

Game AI



Ken-Yi Lee

Game Programming, Fall 2020 @ National Taiwan University

Game Programming

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

Game Programming

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

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 ?

Considerations

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

Considerations

- Performance
- Explainability
 - AlphaGo ?

Considerations

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

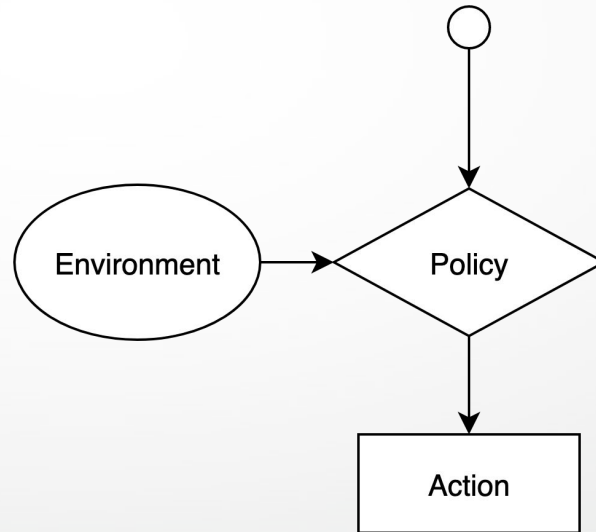
Considerations

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

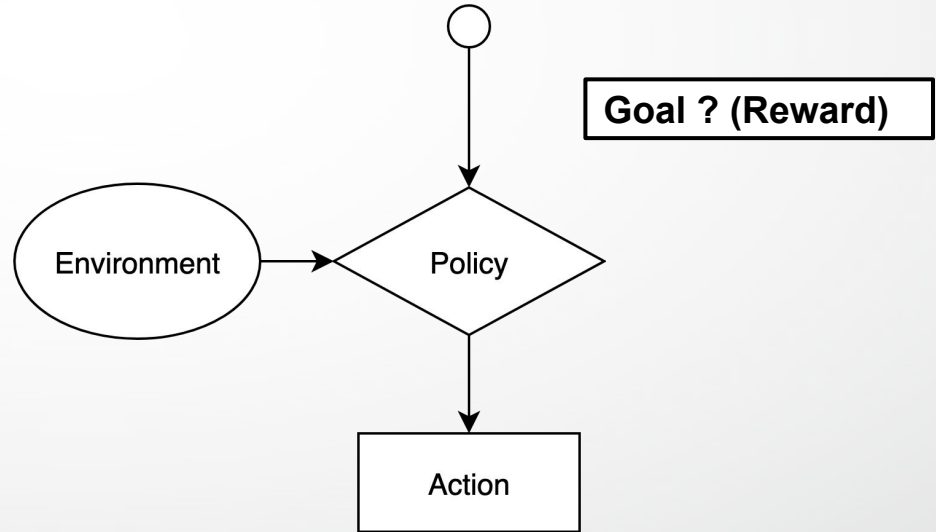
How to design / implement a Breakout AI ?



How to design / implement a Breakout AI ?

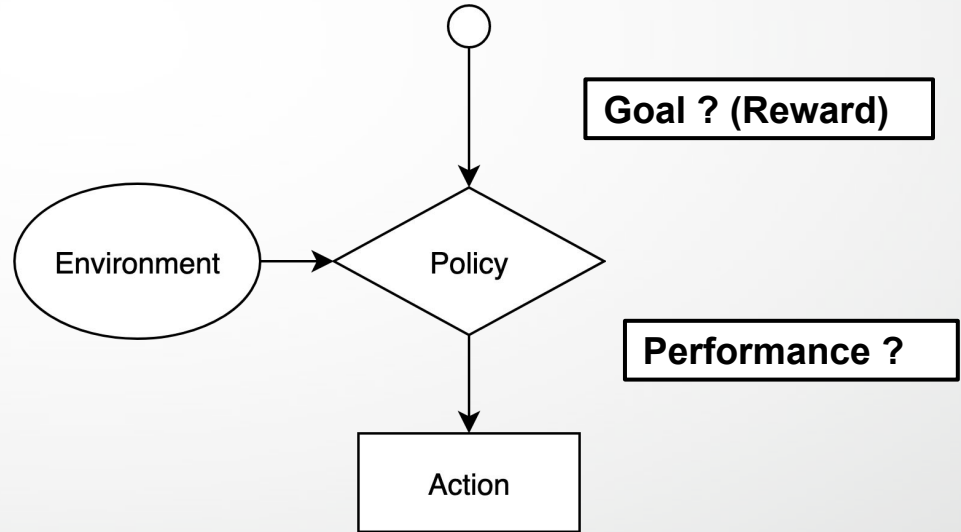


How to design / implement a Breakout AI ?



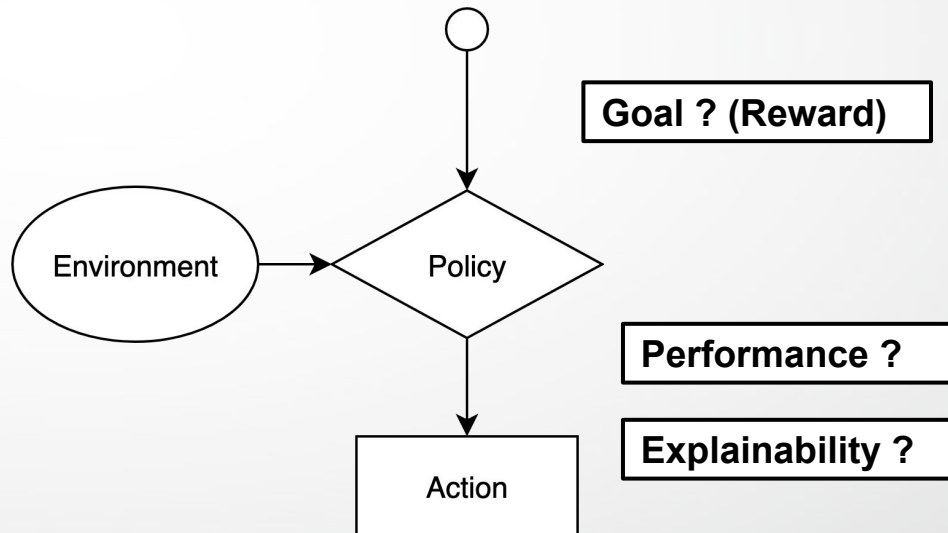
DeepMind : DQN Breakout

How to design / implement a Breakout AI ?



