

РОЗРОБКА ПІДСИСТЕМИ ІГРОВОГО ШТУЧНОГО ІНТЕЛЕКТУ

Текст слайда

ВИКОНАВ: ВЕЛІГУРСЬКИЙ ОЛЕКСАНДР

ШТУЧНИЙ ІНТЕЛЕКТ У ІГРОВІЙ ІНДУСТРІЇ

- Відчуття: агент отримує або знаходить інформацію про об'єкти навколо нього, які можуть впливати на його стан.
- Мислення: агент приймає рішення, як діяти у тій чи іншій ситуації, залежно від стану інших об'єктів у ігровому світі.
- Дії: агент виконує дії для реалізації попереднього рішення. Ситуація міняється після проведених дій, після цього цикл повторюється.

ШТУЧНИЙ ІНТЕЛЕКТ У ІГРОВІЙ ІНДУСТРІЇ

Хороший ігровий штучний інтелект повинен мати такі властивості:

- Взаємодіє з ігровими системами.
- Реагує на дії гравця.
- Передбачуваний.
- Дає гравцю трохи більше можливостей.
- Агенти не повинні знаходити найкращий підхід проти людей, але бути реалістичними.

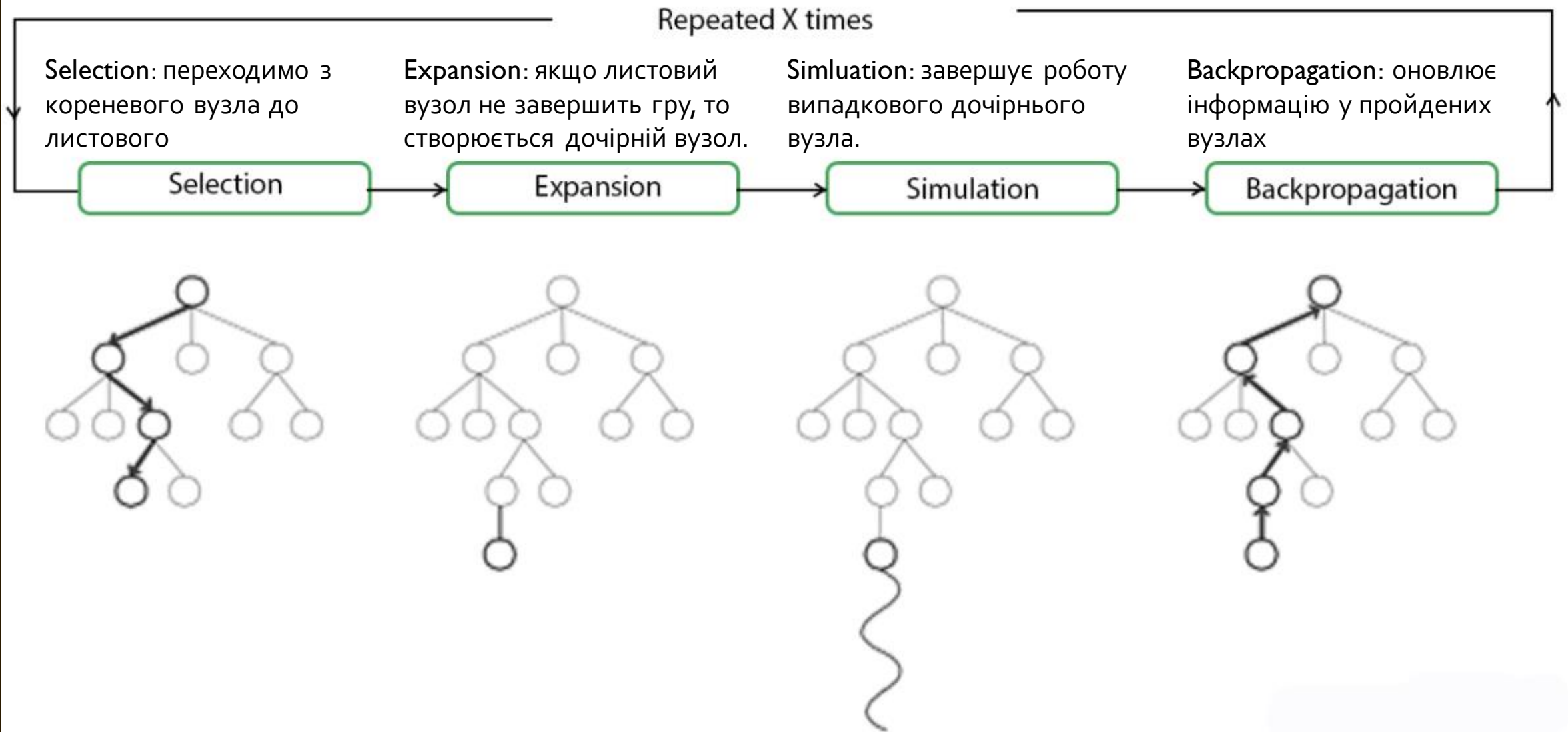
UPPER CONFIDENCE BOUNDS APPLIED TO TREES

$$\frac{w_i}{n_i} + c \sqrt{\frac{\ln N_i}{n_i}}$$

- $W(i)$ - кількість виграшних ситуацій
- $n(i)$ - кількість проходжень через вузол
- $N(i)$ - загальна кількість проходжень після i -го переміщення
- c – константне значення, яке обирається емпіричним шляхом (теоретично $= \sqrt{2}$)

```
private double Calculate_UCT(Node node)
{
    var result = (double)node.Wins / node.Visits + Math.Sqrt(ChoosenConstant * Math.Log(Visits) / node.Visits);
    return result;
}
```

