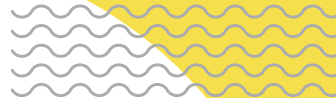




Facultatea de
Matematică și Informatică
Universitatea din București

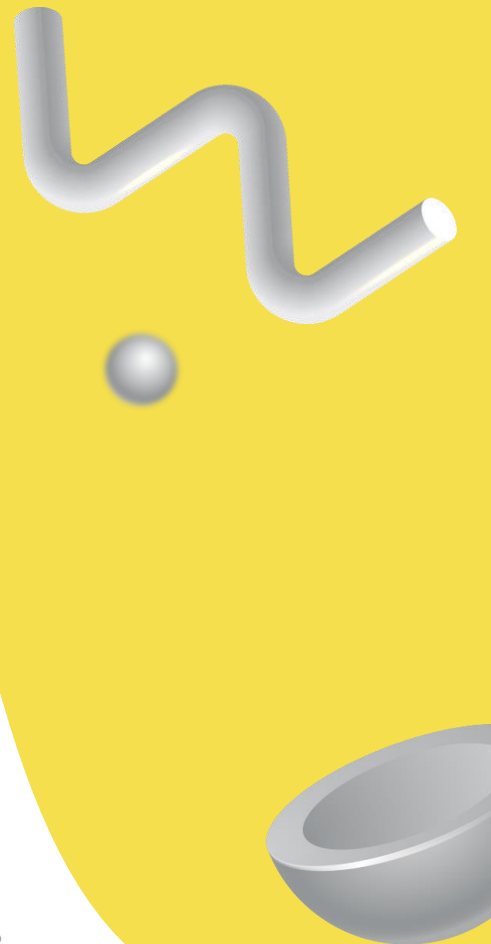


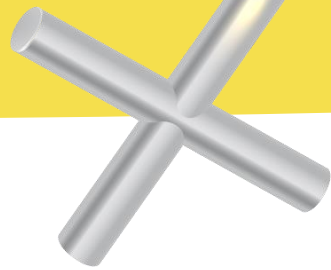
Programarea Aplicațiilor de Simulare

Inteligența Artificială în Jocuri

Curs 4

Pătrânjel David-George





Agenda Cursului

01

IA în jocuri video

02

FSM și
Behavior Trees

03

ML Agents

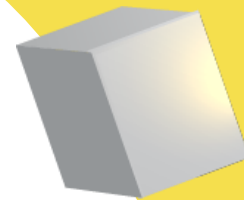
04

Integrarea modelelor
în Unity



01

IA în Jocuri Video





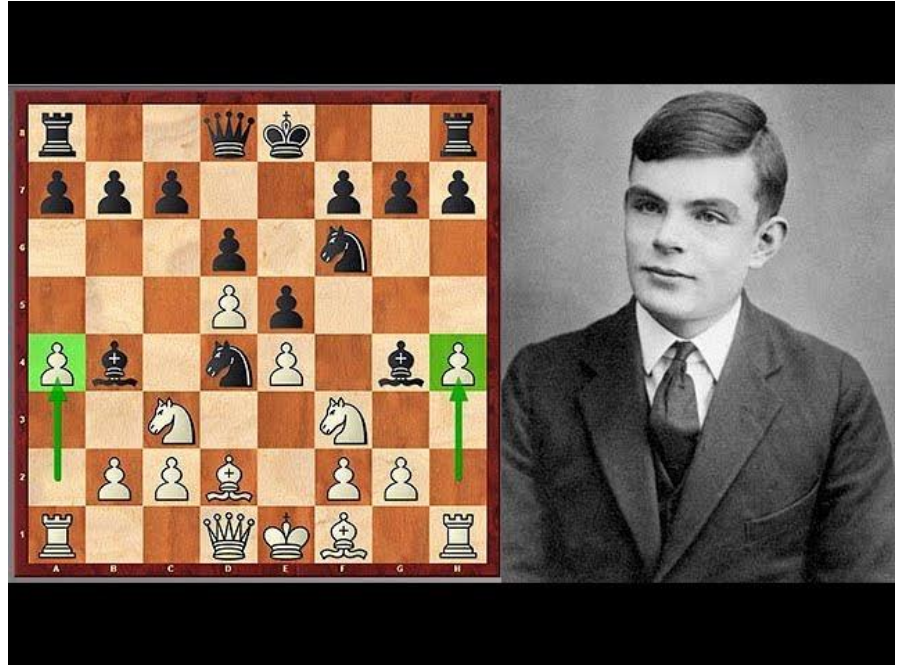
Inteligența artificială în jocuri

- Dorim să integrăm în jocuri sisteme ce prezintă o formă de inteligență.
- Modelele de AI pot:
 - Învăța noi concepte sau acțiuni.
 - Raționa și deduce concepte utile mediului în care se află.
 - Înțelege limbajul natural, imagini, voce etc.
- Sisteme care pot dobândi capacități de inteligență specifice omului.