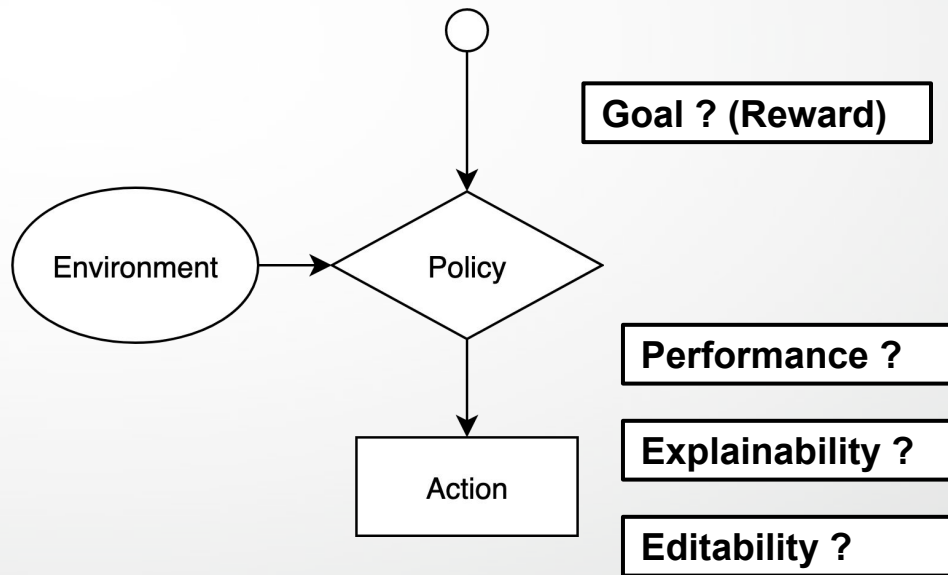
DeepMind : DQN Breakout

How to design / implement a Breakout AI ?



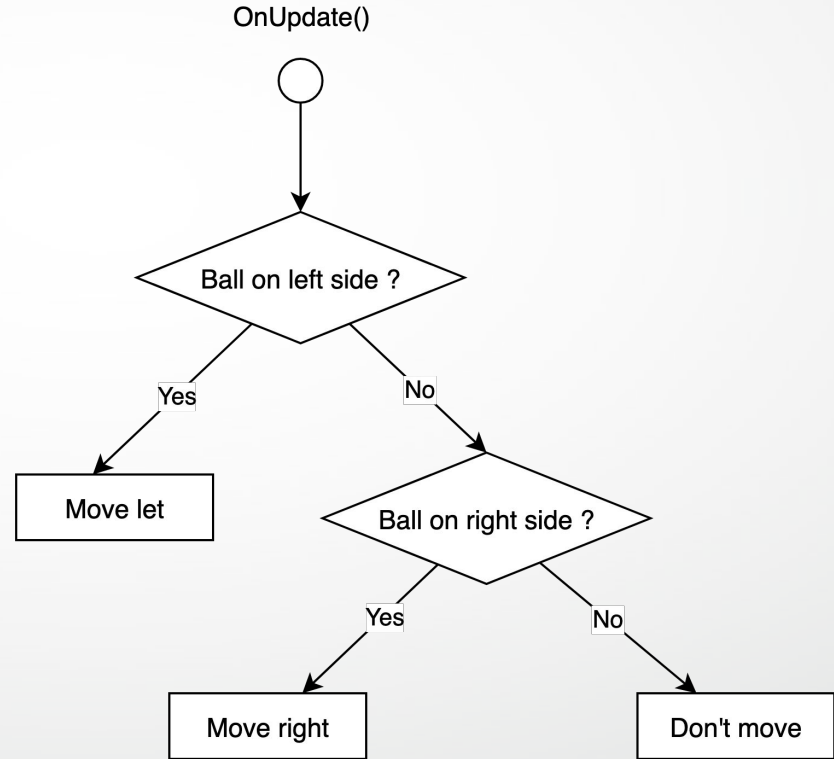
DeepMind : DQN Breakout

How to design / implement a Breakout AI ?



DeepMind : DQN Breakout

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 ?

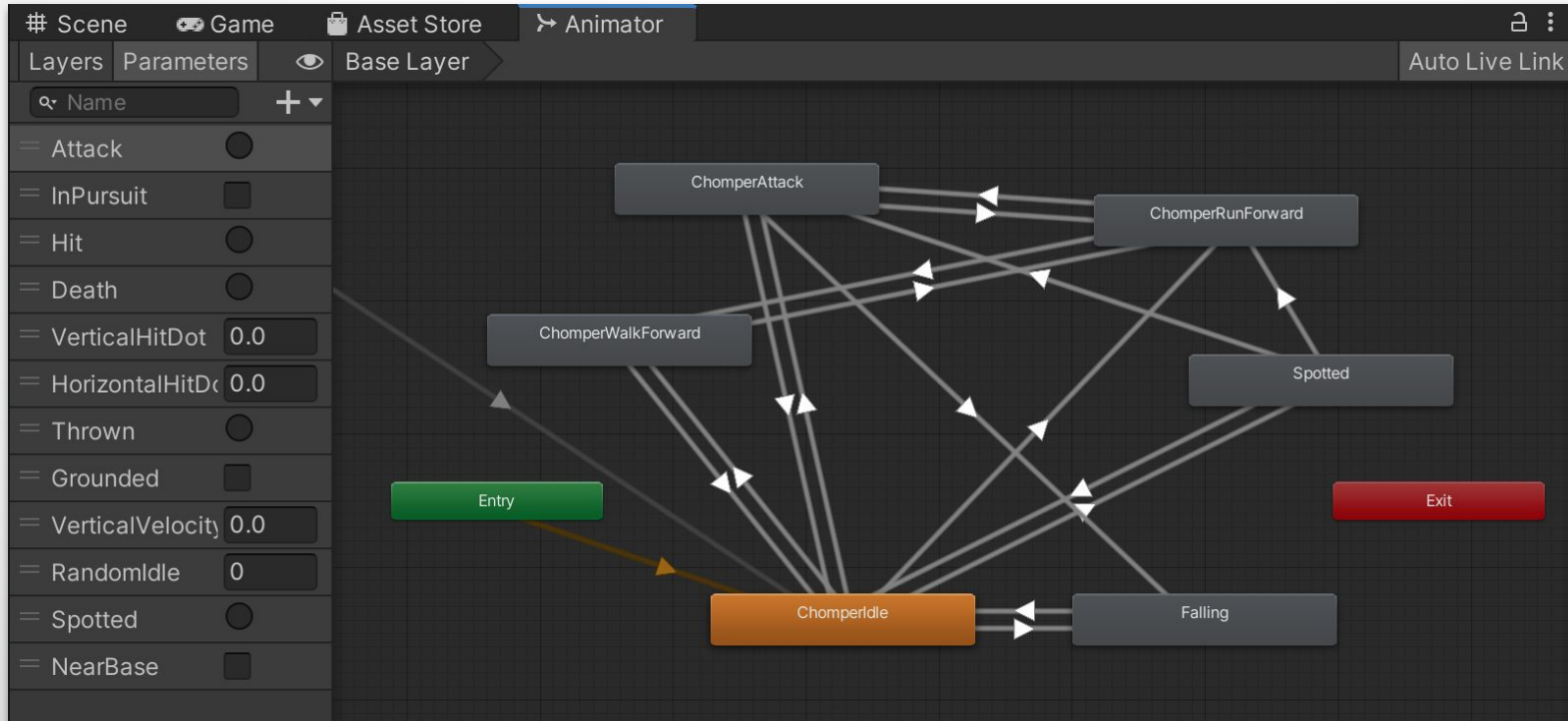
Finite state machines



Try to draw the FSM ?



