigger(animator, name: "Enabled")
        ),
        BT.Wait(delay),
        BT.Call(ActivateShield),
        BT.Wait(delay),
        BT.While(ShieldIsUp).OpenBranch(
            BT.RandomSequence( weights: new int[] { 1, 6, 4, 4 }).OpenBranch(
```

Unity Asset Store : Node Canvas



Navigation



from Roppongi

to Tokyo Tower

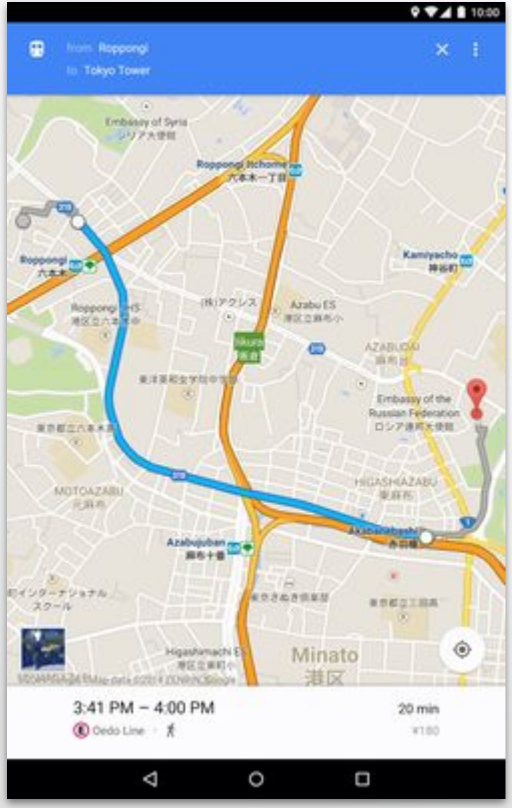
3:41 PM – 4:00 PM

