

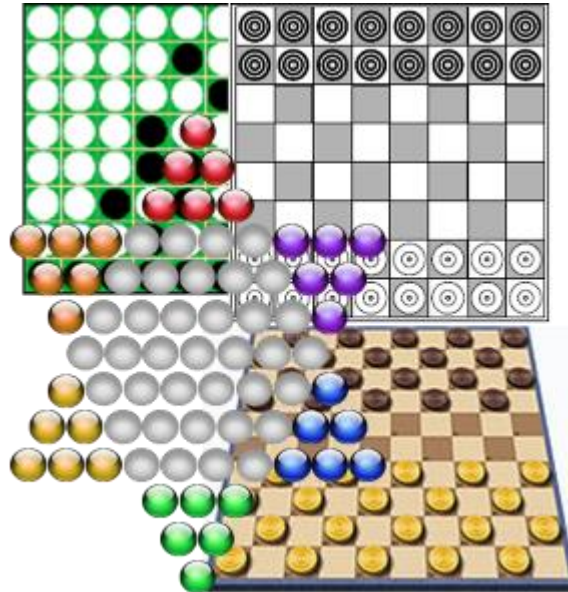
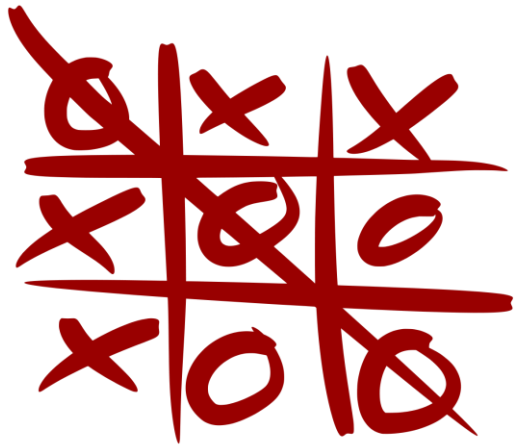
Metody sztucznej inteligencji w grach video

Maciej Świechowski
Seminarium z Metod Inteligencji Obliczeniowej [17.03.2021]

Rodzaje gier

Gry kombinatoryczne
Gry umysłowe

Gry komputerowe
Gry na konsole
Gry na komórki



Wykorzystanie gier w nauce



Założenie tej prezentacji



Metody do tworzenia AI jako zasadniczego elementu gry
twórcy gry autorami AI



Nie o używaniu wydanych gier wideo
jako środowiska testowego do eksperymentowania z AI
- oddzielny, obszerny temat

AI w nurcie badawczym vs. AI w gamedev

Zazwyczaj maksymalizacja siły gry	Realizowanie celów fabularnych i projektowych
Dążenie do poziomu "superhuman"	AI ma być fajne - nie za słabe, nie za silne, do pokonania, by utrzymać gracza we 'flow'
Najważniejsze jest to co AI robi : efekt w środowisku (modelu), w którym działa	Najważniejsze to co widać - przedstawienie
Podejście możliwie algorytmiczne, zautomatyzowane	Kontrola , testowanie
Jeżeli ML, to potrzeba danych treningowych	Łatwa modyfikacja w trakcie powstawania gry
Badanie i porównywanie algorytmów do podejmowania decyzji	Łatwa integracja z resztą gry
	Parametryzowalność (poziomy trudności, re-używanie AI)

AI w nurcie badawczym vs. AI w gamedev

Nurt samoskonalenia się programów	Względnie duża przewidywalność (wszelkie możliwości zaskoczenia sprawdzone przed wydaniem gry)
Względnie dużo czasu na obliczenia	Mało czasu na obliczenia
Sprzęt wysokiej wydajności	Sprzęt docelowego użytkownika gry; 1x "consumer-level" hardware
Open AI Five: 128K CPU; 256 P100 GPU; ~180 lat rozgrywek dziennie	Szybkie akcje: do 16ms na decyzje
Deep Blue: 259 miejsce na liście Top500 w 1997r.	Strategiczne akcje: zależnie od gry

O mnie

- Badania naukowe w grach (głównie *General Game Playing*)
- Doświadczenie z gamedev
- Obecnie w QED Software - projekty wokół gier
 - Grail** – silnik do AI dla branży gier
 - Sensei** – analityka dla graczy (casual i esports)
 - Fogs of War** – realizm botów w obliczu niepełnej informacji i losowości w grach
 - Tactical Troops: Anthracite Shift** – własna gra
 - Ancestors Legacy**: realizacja AI
 - [niejawna nazwa]**: realizacja AI



Obecne inicjatywy

- Wykorzystujące m.in. gry jako środowisko testowe:



Label in the Loop – aktywne uczenie, metody i narzędzia do tagowania i wzbogacania danych

Knowledge Pit – platforma konkursowa dla data science:



- Ogólnie w QED Software:
ponad 25 projektów związanych z AI/ML/Data Science

Nowy konkurs data science na knowledgePit.ai

start 30 kwietnia, zgłoszenia do 12 września

Analiza logów w grze Tactical Troops - Anthracite Shift

(gra turowa wydana przez QED Software w 2021)



przyrost
logów

100 GB

2 TB

screenshotów

Kto
wygra
???