Animator controller





State machine behaviours

The image shows a Unity State Machine Behaviours Inspector window. On the left, a state machine diagram is visible with states: Ledge Hang, Falling, Grounded, and Die. Transitions are shown with arrows: Grounded to Ledge Hang, Falling to Ledge Hang, Falling to Grounded, and Grounded to Die. The 'Grounded' state is highlighted in orange. On the right, the Inspector panel shows the 'Grounded' state selected. It includes a 'Tag' field, 'Speed' (1), 'Motion' (Blend Tree), 'Foot IK' (unchecked), 'Write Defaults' (checked), and 'Mirror' (unchecked). Below these are 'Transitions' with columns for 'Solo' and 'Mute'. Three transitions are listed: 'Grounded -> Die', 'Grounded -> Jumping', and 'Grounded -> Falling'. At the bottom, a 'Ground Detection (Script)' behaviour is added, with the script 'GroundDetection' selected. An 'Add Behaviour' button is at the bottom.

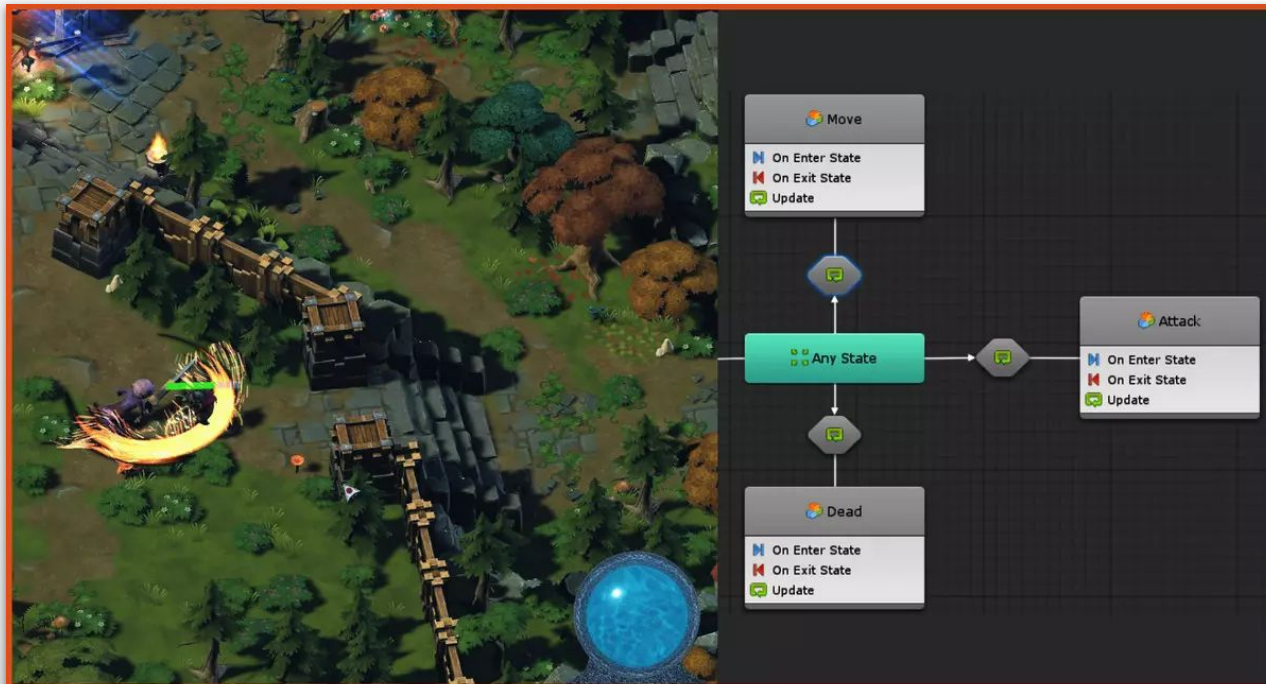


DEMO: 3D Game Kit

- 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 ?

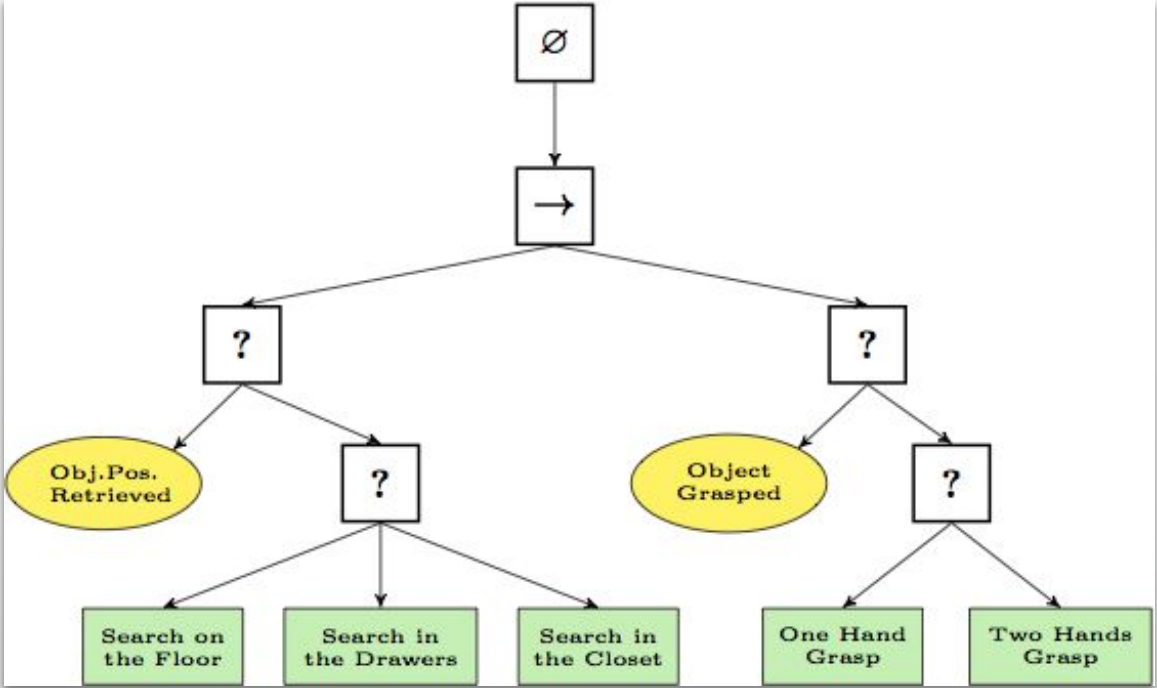
Try to draw the HFSM ?