ПОШУК ПО ДЕРЕВУ ЗА МОНТЕ-КАРЛО



ПОШУК ПО ДЕРЕВУ ЗА МОНТЕ-КАРЛО

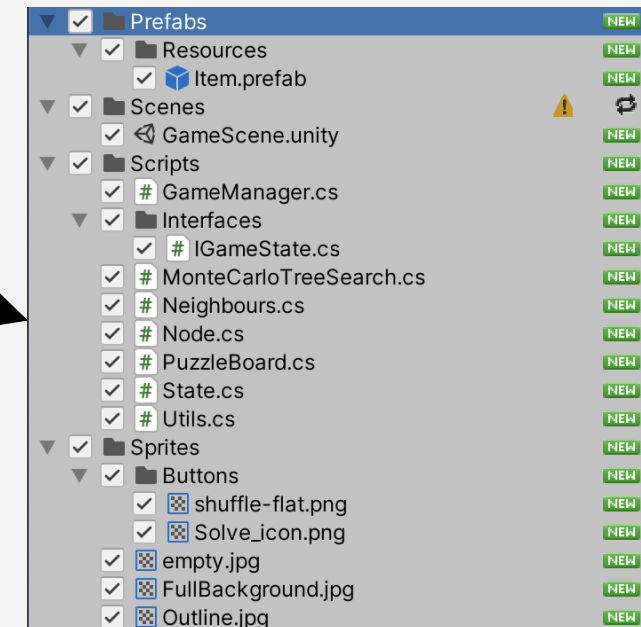
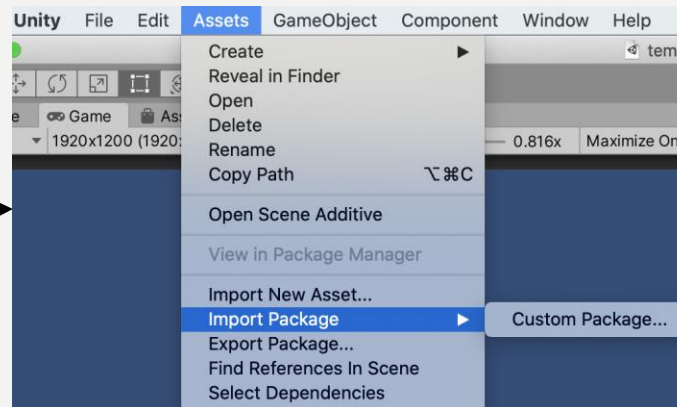
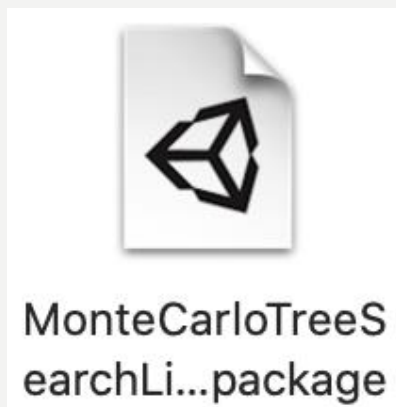
```
for (int i = 0; i < iterations; i++)  
{  
    Node newNode = rootNode;  
    var state = new State(rootState.Size);  
    newNode = SelectNextNode(newNode, state);  
    newNode = ExpandChildNodes(newNode, state);  
  
    Rollout(newNode, state);  
    Backpropagate(newNode);  
}
```

Раунди пошуку
повторюються до тих пір,
поки залишається час або
кількість ітерацій, які
відведені на прохід.

В кожному вузлі обирається хід, значення УСТ якого – найвище.

ВПРОВАДЖЕННЯ ПІДСИСТЕМИ

1) Завантажуємо MonteCarloTreeSearchLibrary.unitypackage та імпортуємо у проект





2) Бібліотека одразу містить у собі необхідні методи та демо приклад використання алгоритму.

ГРА "ПЯТНАШКИ"

Рахунок: 0

	1	7	2	4	
	9	5	3	8	
		6	10	15	
	13	14	12	11	

Рахунок: 0

7	3	8
1	4	
2	5	6



Рахунок: 22

1	2	3
4	5	6
7	8	

ГРА "ПЯТНАШКИ"

Рахунок: 0

1	2	4	11
	5	12	3
10	7	14	8
9	6	13	15