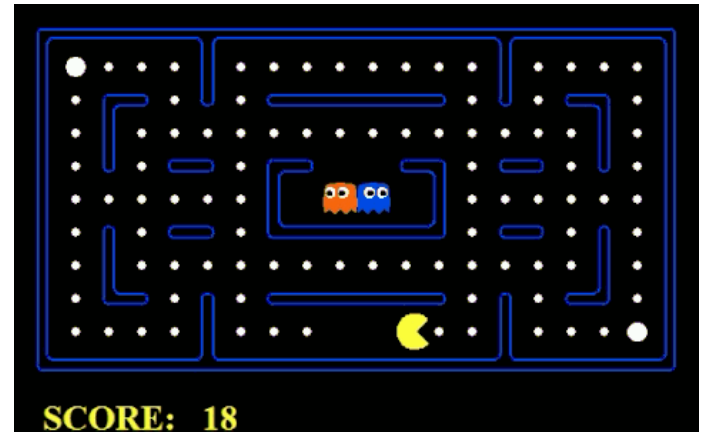
Reguli if-else

- Finite State Machines / Finite State Machines (FSM)
- Chess AI (1951): Alan Turing
- Pong (1972)



Pathfinding și luarea deciziilor

- Algoritmi de parcurgere.
- Arbori de decizie.
- Sisteme bazate pe reguli.
- Pac-Man (1980): Fantomele din joc folosesc algoritmi de mișcare.
- Prince of Persia (1989): Inamicii pot reacționa la atacuri.



Adaptive AI

- Behaviour Trees (BT)
- Finite State Machines Advanced
- Reactive AI
- Halo: Combat Evolved (2001): Inamicii cooperează, se acoperă și își schimbă tactica în mod dinamic.

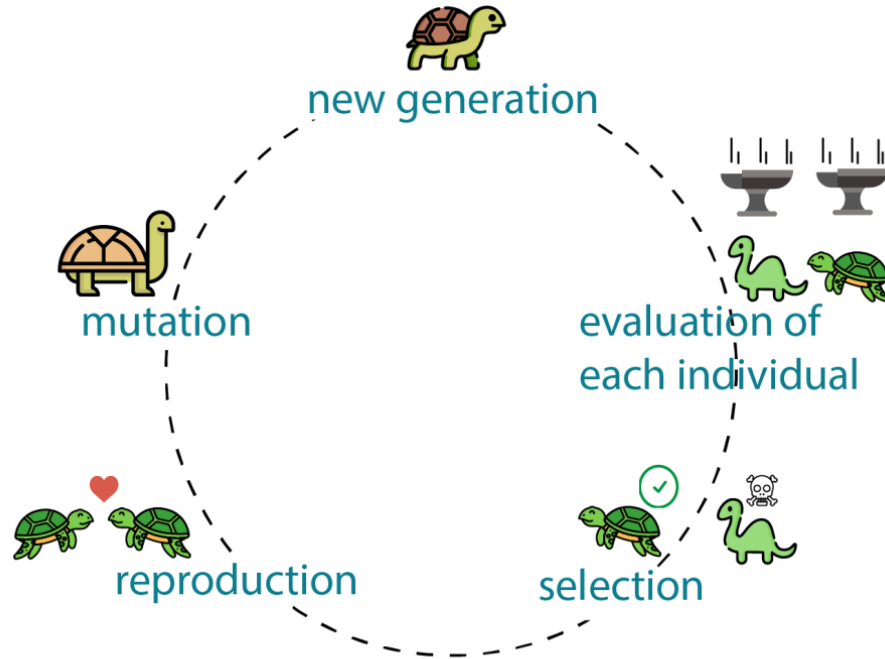


Learning & Deep Learning

- Învățare prin recompensă
- Algoritmi genetici
- Rețele neurale
- Dota 2 (2020): utilizare învățare prin recompensă pentru a învinge jucători umani.
- The Sims 4 (2014): personajele conduse de AI au personalități și emoții mai realiste



Algoritmi genetici



AI Generativ



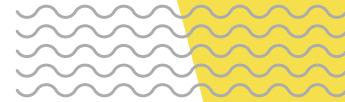
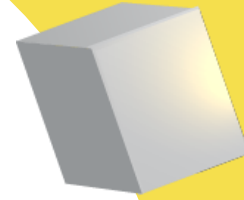
Procesarea de Limbaj Natural (NLP)