50K

rozgrywek

Human
& AI

rodzaje graczy



Nagrody dla 3 rozwiązań

- nagrody finansowe
- możliwość napisania tematycznej publikacji na grudniową konferencję IEEE Big Data 2021
- Najlepsze rozwiązania posłużą do ulepszenia portalu, który analizuje rozgrywki graczy w Tactical Troops - Anthracite Shift



Rejestracja:
www.knowledgepit.ai

Pomocne linki

Tactical Troops: Anthracite Shift

<https://tacticaltroops.net/>

https://store.steampowered.com/app/1266890/Tactical_Troops_Anthracite_Shift/

Grail

<https://grail.com.pl/documentation/>

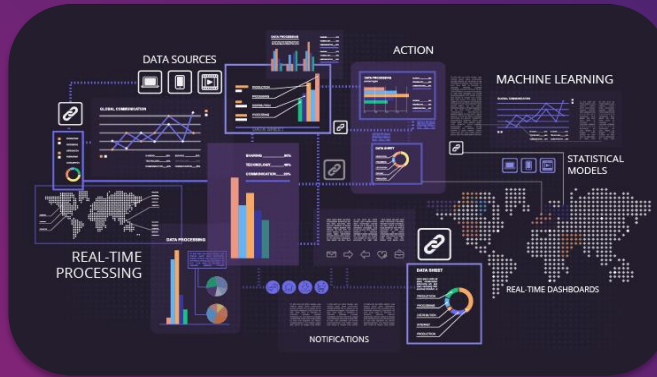
Grail: A Framework for Adaptive and Believable AI in Video Games

<https://ieeexplore.ieee.org/abstract/document/8609687>

Zastosowania AI w grach video



NPCs / Boty



Analitka gry



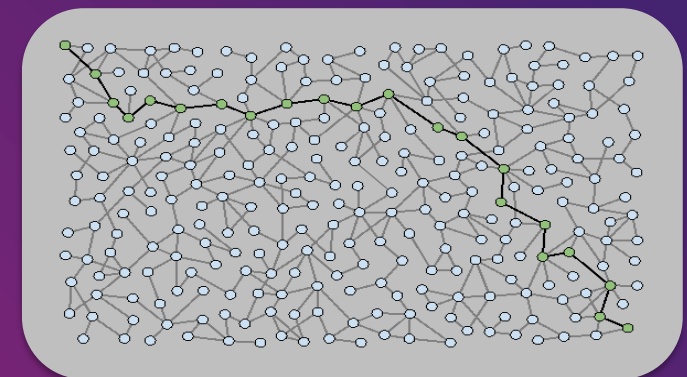
PCG



QA/Testowanie



Analitka
biznesowa



Pathfinding itp
(dawne rozumienie AI
w gamedev)

Plan

Metody podejmowania decyzji

grail

- drzewa behawioralne
- utility AI
- planery
- MCTS
- uczenie offline i drzewa decyzyjne

Analityka – bardzo ogólnie

Bonus na koniec:

- poruszanie i nawigacja



Drzewa behawioralne



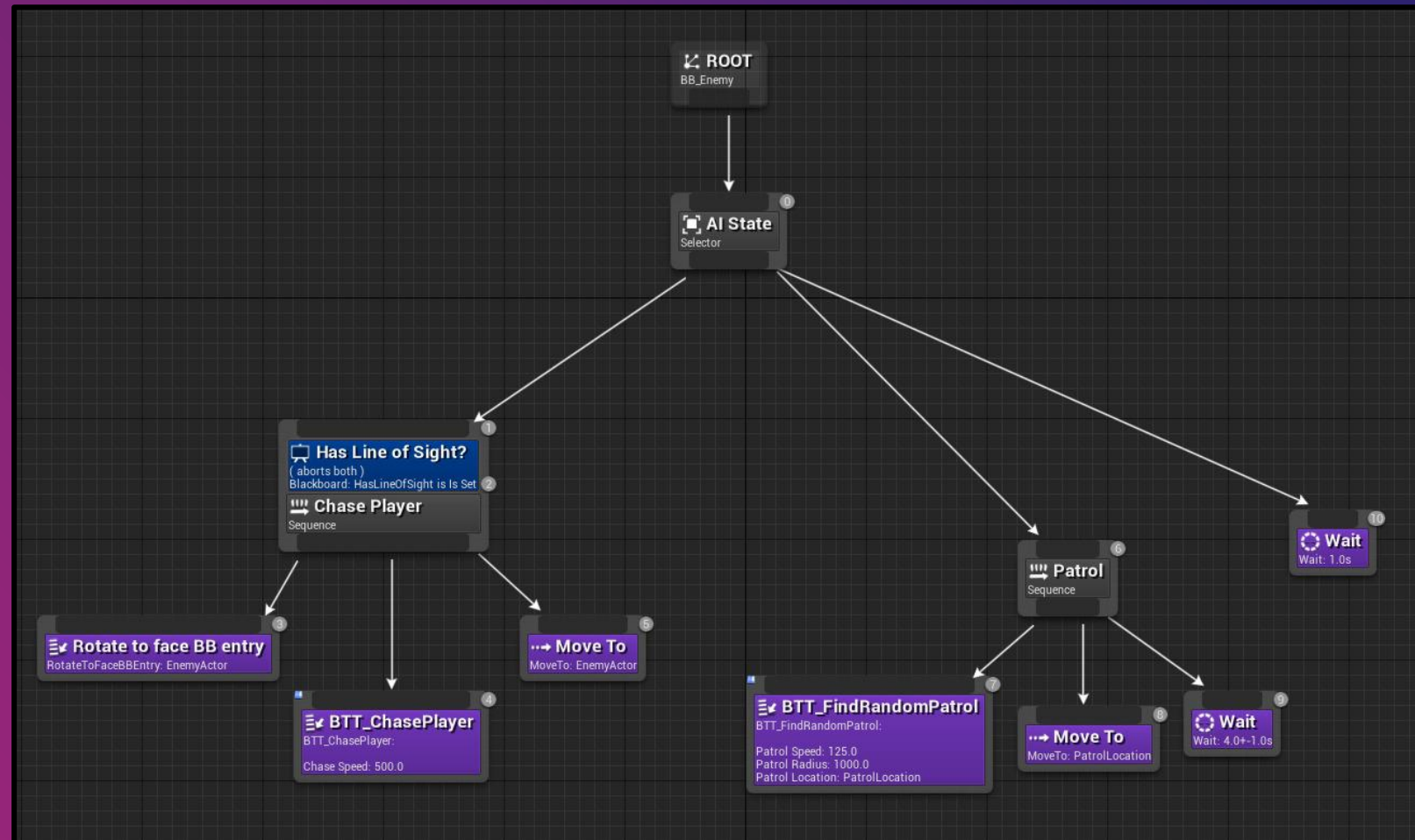
Drzewa behawioralne

Zachowanie (Behavior) jako uogólnienie akcji w grze

Dominująca metoda do AI w grach video

Wizualny sposób pisania skryptów

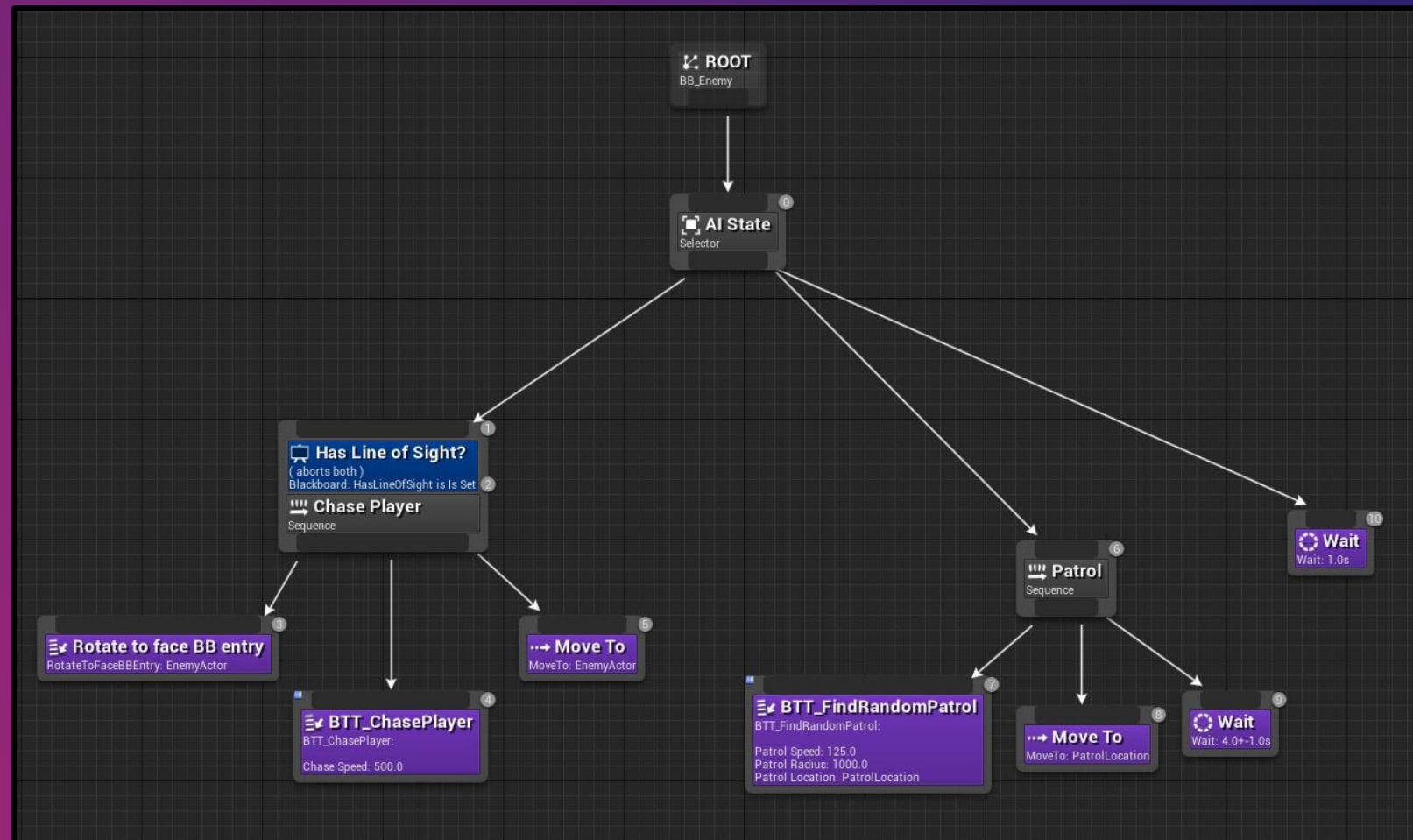
- wykonanie programu zdefiniowanego jako drzewo
- instrukcje w węzłach
- zachowania w liściach