DEMO: 3D Game Kit

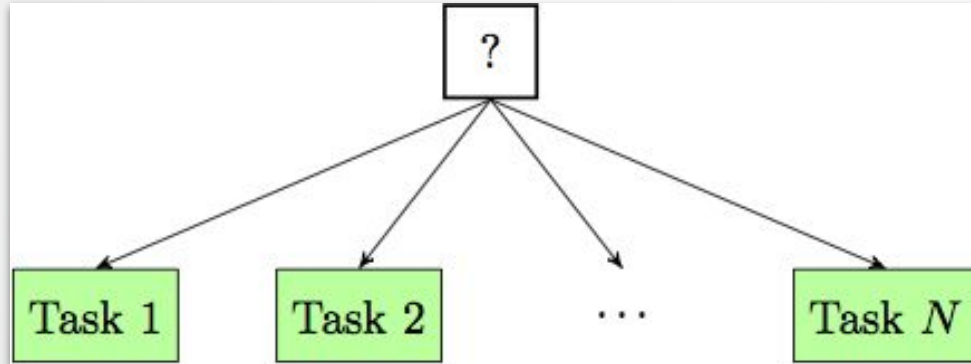
- Steps :
 - Open scene “**Level2**”
 - Show the Animator of “**Ellen**”
 - Explain the **Hierarchical state machines**
 - Explain the “**Any**” state

Behaviour trees

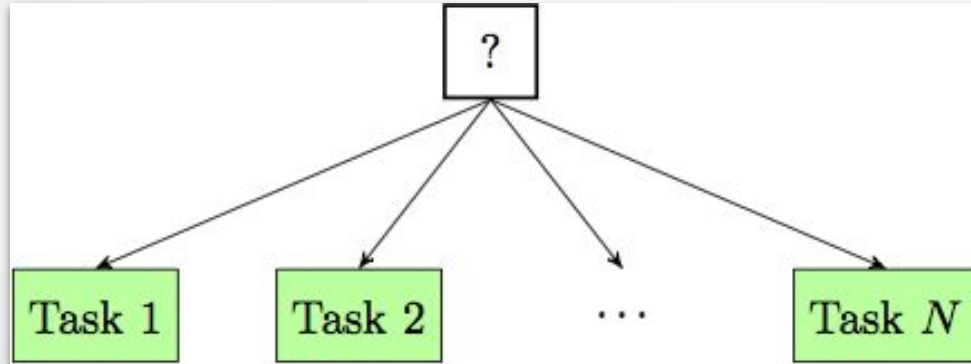


Wikipedia

Selector (fallback) nodes

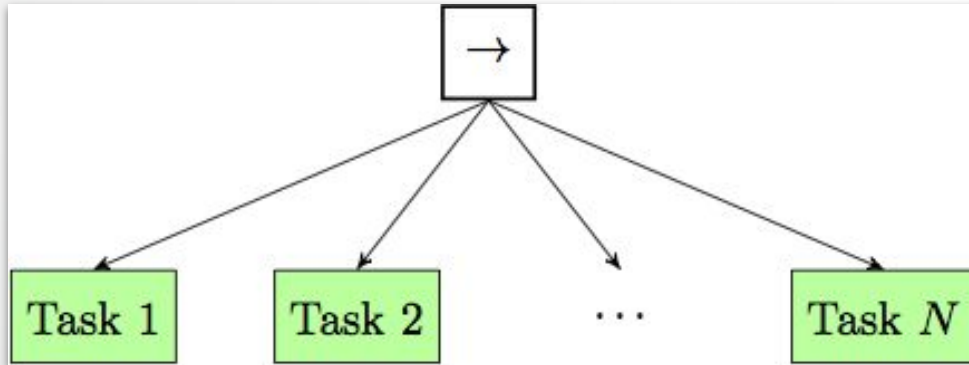


Selector (fallback) nodes



```
1 for i from 1 to n do  
2   childstatus ← Tick(child(i))  
3   if childstatus = running  
4     return running  
5   else if childstatus = success  
6     return success  
7 end  
8 return failure
```

Sequence node



```
1 for i from 1 to n do  
2   childstatus ← Tick(child(i))  
3   if childstatus = running  
4     return running  
5   else if childstatus = failure  
6     return failure  
7 end  
8 return success
```




DEMO: 2D Game Kit

- Steps:
 - Open scene “**Zone 5**”
 - Show the animator of **Gunner**
 - Explain the state topology
 - Explain “**Missile 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

nodeCanvas v3.x

```
graph TD; START[START] --> Design[Design]; START --> Develop[Develop];
```