02

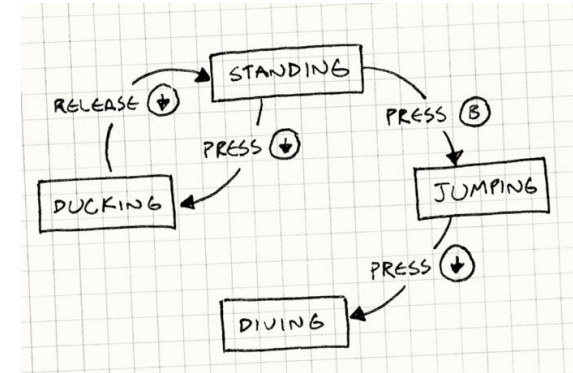
FSM şı

Behavior Trees

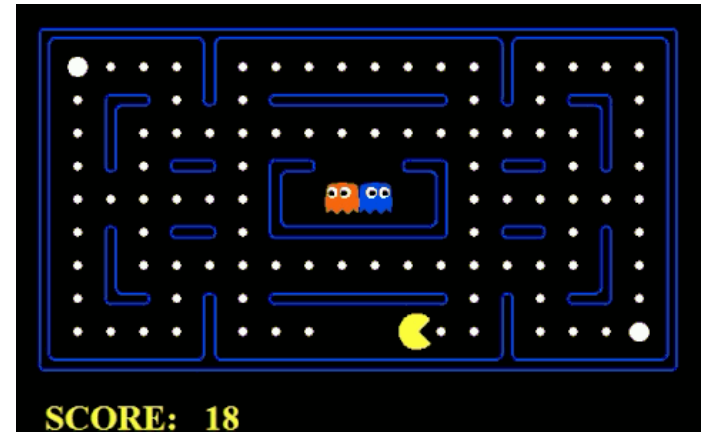
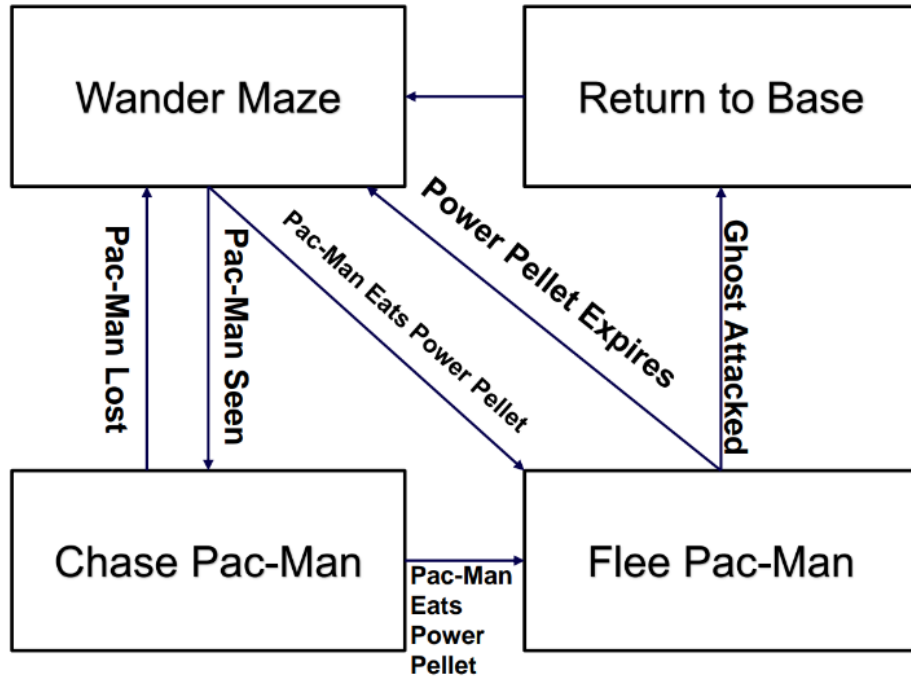


Automate cu stări finite (FSM)

- en. Finite State Machines (FSM)
- Model matematic de comportament compus din stări, tranziții și acțiuni.
- Trecerea de la o stare la alta se face prin tranziții determinate de condiții specifice.
- Acțiuni: de intrare, de ieșire, de tranziție, de intrare de date.
- Reprezentare sub formă de graf orientat sau folosind o diagramă de stări.



FSM - Exemplan





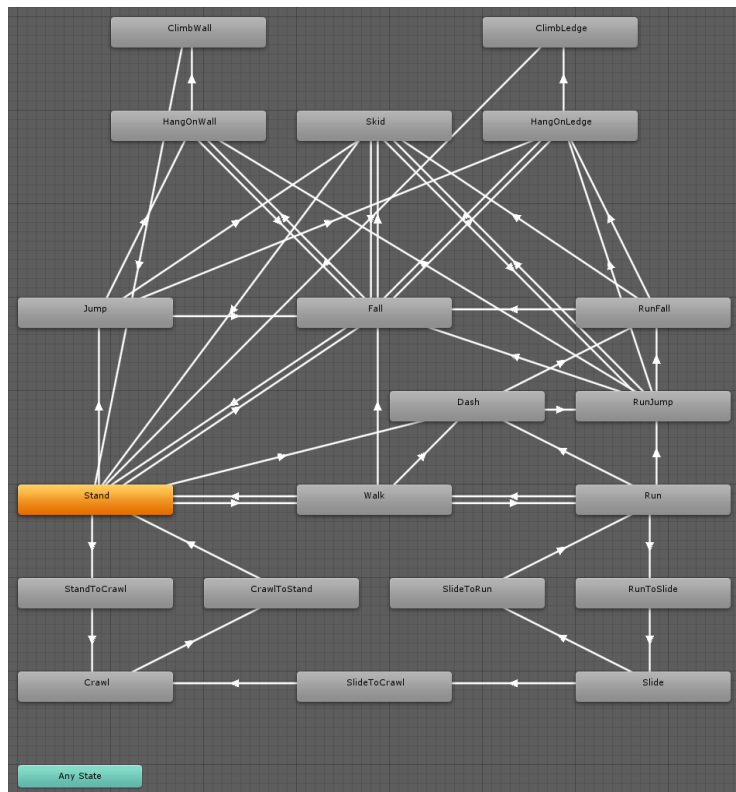
Automate cu stări finite (FSM)

La fiecare frame:

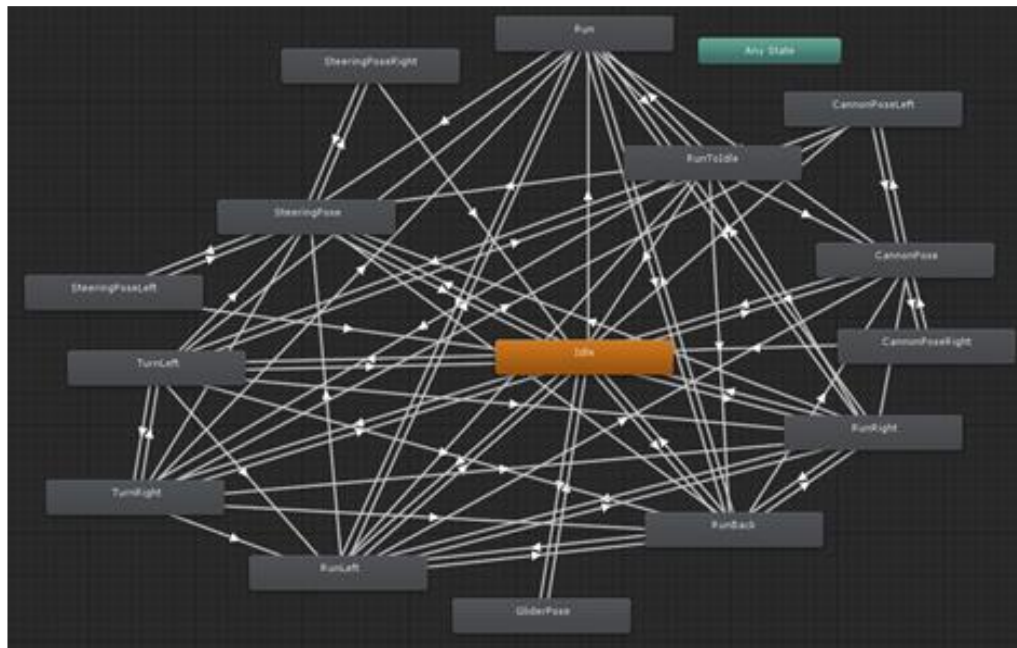
- Are loc o acțiune (a jucătorului/inamicului)
- Se verifică dacă are loc tranziția spre o nouă stare:
 - **Da** – Se schimbă starea
 - **Nu** – Starea rămâne aceeași

Aplicații:

- Realizarea animațiilor
- Gestionarea inputului
- Gestionarea stării jucătorului
- Modele simple de AI



Utilizarea FSM în Unity



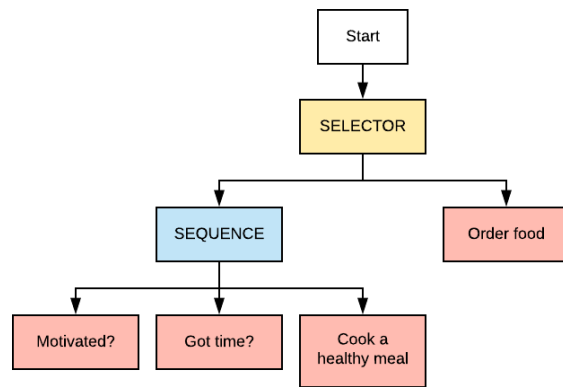
**Probleme
la scalabilitate!**



Utilizarea FSM în Unity

Arbori de comportament/Behavior Trees

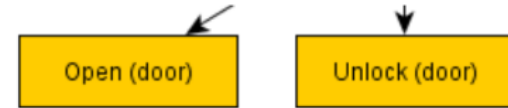
- Structuri ierarhice utilizate pentru a modela comportamente într-un mod modular și scalabil.
- Permite modificarea și extinderea cu ușurință a comportamentelor.
- Pot fi folosiți alături de arbori de decizie.
- Model de AI flexibil prin combinarea acțiunilor simple pentru **formarea acțiunilor complexe**.



Arbori de comportament - Elemente

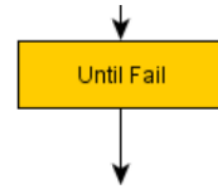
Noduri frunză:

- Reprezintă acțiunile principale
- Întoarce **Running**/**Success**/**Failure**



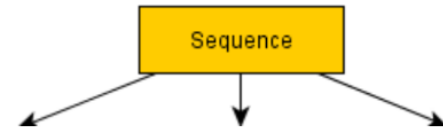
Noduri decorator:

- Are un singur nod fiu
- Întoarce **Running**/**Success**/**Failure** de la nodul fiu



Noduri composite:

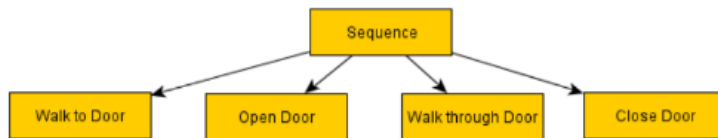
- Are unul sau mai multe noduri fiu
- Întoarce **Running până** la finalizarea nodului fiu



Noduri Composite

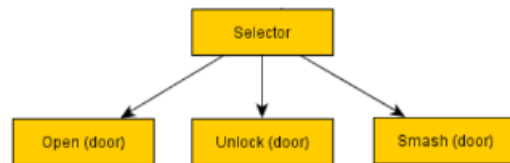
Noduri secvențiale (= AND):

- Execută, în ordine, toate nodurile fiu
- Întoarce dacă **Success** **toate** nodurile fiu au avut **Success**



Noduri de selecție (= OR):

- Execută, în ordine, toate nodurile fiu
- Întoarce dacă **Success** **orice** nod fiu au avut **Success**



Noduri secvențiale/de selecție aleatoare:

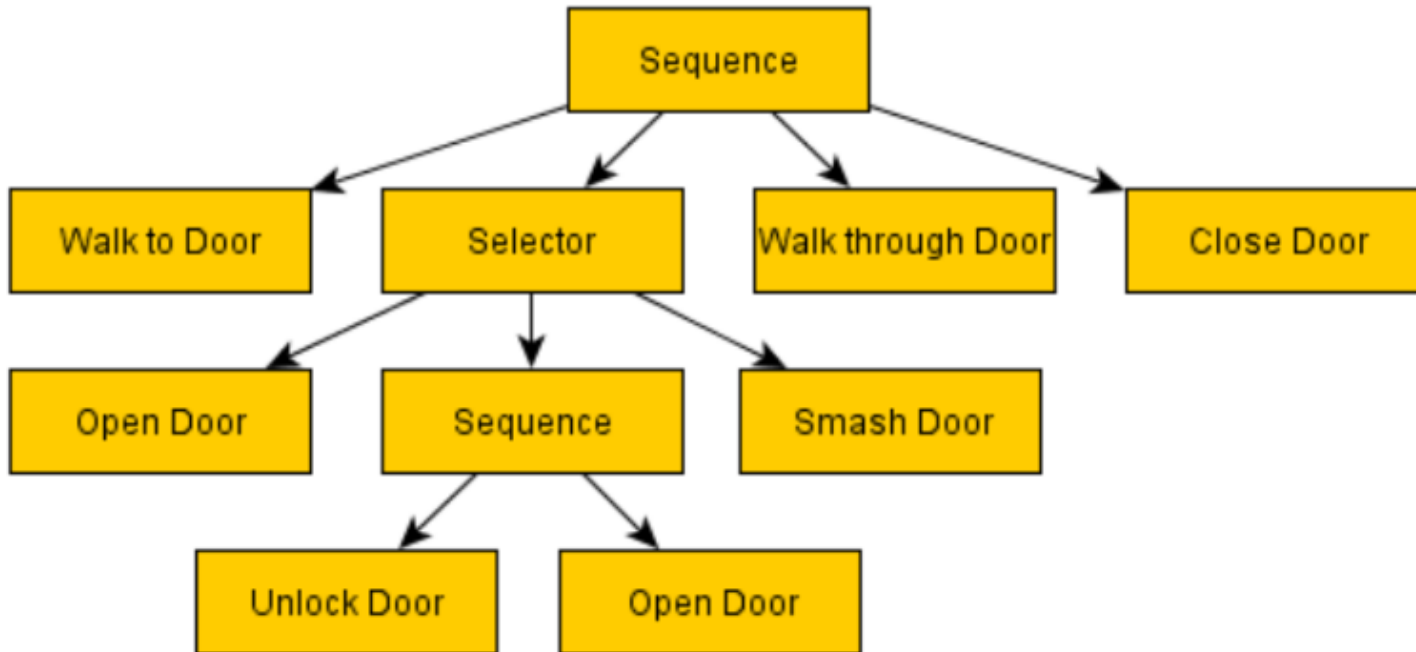
- Execută nodurile fiu în ordine aleatoare



Arbori de comportament (BT)

La fiecare frame:

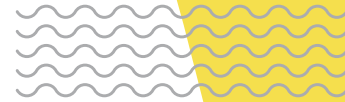
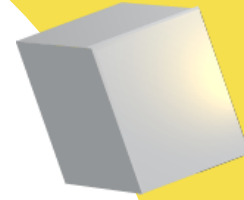
- Se parcurge arborele pornind de la nodul rădăcină
 - Parcurgere DFS (în adâncime).
 - Se verifică nodul curent:
 - **Running:**
 - Întoarce recursive starea până la nodul rădăcină
 - **Success/Failure:**
 - Întoarce rezultatul la nodul părinte unde va fi evaluată
 - În funcție de tipul de nod al părintelui se pot parcurge și următoarele noduri fiu.