20 min

Oedo Line

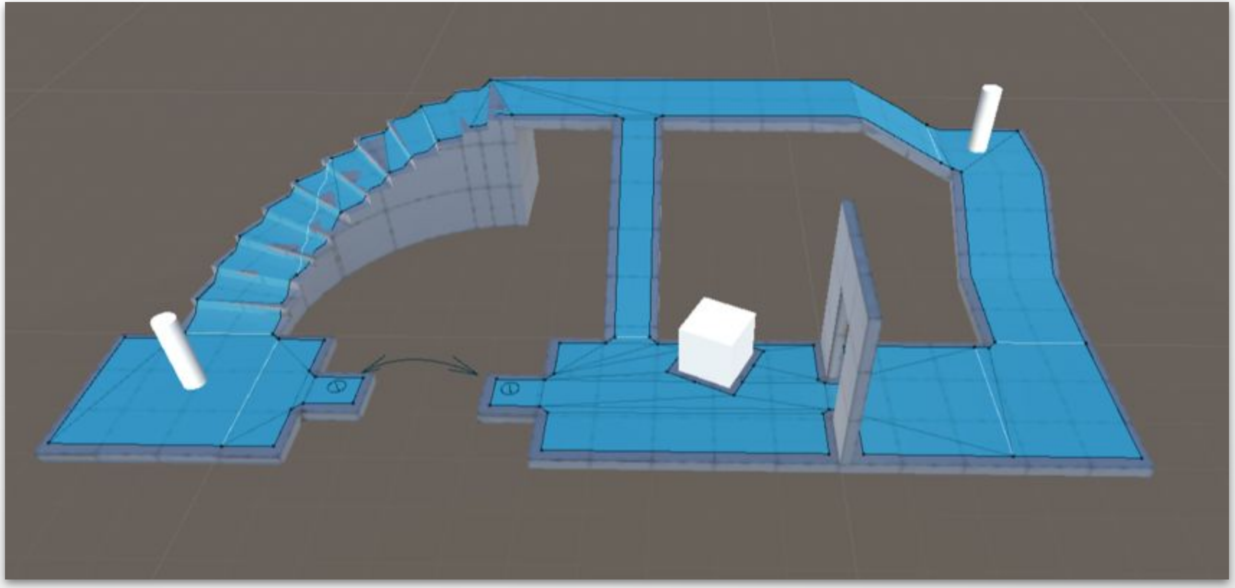
¥180

Navigation

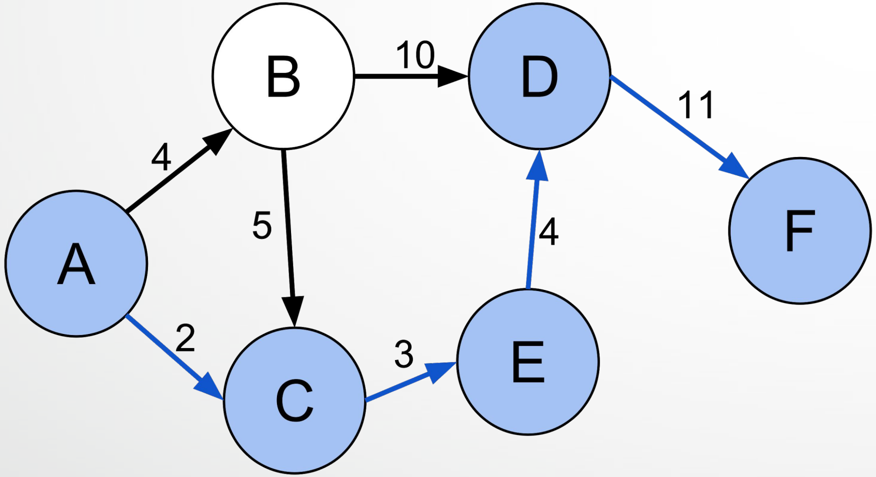


from Roppongi
to Tokyo Tower

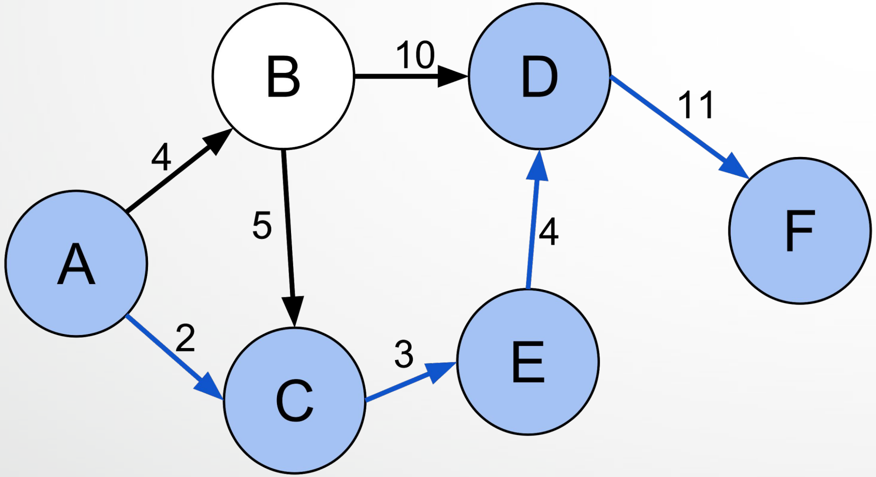
3:41 PM – 4:00 PM
Oedo Line
20 min
¥180



Path finding

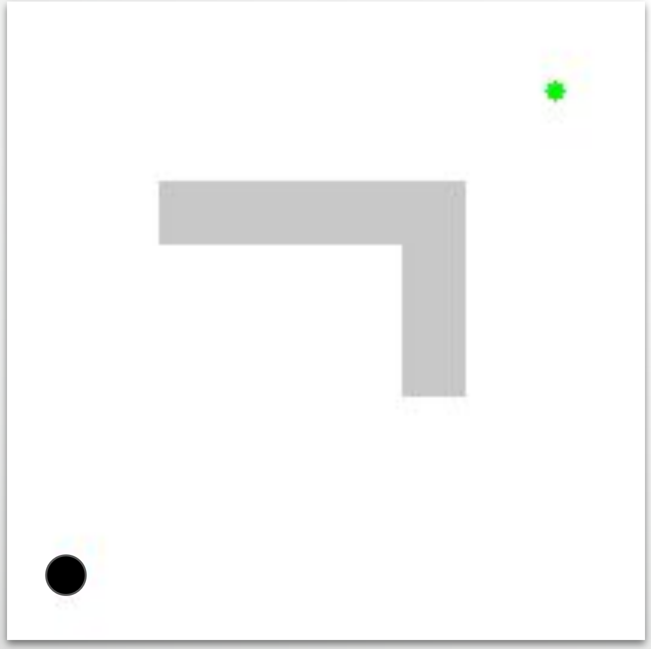


Path finding



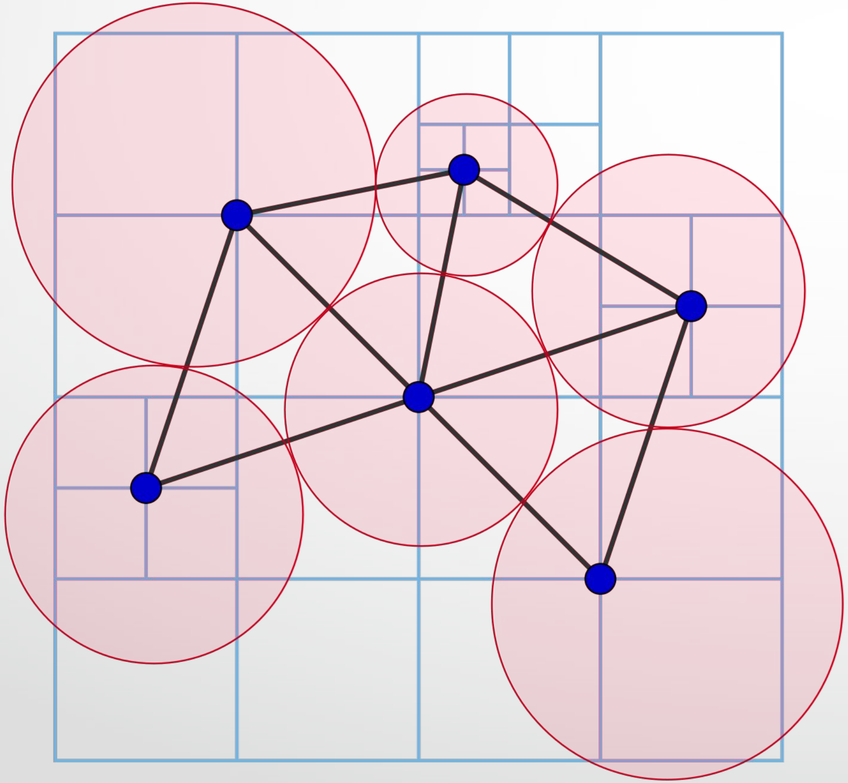
- Common algorithms :
  - BFS / DFS

Path finding