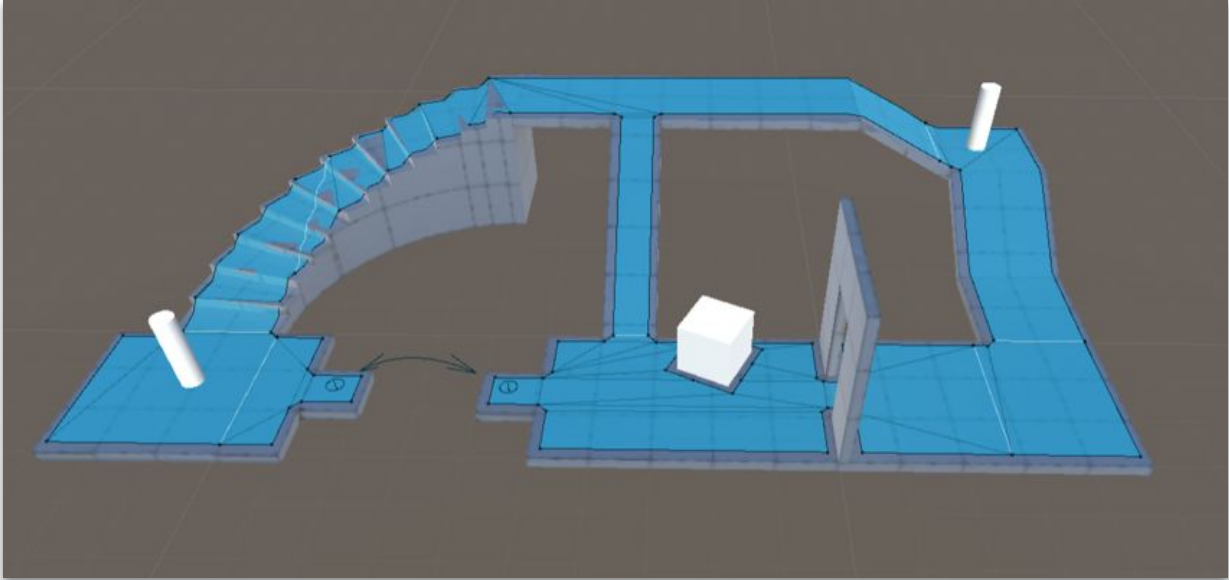
The Complete Framework

Behaviour Trees | State Machines | Dialogue Trees

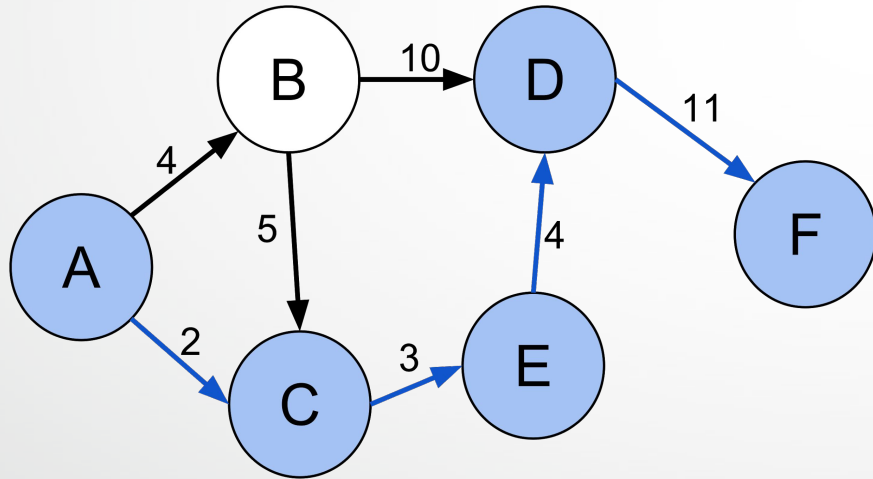
Navigation



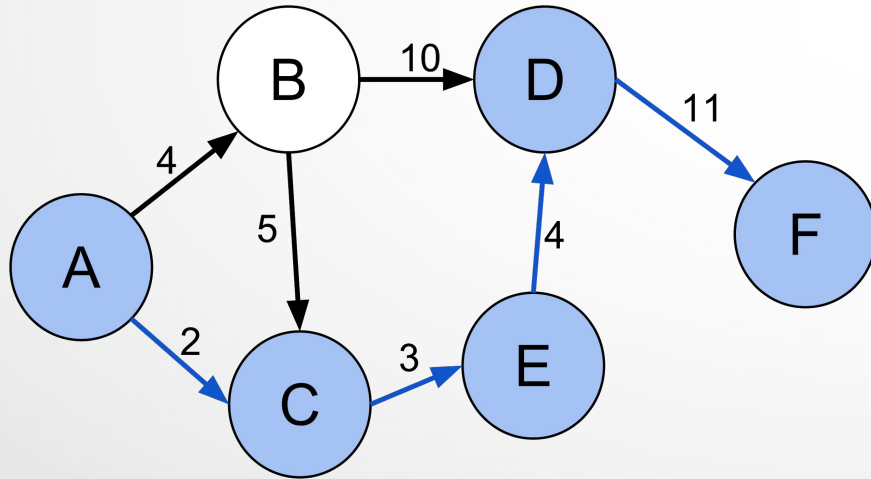
Navigation



Path finding

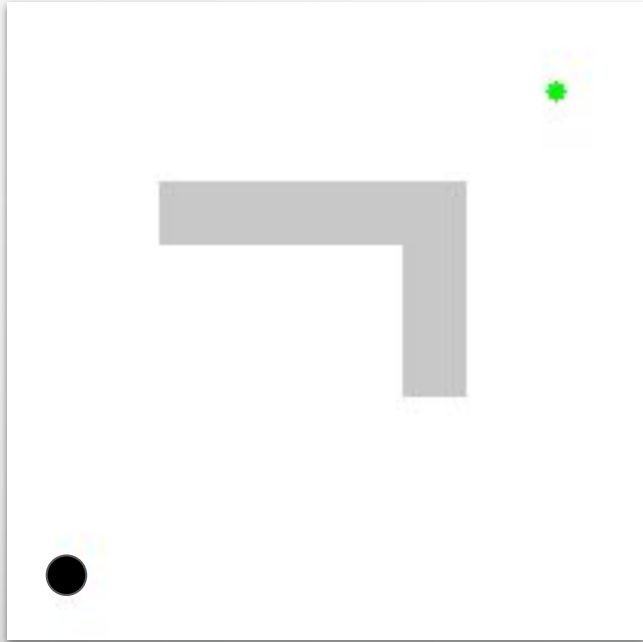


Path finding



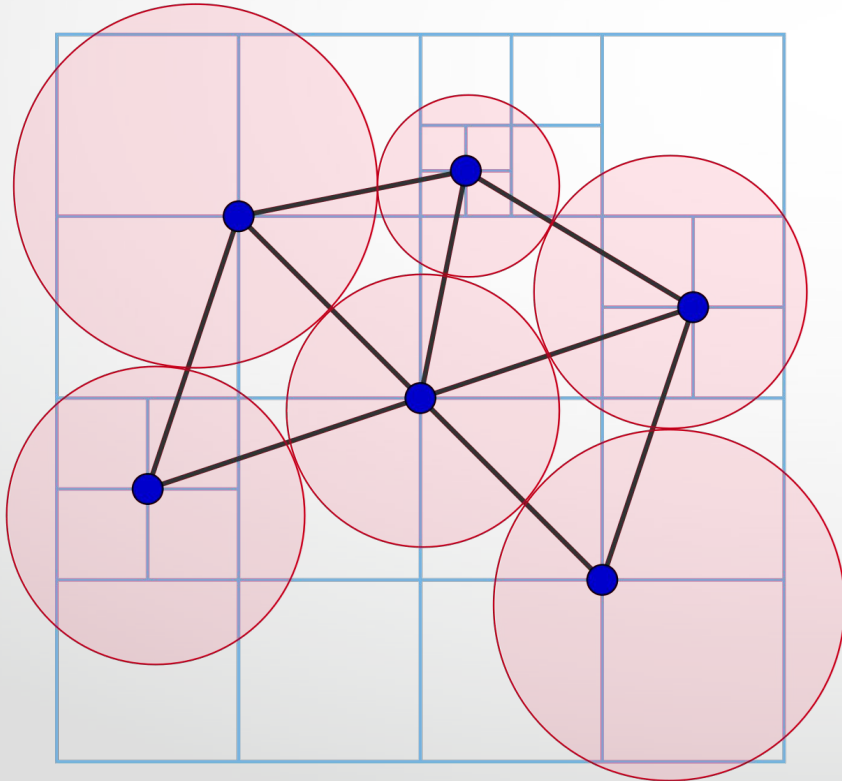
- Common algorithms :
 - BFS / DFS

Path finding



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

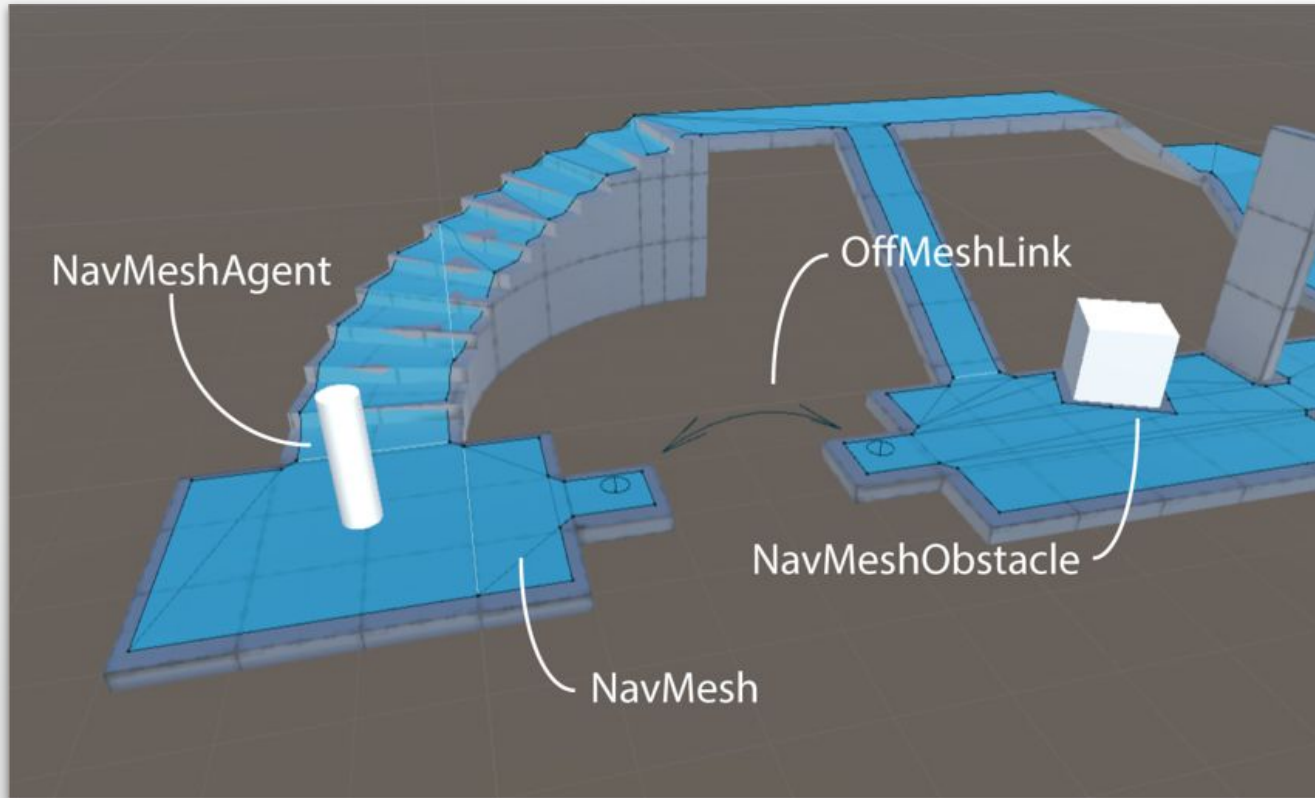
Hierarchical path finding



- 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.



Build a navmesh

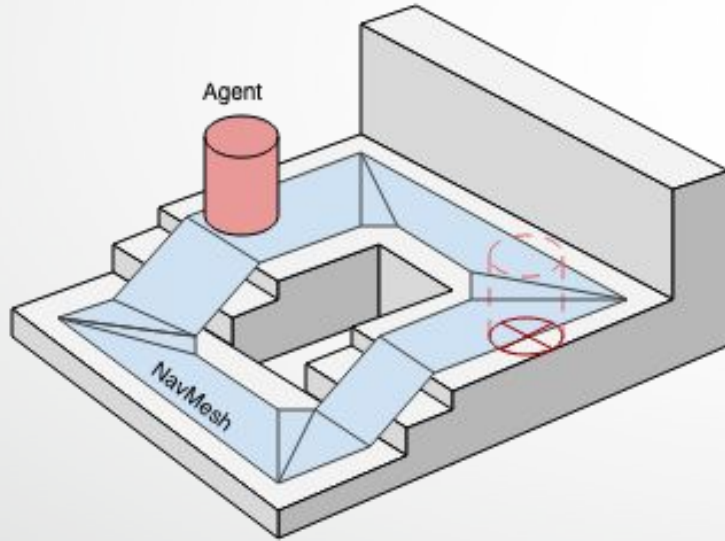
The screenshot shows the Unity interface with the Navigation window open. The Hierarchy panel on the left shows a scene named 'SampleScene*' with various objects including 'NavMeshedSceneGeometry', 'Upper Platform w/ramps', 'Starting Platform', 'Stairs', 'Door', 'Jumping Platform 1', 'Jumping Platform 2', 'Lower Platform 1', and 'Jumping Platform 2 (1)'. The Scene view shows a 3D environment with a blue navmesh overlaid on the ground. A white cube is placed on the navmesh. The Navigation window has tabs for 'Agents', 'Areas', 'Bake', and 'Object'. The 'Bake' tab is selected, and the 'Baked Agent Size' section shows a diagram of an agent with a radius of 0.15, a height of 1.5, and a max slope of 60 degrees. The 'Agent Radius' is set to 0.15, 'Agent Height' to 1.5, and 'Max Slope' to 60. The 'Step Height' is set to 0.53. The 'Generated Off Mesh Links' section has 'Drop Height' and 'Jump Distance' both set to 0. The 'Advanced' section has 'Clear' and 'Bake' buttons. A text box in the scene view says '3. Adjust the bake settings, under the Bake tab of the Navigation Window.' Another text box at the bottom of the scene view says '4. Click Bake to build the NavMesh.'