Exemplu - Arbori de comportament

03

ML Agents



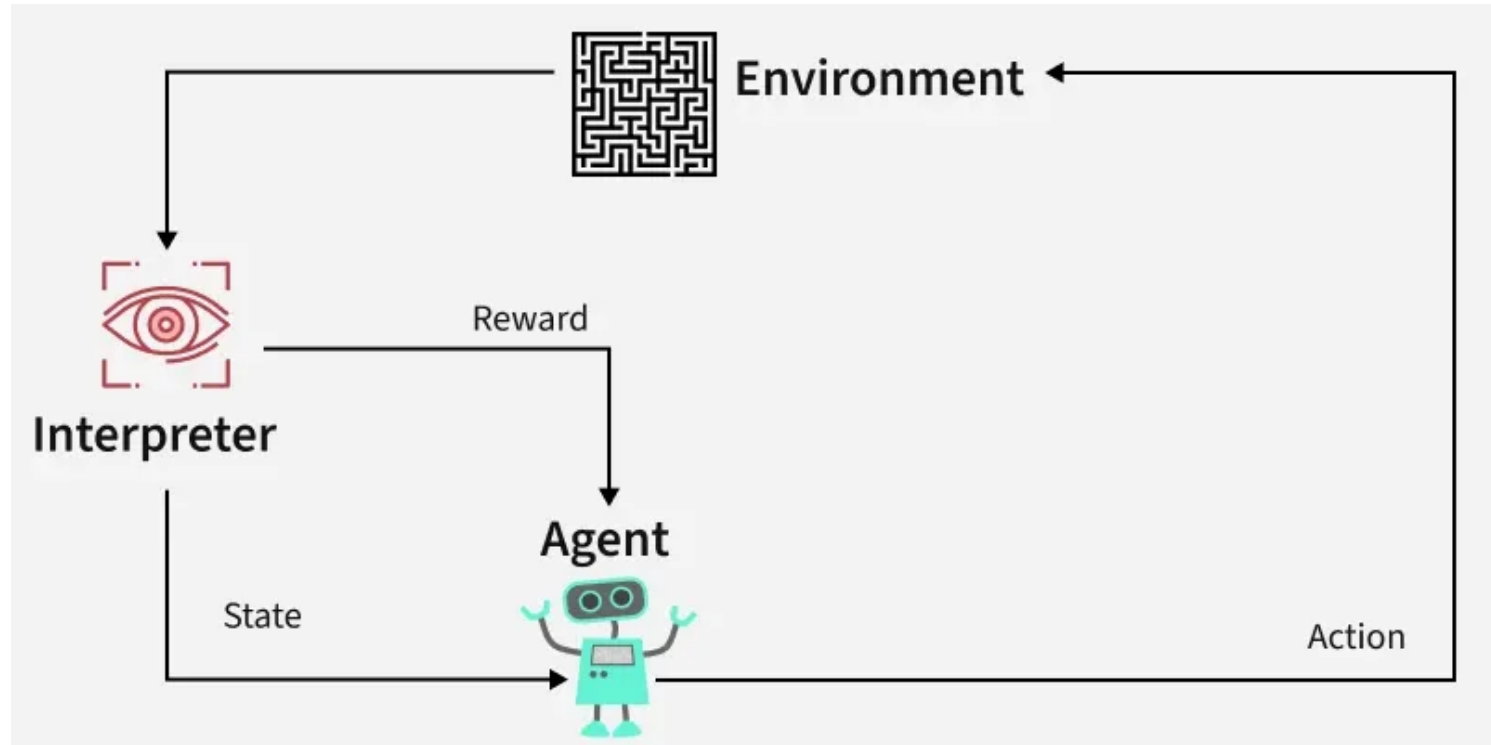
ML-Agents (Machine Learning Agents)

Un toolkit open-source dezvoltat de Unity pentru antrenarea agenților AI în medii simulate.

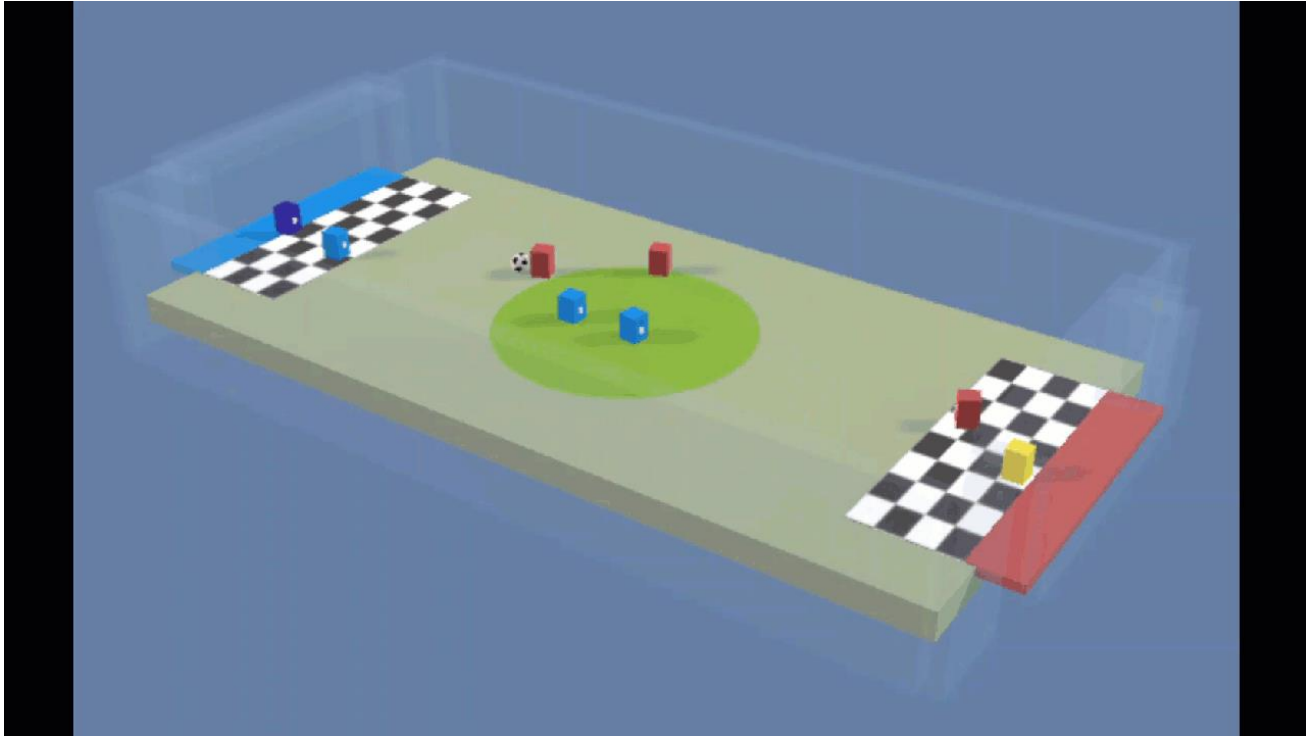
Util în antrenarea NPC-urilor, roboților, în simulări de conducere auto, optimizarea mișcărilor etc.