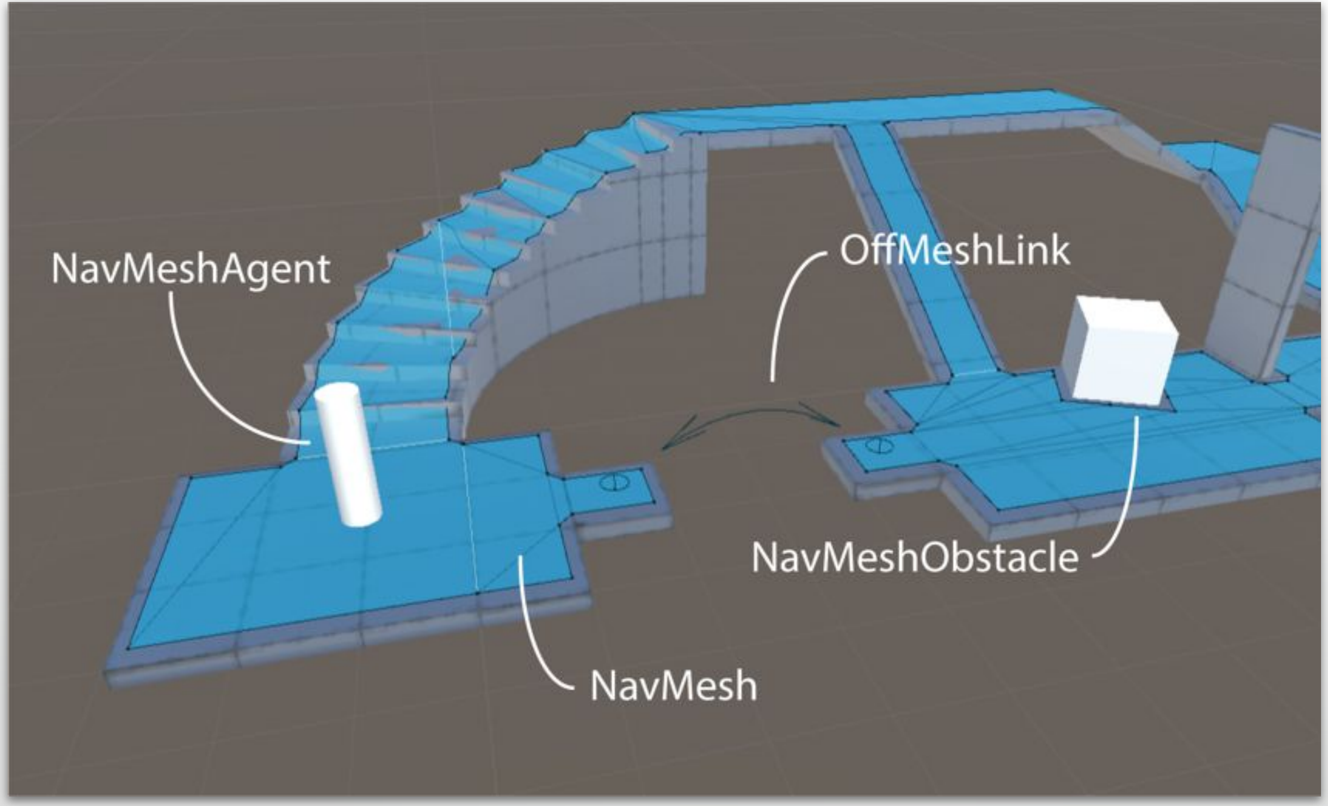
- Common algorithms :
    - BFS / DFS
        - A*
            - Dijkstra's algorithm

Hierarchical path finding

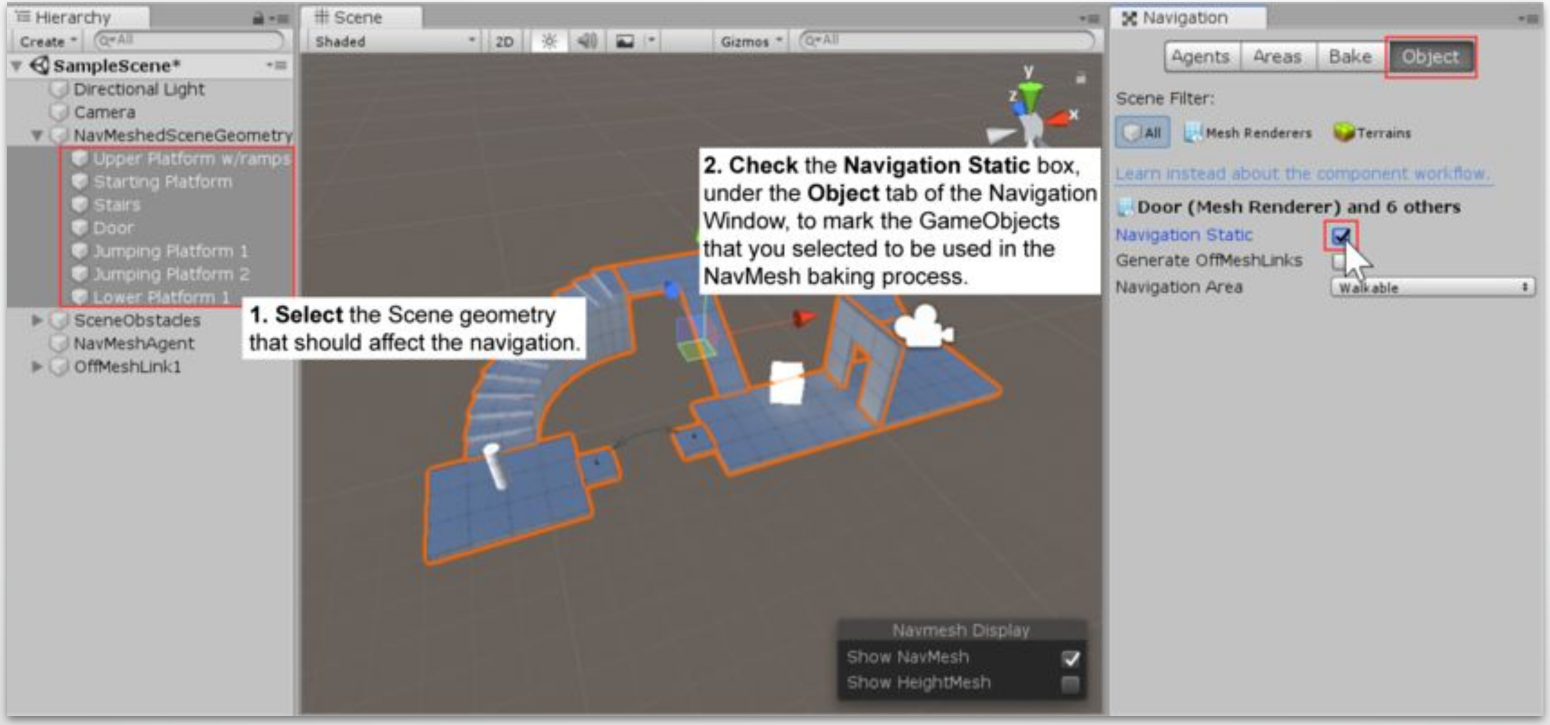


- Common algorithms :
  - BFS / DFS
    - A*
      - Dijkstra's algorithm
- Global / Local

Navmesh



Build a navmesh



1. **Select** the Scene geometry that should affect the navigation.