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

4. Click Bake to build the NavMesh.



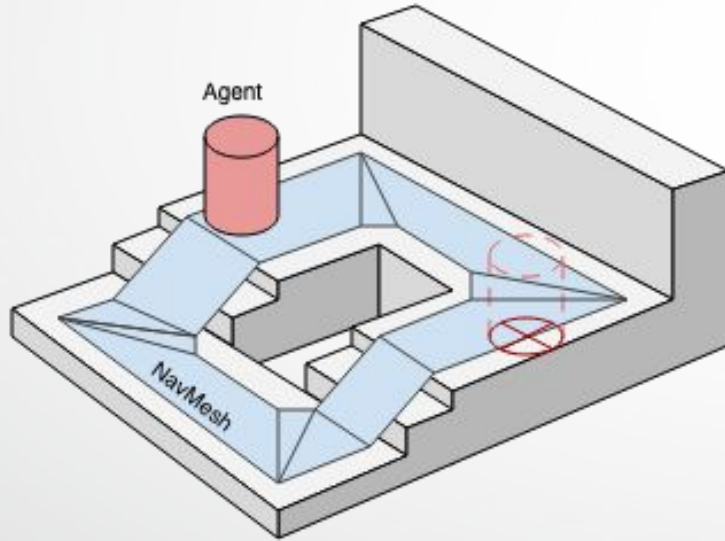
Walking



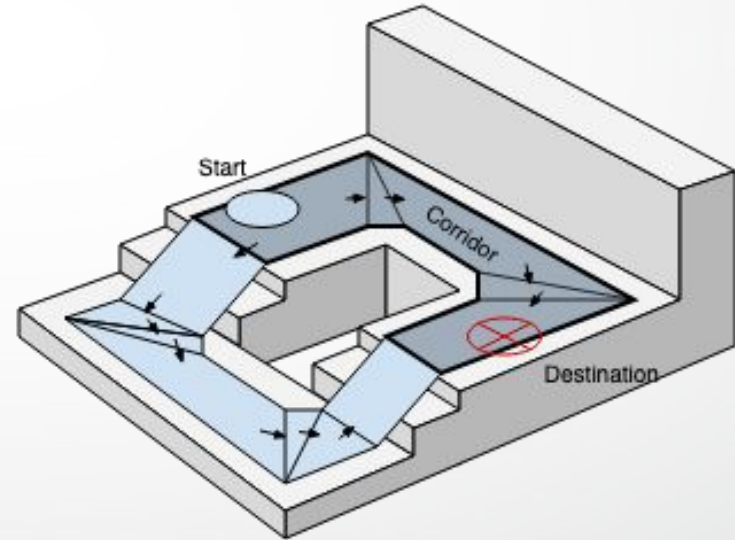
Walkable area



Walking



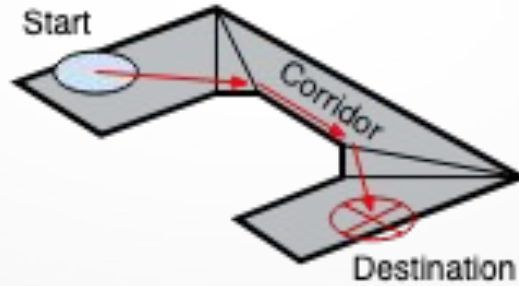
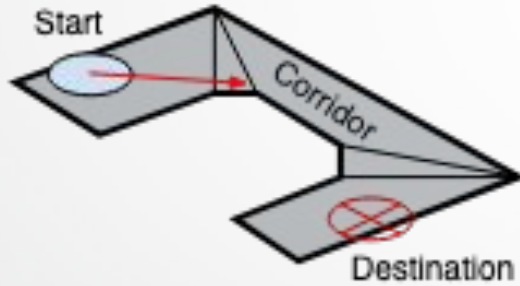
Walkable area



Finding paths



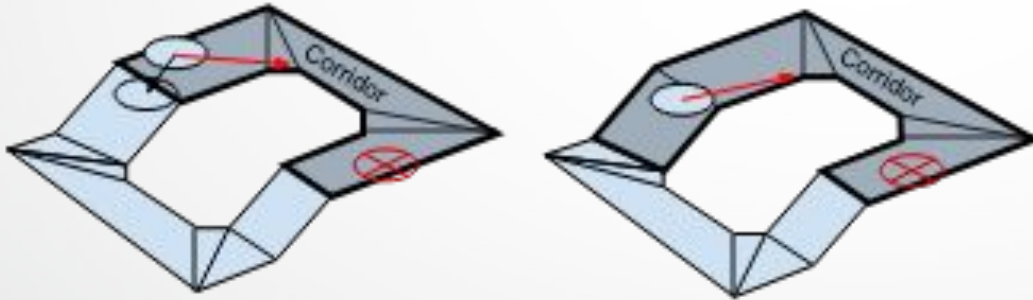
Walking



Global to local



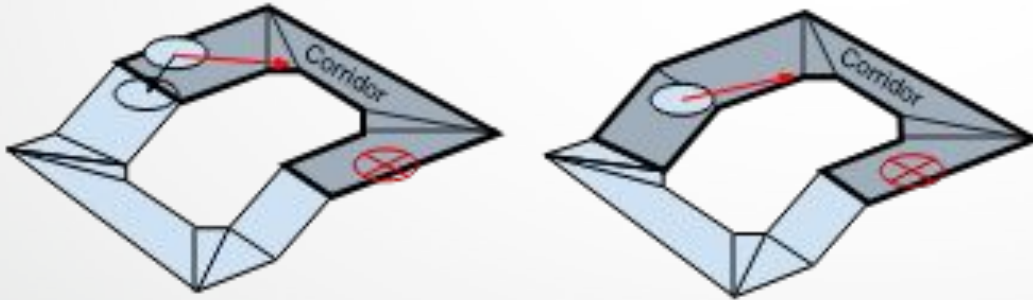
Avoid agents/ obstacles



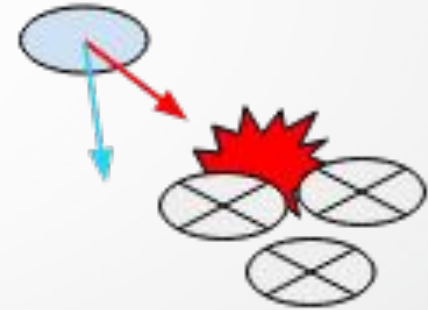
Local avoidance



Avoid agents/ obstacles



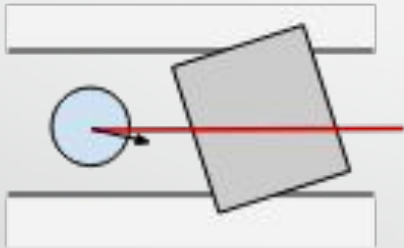
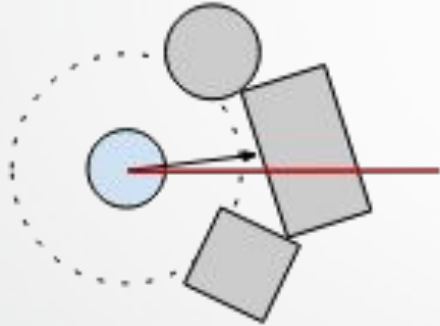
Local avoidance



reciprocal velocity obstacles

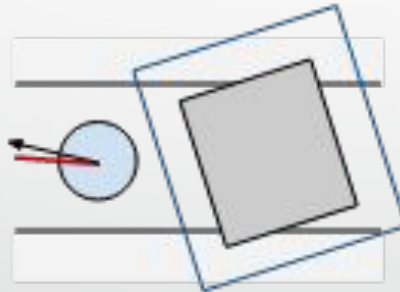
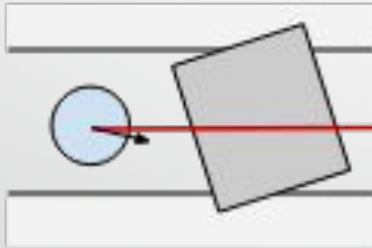
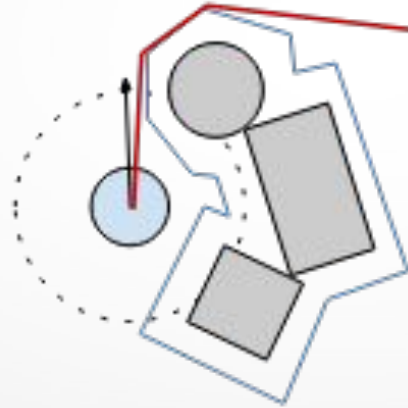
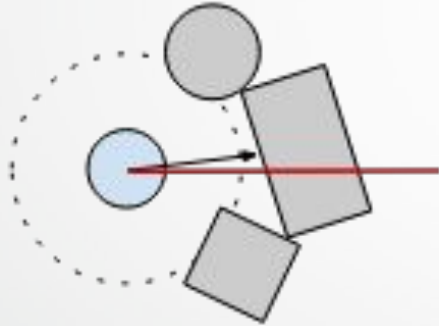