1. Învățare prin recompensă (en. Reinforcement Learning).
2. Învățare progresivă (en. Curriculum Learning).
3. Învățare prin imitare (en. Imitation Learning).



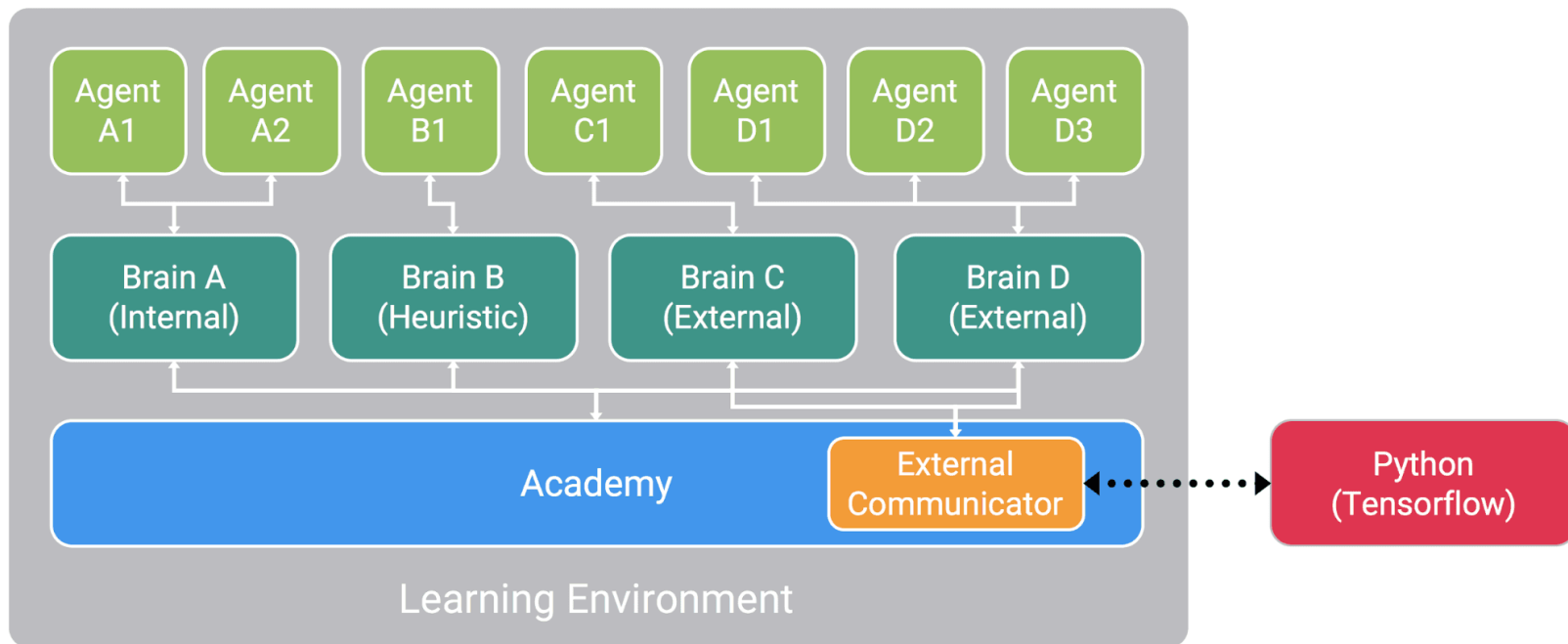


Învățare prin recompensă



Antrenarea Agenților pentru Jocuri Sportive

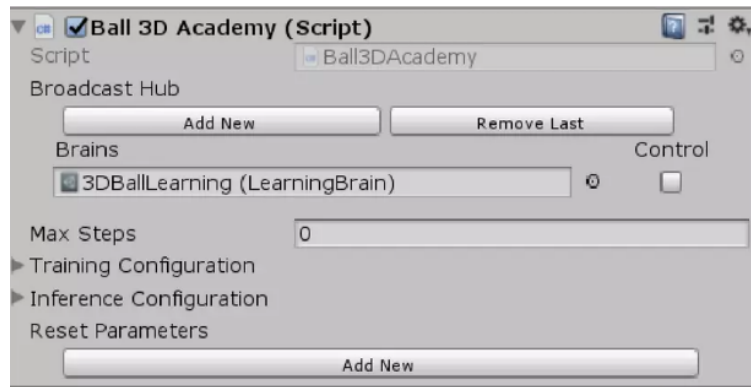
<https://christinemcleavey.com/world-cup-reinforcement-learning/>



►► Crearea unui mediu în Unity

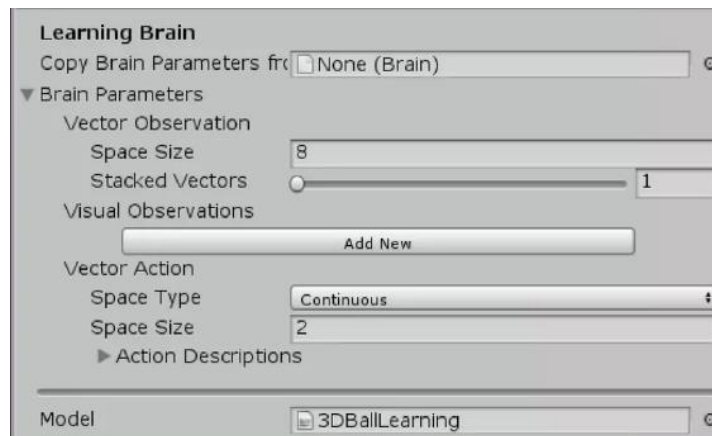
1. Academy (Academia)

- Asigură comunicarea cu API-ul de Python extern.
- Conține „creierii” utilizați în procesul de antrenare.
- Se poate configura antrenarea/integrarea modelului.