2. **Check** the **Navigation Static** box, under the **Object** tab of the Navigation Window, to mark the GameObjects that you selected to be used in the NavMesh baking process.

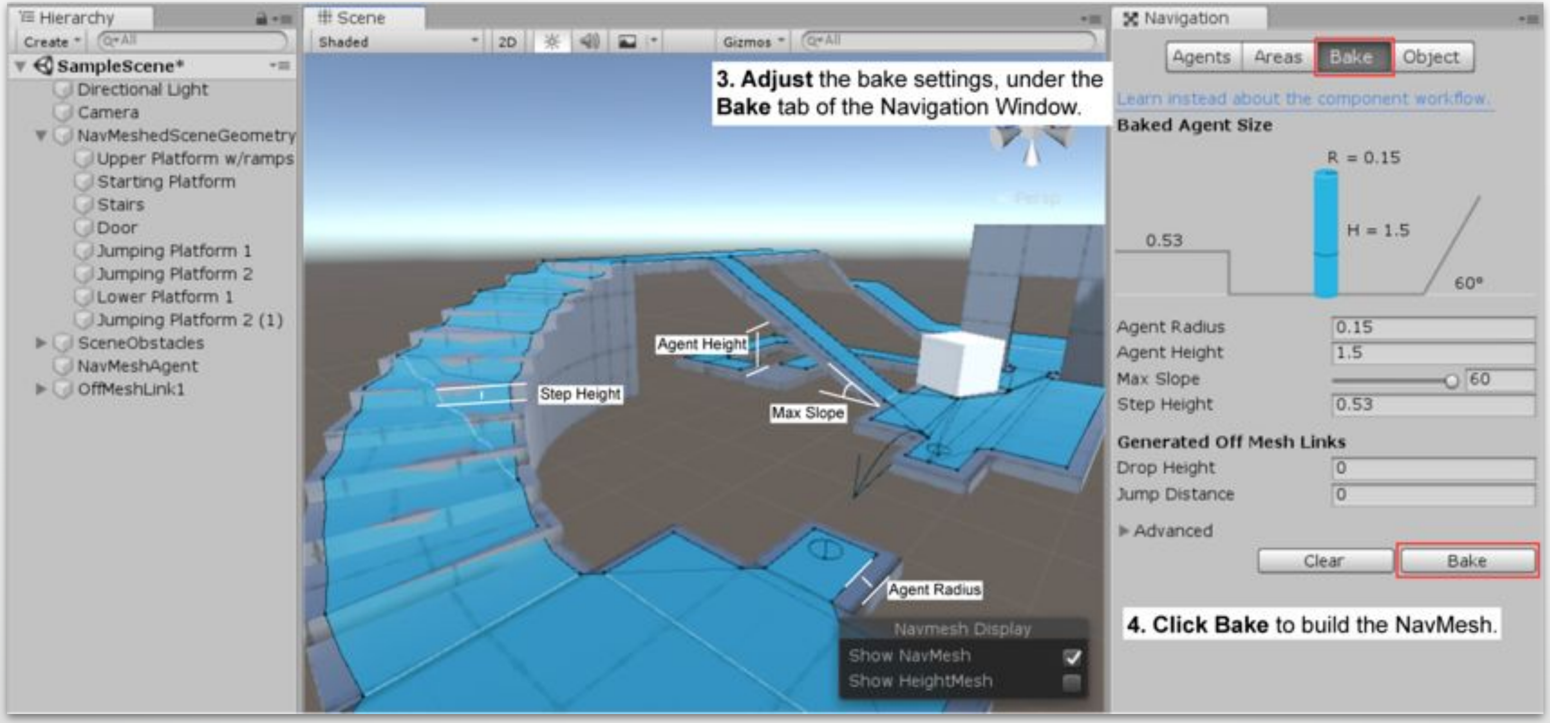
Door (Mesh Renderer) and 6 others

Navigation Static ☑

Generate OffMeshLinks □

Navigation Area: Walkable

Build a navmesh



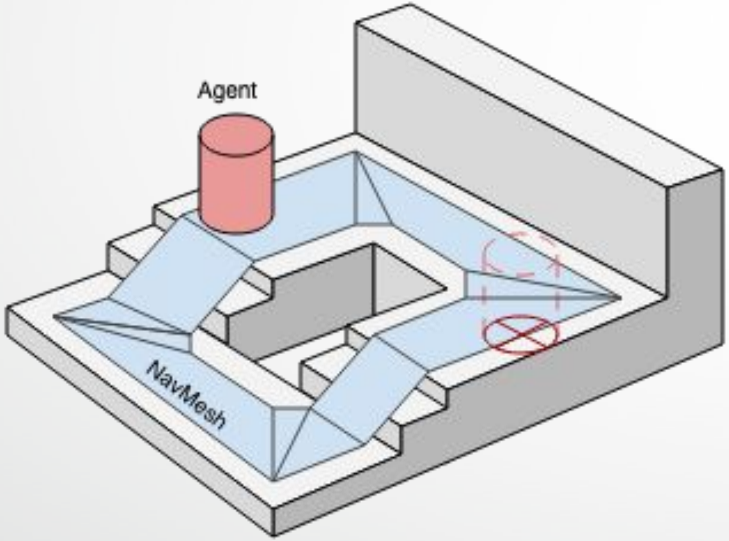
Show NavMesh ☑
Show HeightMesh □

Agent Radius: 0.15
Agent Height: 1.5
Max Slope: 60
Step Height: 0.53
Drop Height: 0
Jump Distance: 0