Drzewa behawioralne

Flow: przechodzenie po drzewie – nie zawsze wraca do korzenia

Każdy węzeł zwraca {SUCCESS, FAIL, RUNNING}

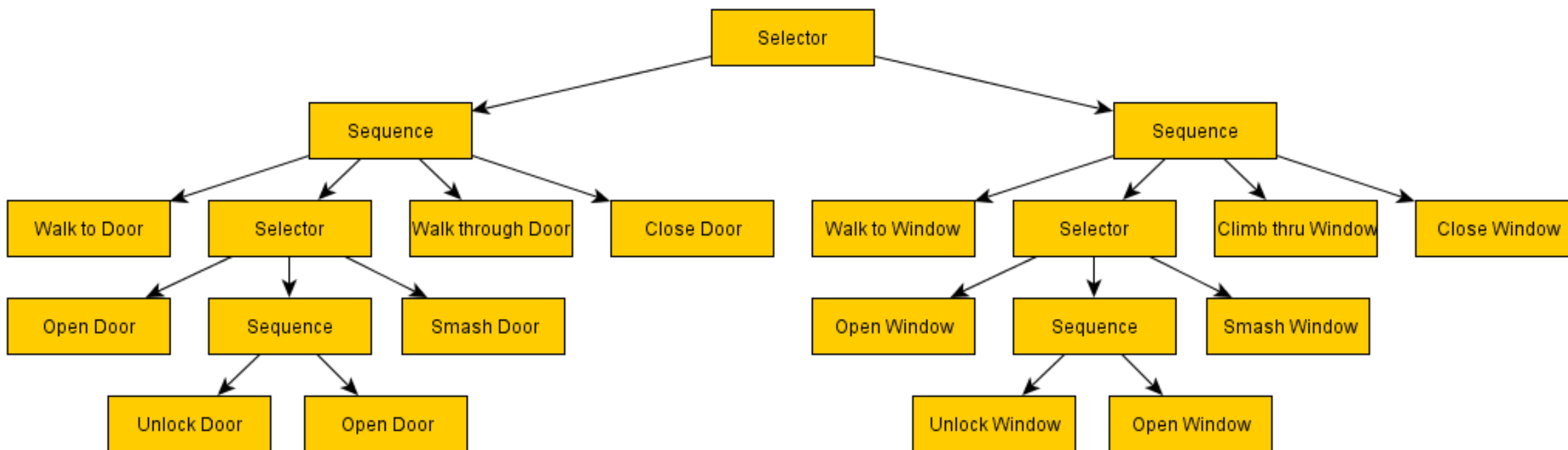


Drzewa behawioralne

Selektor – znajdź i wykonaj pierwszego potomka zwracającego SUCCESS

Sekwencja – znajdź i wykonaj pierwszego potomka, który jeszcze nie zwrócił SUCCESS

Dekorator – dodatkowe warunki (np. IF)



Zastosowania w grach video

Większość gier z AI

Pierwsze zastosowanie:



Utility AI (system użyteczności)