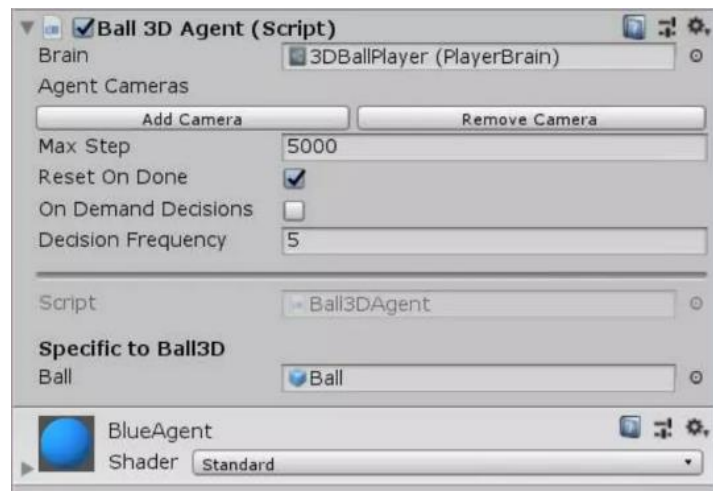
2. Brain (Creierul)

- Reprezintă modelul de AI.
- Primește observații de la agenți.
- Creierul este responsabil de luarea deciziilor și transmiterea acțiunilor către agenți.
- Acțiunile pot avea la bază: acțiunile utilizatorului, o euristică, sau acțiuni externe.



3. Agents (Agenți)

- Adună observații.
- Comunică cu „creierul”: execute acțiunile primite și transmite observațiile acumulate.
- Calculează recompensa unei acțiuni.
- **Observații, Acțiuni, Recompense!**
- Observații: diverși parametri, vectori (distanțe, viteze etc.), elemente vizuale (i.e. imagini).



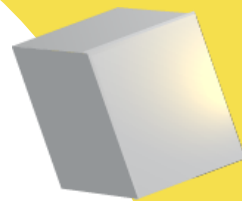


Antrenarea Agentului

<https://www.youtube.com/watch?v=-QlbDC6wdb8&t=1s>

04

Integrarea modelelor în Unity





1. Integrare Locală

- Integrarea directă a modelului de AI în aplicații realizate în Unity, fără necesitatea unei conexiuni la internet/rețea.
- Exemple:
 - TensorFlowSharp (pentru Unity și aplicații .NET)
 - ML Agents
 - Barracuda
 - ONNX Runtime





2. Integrare prin Cloud

- Integrarea indirectă a modelului de AI în aplicații realizate în Unity, fiind necesară conexiune la internet/rețea.
- **Necesită costuri adiționale!**
- Exemple:
 - **Construirea propriului API**
 - API pentru modele tip LLM (OpenAI API/ChatGPT/Claude)
 - Platforme de deploy pentru modele AI (Google Vertex AI/Azure AI)



Unity - Comunicarea Web

- Clasa `UnityWebRequest` din pachetul `UnityEngine.Networking` asigură comunicarea HTTP.
- Aplicația din Unity acționează drept client, comunicând cu serverul.
- Folosind o corutină, se pot utiliza metode:
 - `UnityWebRequest Get(string uri);`
 - `UnityWebRequest Post(string uri, string postData, string contentType);`
 - `UnityWebRequest Put(string uri, byte[] bodyData);`



Unity Web Requests

Integrarea modelelor de AI folosind API custom

Putem folosi framework-uri de Python pentru a construi un API ce are rute specifice de integrare/antrenare a modelului (FlaskAPI, FastAPI, Bottle, Django, Sanic etc.)

```
app.py x
1  from flask import Flask, request
2  from transformers import pipeline
3  /
4  app = Flask(__name__)
5  sentiment_analysis_model = pipeline("sentiment-analysis")
6  /unity_ml
7  @app.route(rule: '/unity_ml', methods=['GET'])
8  def unity_ml(): # put application's code here
9      text = request.args.get("text")
10     result = sentiment_analysis_model(text)
11     print(result)
12     return result
13
14  if __name__ == '__main__':
15     app.run()
```

FlaskAPI și HuggingFace

Resurse

L. Rokach and O. Maimon, “Decision Trees,” in Springer eBooks, 2006, pp. 165–192. doi: 10.1007/0-387-25465-x_9.

Ludic Worlds, “Get Started with ML-Agents in Unity - Part 1: Setup & Installation,” YouTube. Nov 5, 2024. [Online]. Available: <https://www.youtube.com/watch?v=bT3SV1SLqHA>

Unity-Technologies/ml-agents: <https://github.com/Unity-Technologies/ml-agents>

Unreal Engine 5 - Learning Agents Introduction (5.3):
<https://dev.epicgames.com/community/learning/tutorials/8OWY/unreal-engine-learning-agents-introduction-5-3>

Unity Documentation - UnityWebRequest:
<https://docs.unity3d.com/6000.0/Documentation/ScriptReference/Networking.UnityWebRequest.html>

Cursuri - „Algoritmi pentru Logica Jocurilor Video” – UPB, ACS, GMRV