3. Adjust the bake settings, under the **Bake** tab of the Navigation Window.

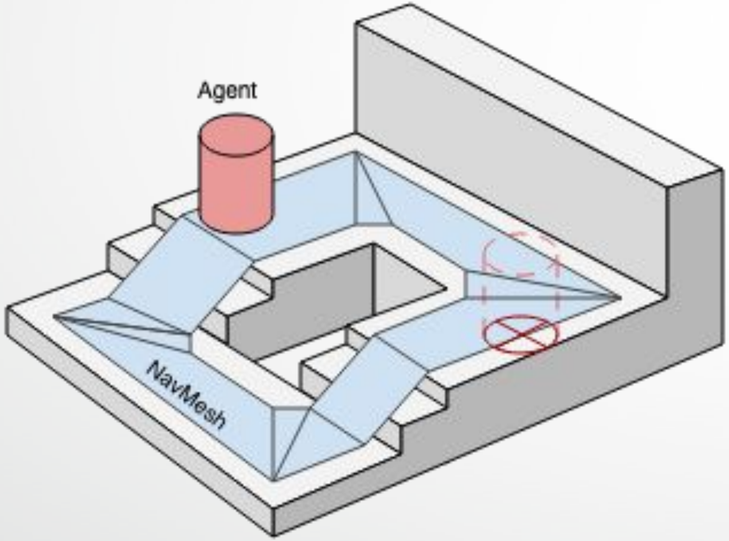
4. Click Bake to build the NavMesh.

Walking

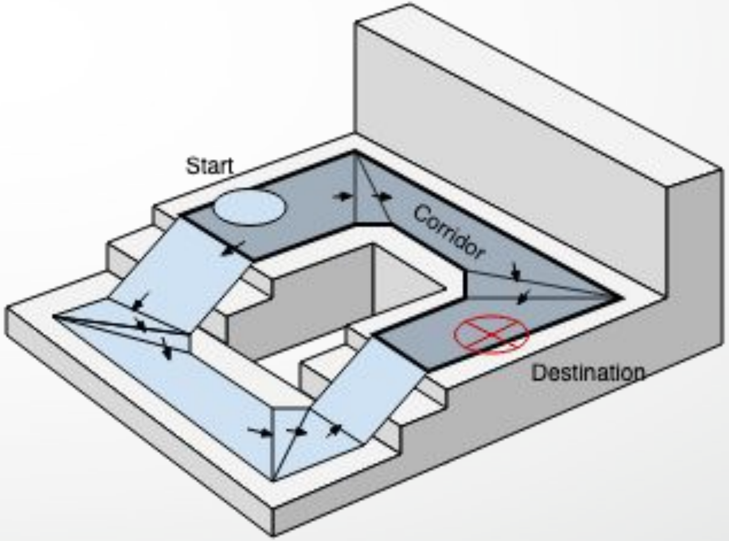


Walkable area

Walking

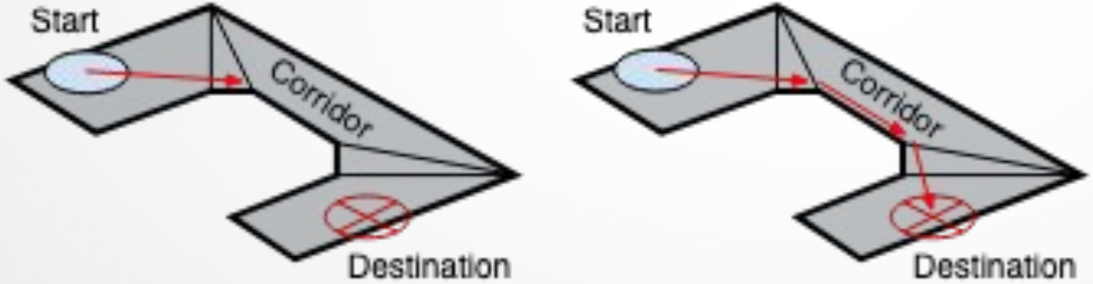


Walkable area