EMIL KAUDER

A History of Marginal Utility Theory



PRINCETON LEGACY LIBRARY

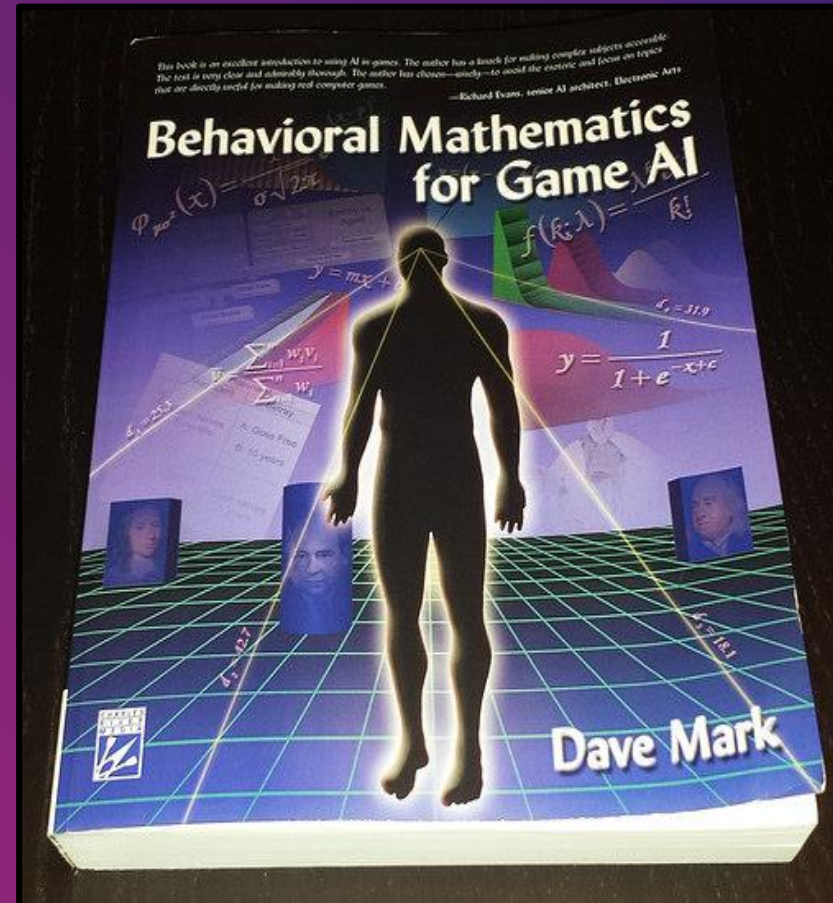
Utility AI

Utility jako uniwersalna miara użyteczności zachowania

Inteligentna jednostka wybiera zachowanie, które w danym miejscu i w danej chwili jest dla niej najbardziej użyteczne

Termin stosowany również w psychologii, ekonomii i behawiorystyce

Dave Mark propagatorem tej teorii w grach
W 2009 wydał książkę =>



Czynniki (Considerations)

Czynniki to aspekty świata gry, które agent bierze pod uwagę przy wykonywaniu akcji

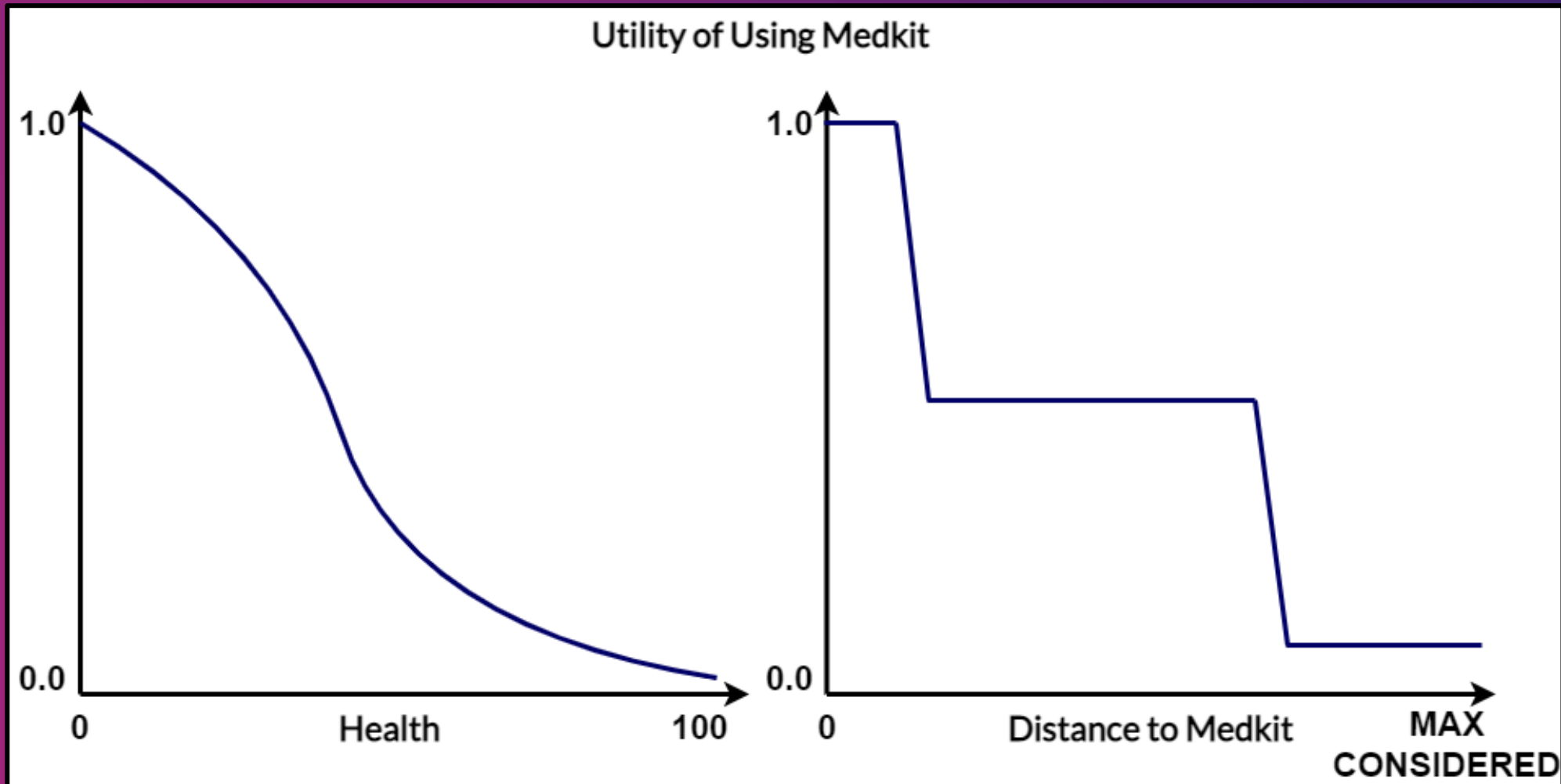
Implementacja:

- reużywalne funkcje
- cache'owanie wyników

$$F(state) \rightarrow R$$

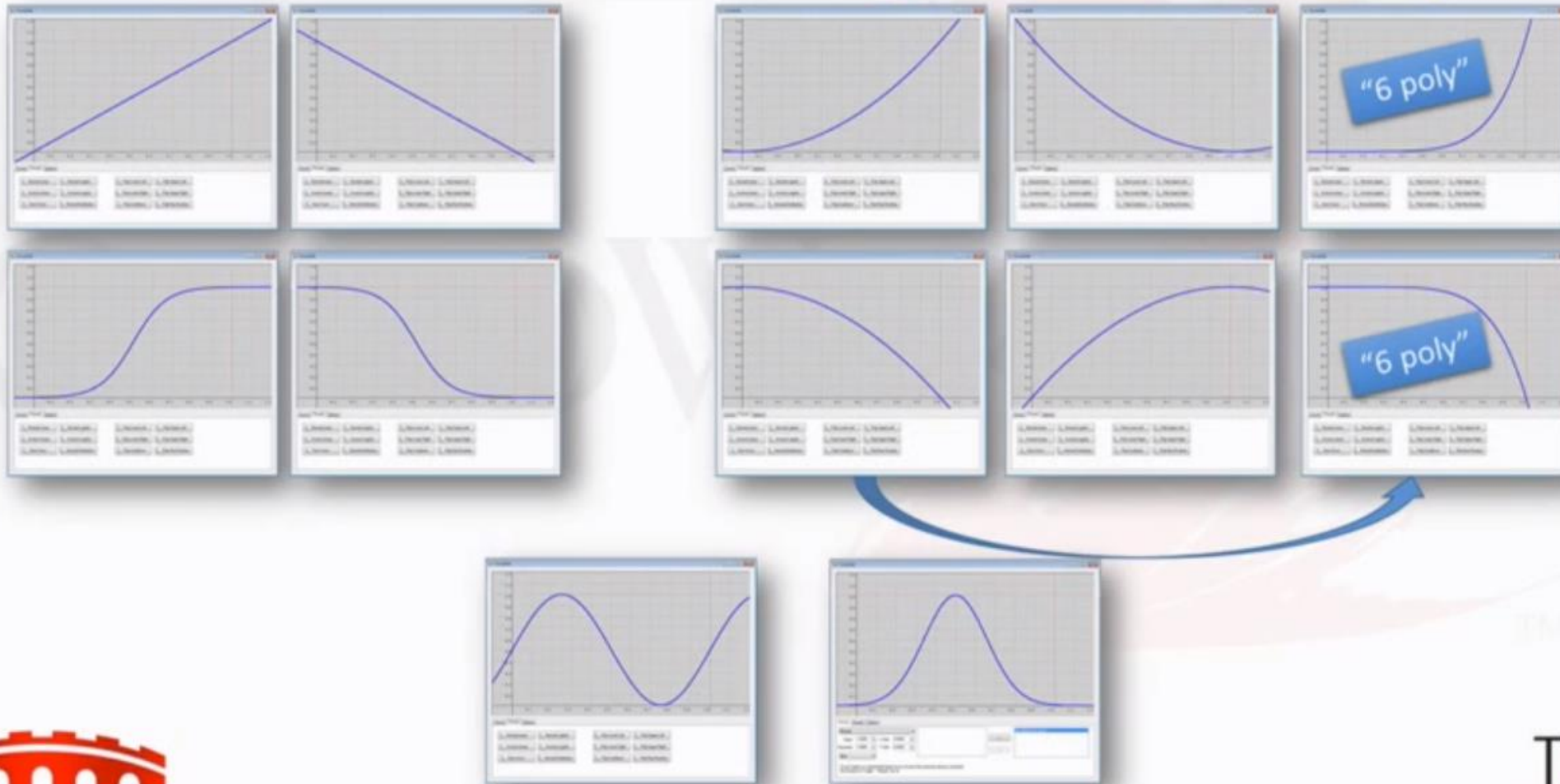
Utility Curves / Response Curves

Dla każdej akcji modelowana jest jej użyteczność względem danego czynnika (Consideration)