Avoid agents/ obstacles





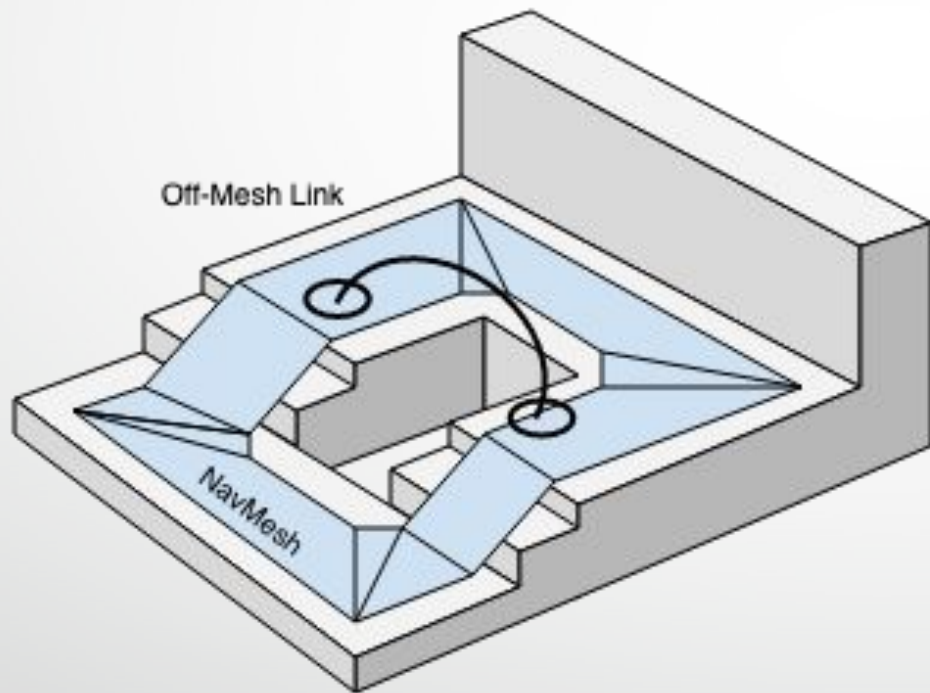
Avoid agents/ obstacles



Carving



Off-mesh link



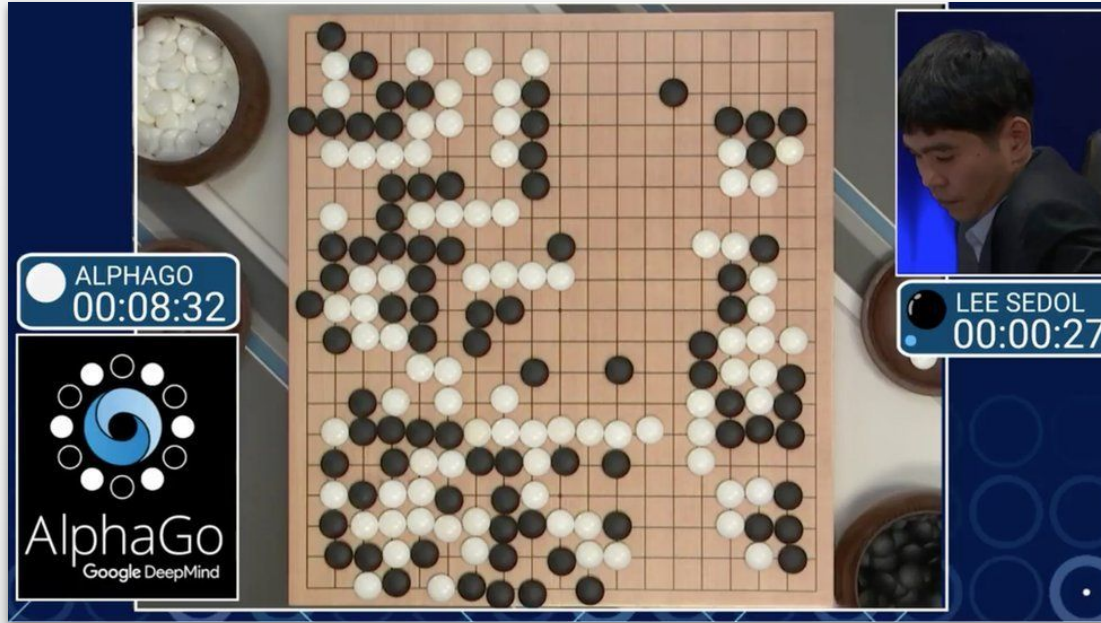
Jump / Climb / Teleport / ...



DEMO: 3D Game Kit

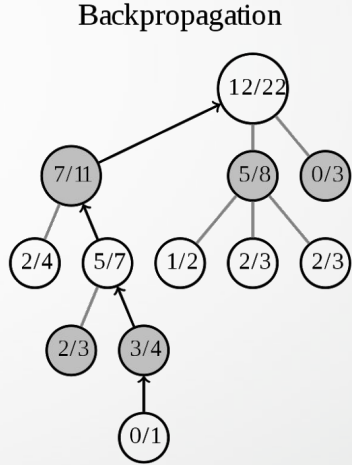
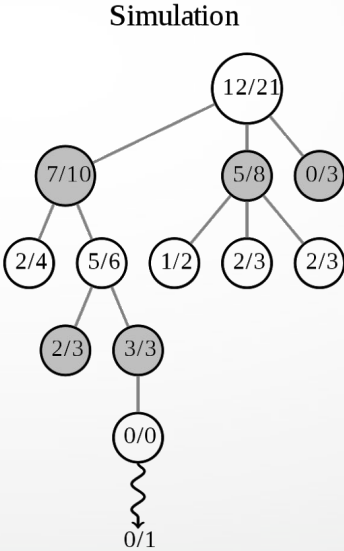
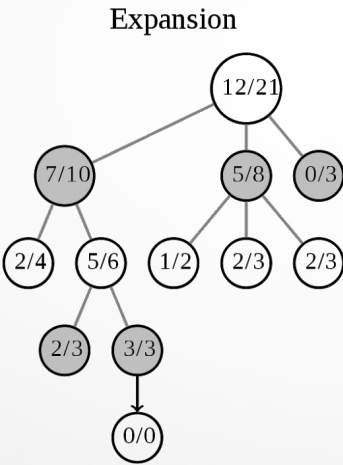
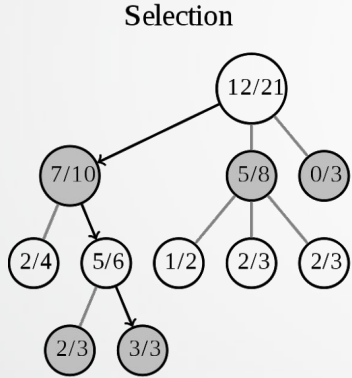
- Steps:
 - Create new **Scene**
 - Add “**Chomper**”
 - Select agentType “**Humanoid**”
 - Bake **Navmesh**

Planning



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

Monte Carlo tree search



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