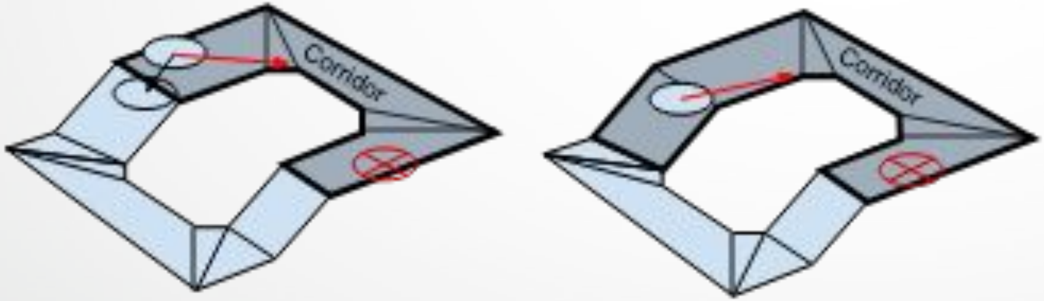
Finding paths

Walking



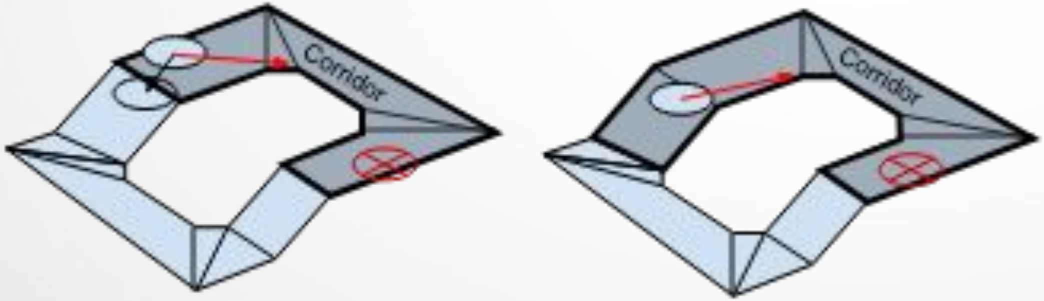
Global to local

Avoid agents/ obstacles

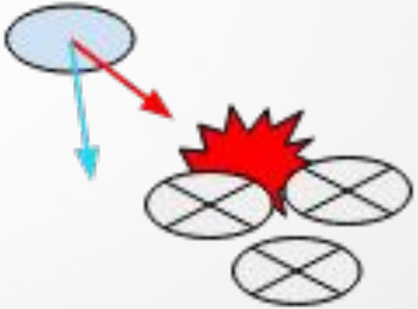


Local avoidance

Avoid agents/ obstacles

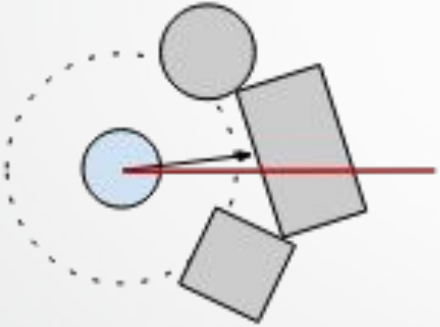


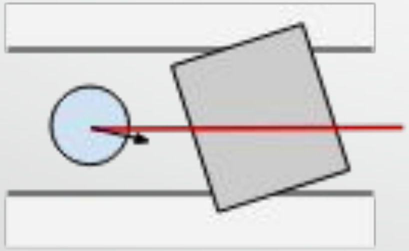
Local avoidance



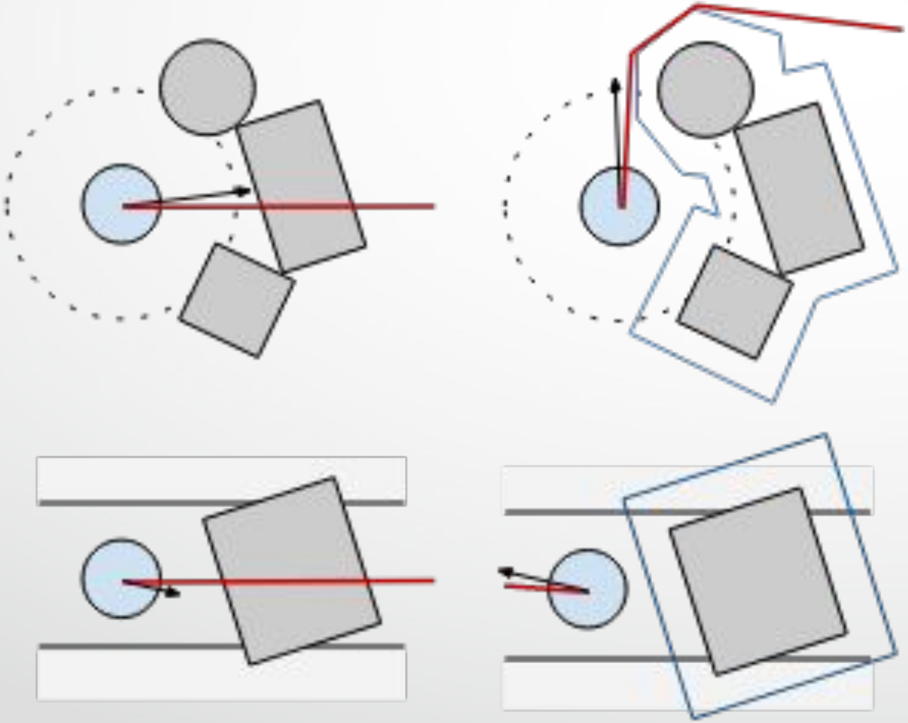
reciprocal velocity obstacles

Avoid agents/ obstacles



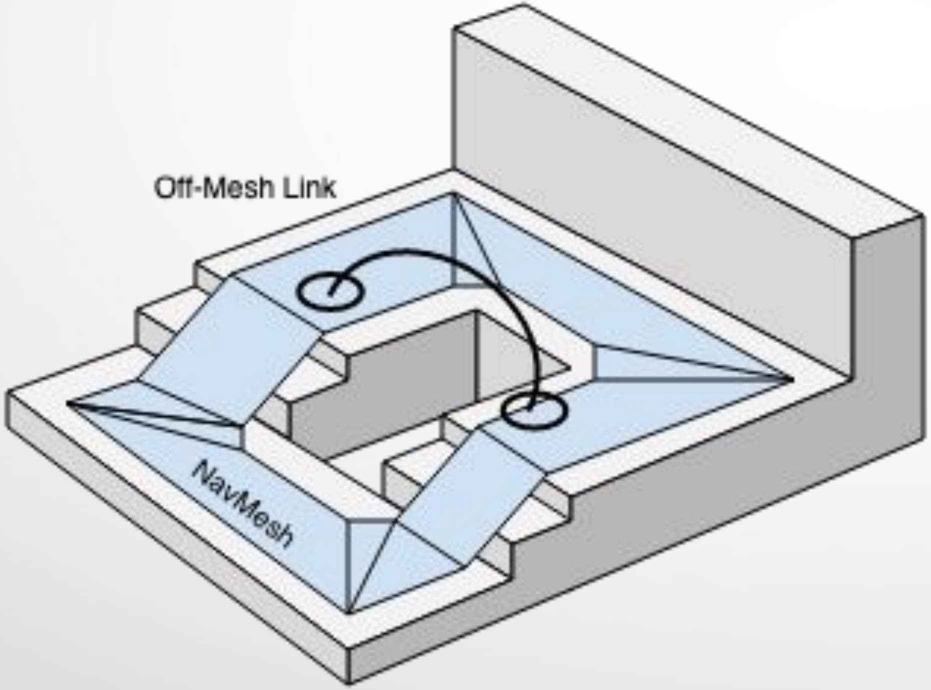


Avoid agents/ obstacles



Carving

Off-mesh link

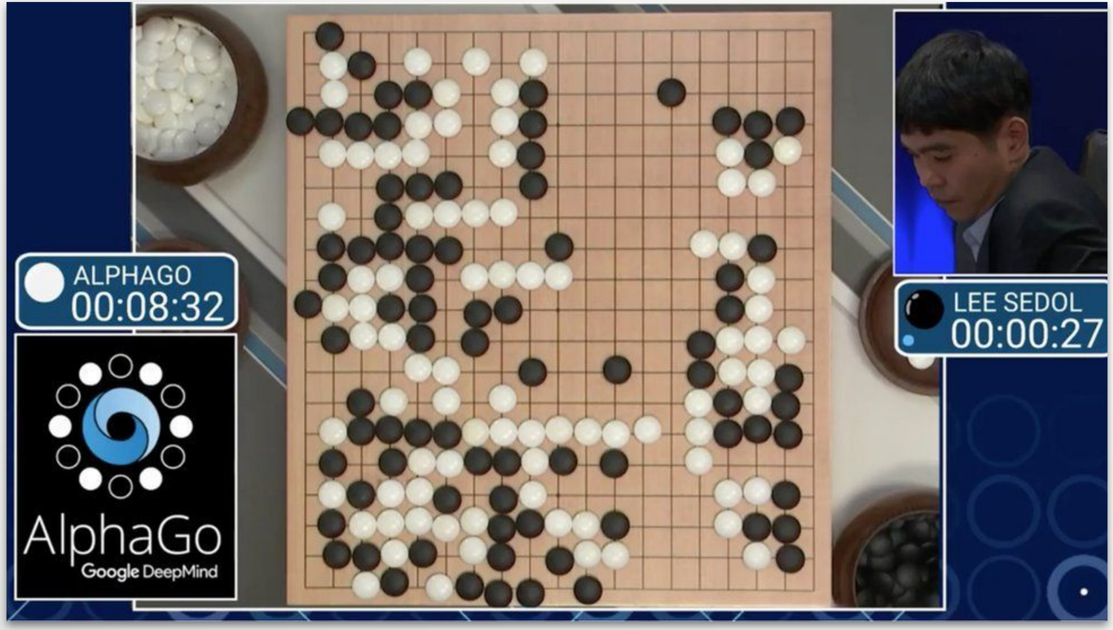


Jump / Climb / Teleport / ...

DEMO: 3D Game Kit

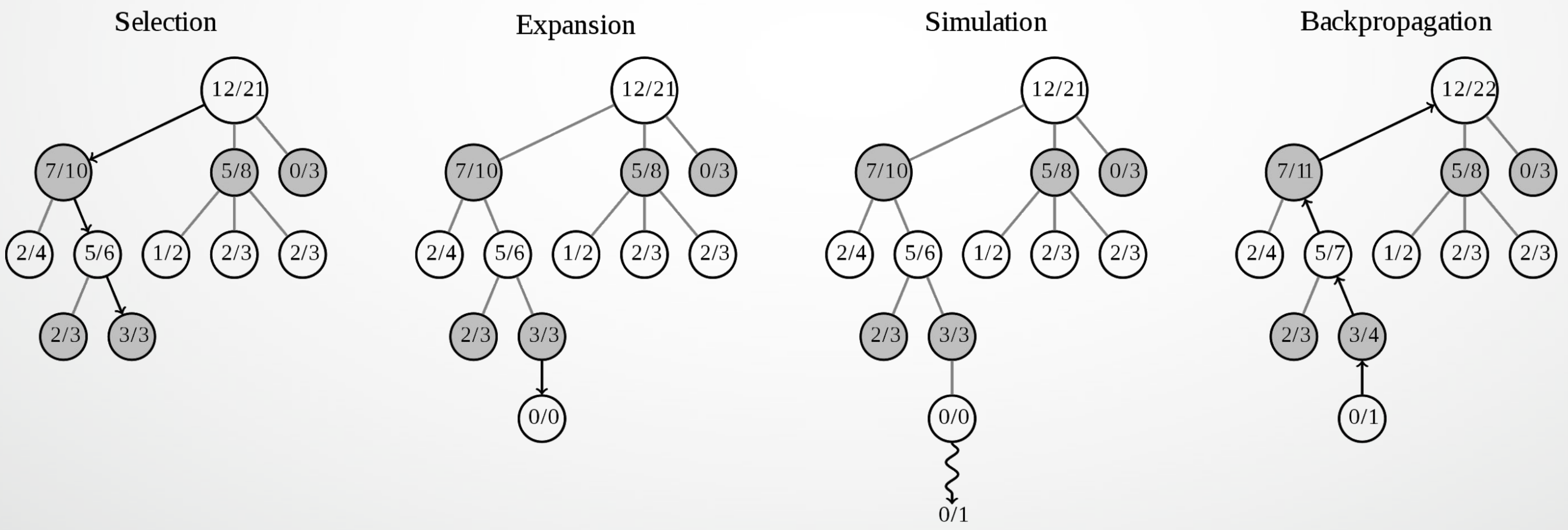
- Steps:
  - Create new **Scene**
  - Add “**Chomper**”
    - Select agentType “**Humanoid**”
  - Bake **Navmesh**

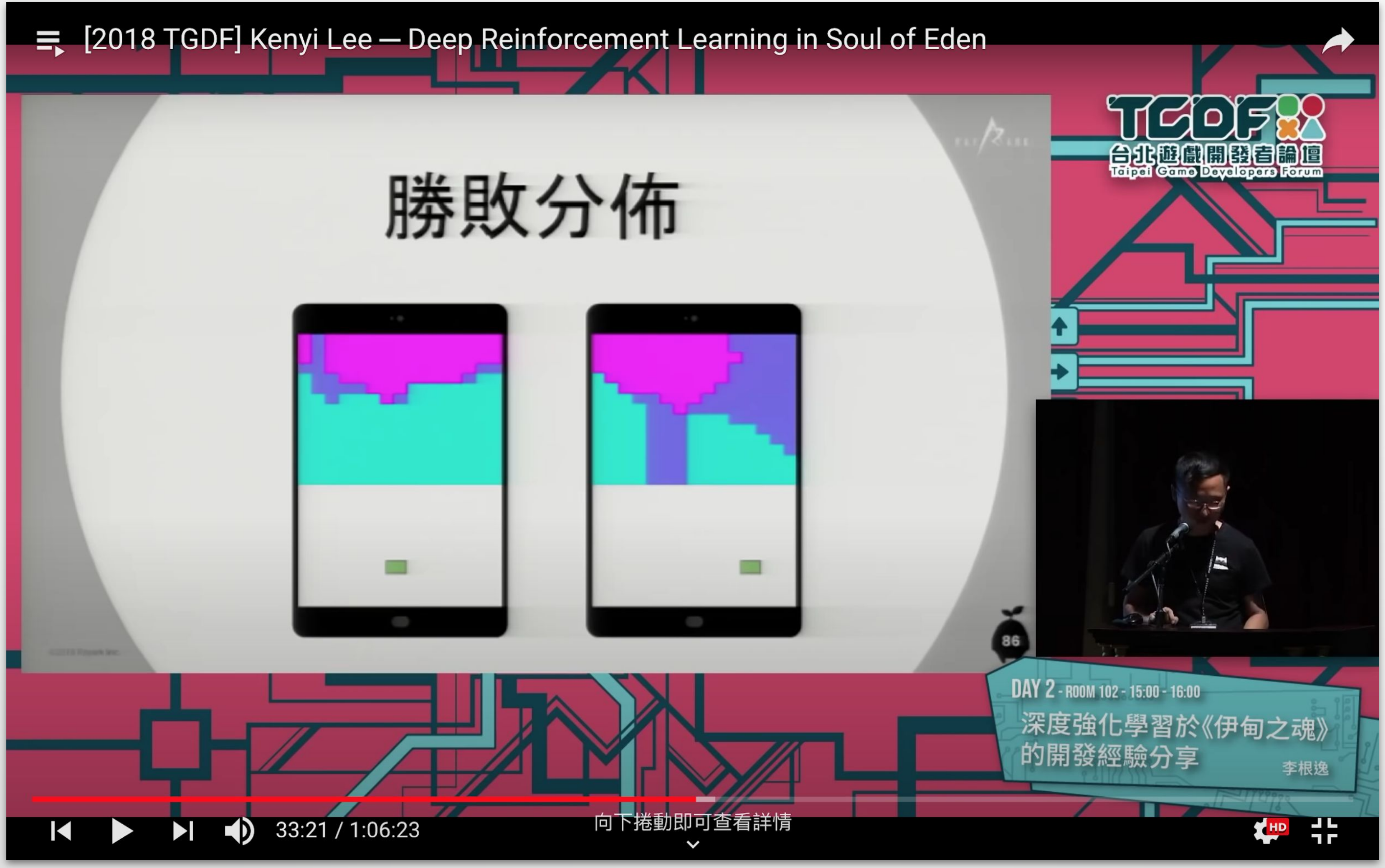
Planning



- BFS / DFS , A* algorithms, ...

Monte Carlo tree search





[2018 TGDF] Kenyi Lee — Deep Reinforcement Learning in Soul of Eden

勝敗分佈

DAY 2 - ROOM 102 - 15:00 - 16:00

深度強化學習於《伊甸之魂》的開發經驗分享

李根逸

Q & A