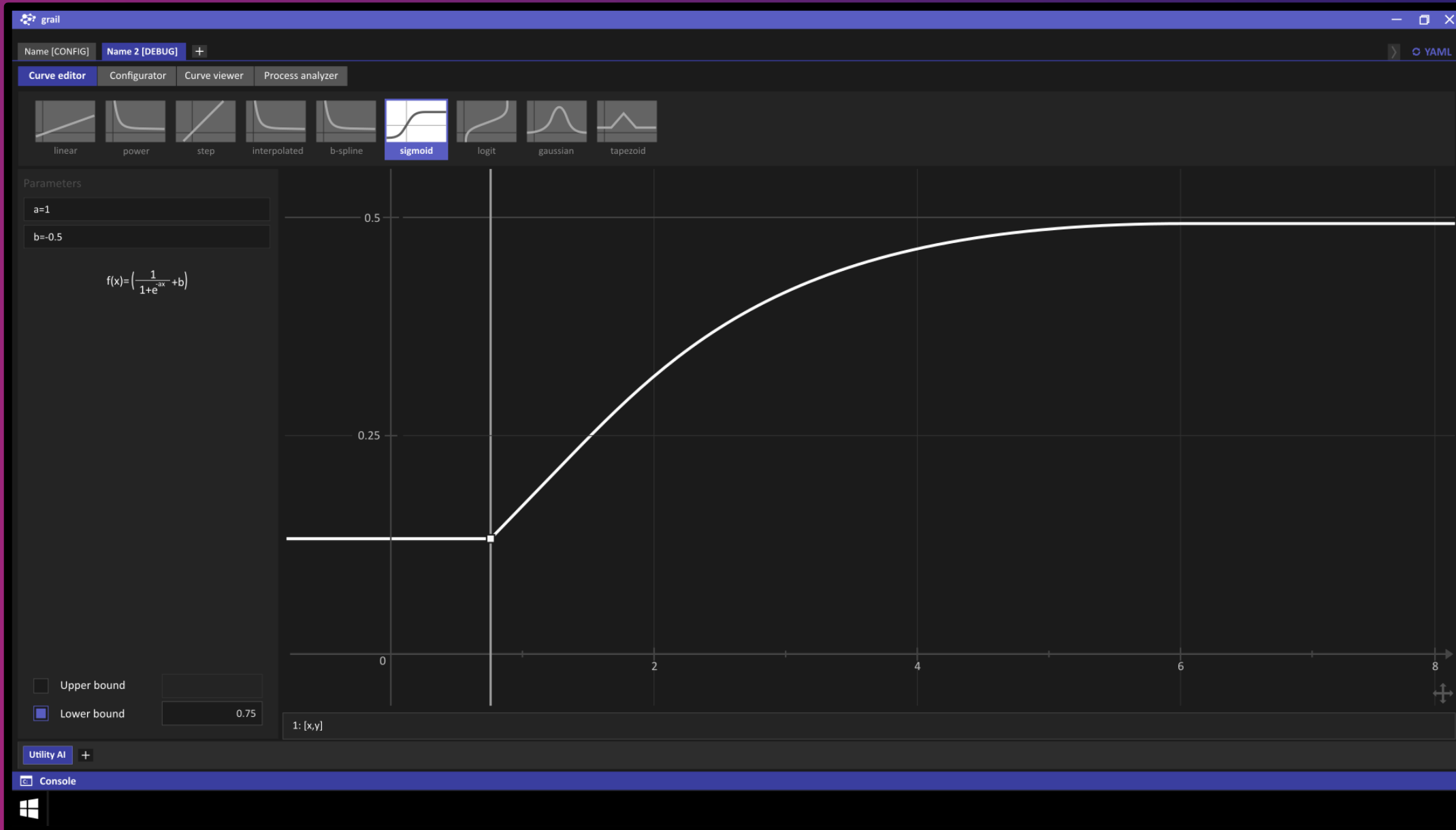


Utility Curves / Response Curves

Response Curve Presets



Utility Curves / Response Curves



Ewaluator

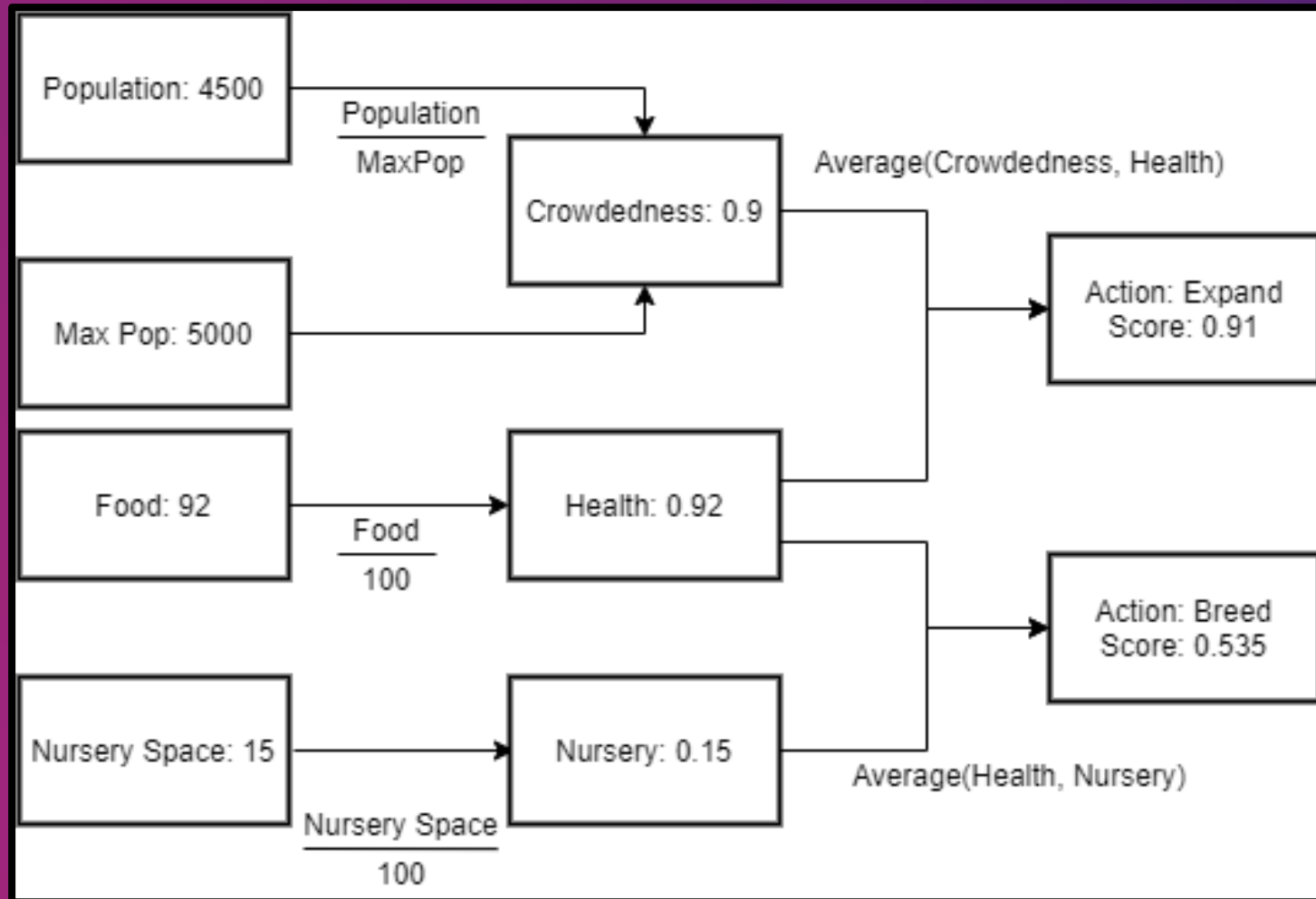
- Agregacja dla akcji
- Łączenie użyteczności danej akcji względem różnych czynników
- Wyjście: finalna użyteczność akcji

Metody:

- MAX
- MIN
- Średnia ważona
- Suma
- Iloczyn
- Inne np. S-normy I T-normy z logiki rozmytej

Ewaluator

- Staranna kontrola nad wejściami i wyjściami pozwala na:



Utility Selector

Wybór zachowania w grze

Najpopularniejsze metody:

- argmax
- pseudo-ruletka
- losowo z k najlepiej ocenionych

Opcjonalne modyfikacje:

- Ranks
- Persistence

Wybrane zastosowania w grach video



Planer



Goal Oriented Action Planning (GOAP)

- Metoda z dziedziny planowania i kontroli
- W poważnej nauce znana od 1971 roku

Planowanie w grach video:

- Pierwszy raz zastosowane w F.E.A.R. (2005)
- Jeff Orkin, PhD

SELECTED PUBLICATIONS & PRESENTATIONS

Orkin (2013), [Collective Artificial Intelligence: Simulated Role-Playing from Crowdsourced Data](#). *PhD Thesis, MIT Media Lab*.

Orkin & Roy (2012), [Understanding Speech in Interactive Narratives with Crowdsourced Data](#). *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE)*.

Orkin, Smith, Reckman & Roy (2010), [Semi-Automatic Task Recognition for Interactive Narratives with EAT & RUN](#). *Proceedings of the 3rd Intelligent Narrative Technologies Workshop (INT3)*.

Orkin & Roy (2009), [Automatic Learning and Generation of Social Behavior from Collective Human Gameplay](#). *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Orkin & Roy (2007), [The Restaurant Game: Learning Social Behavior and Language from Thousands of Players Online](#). *Journal of Game Development*, 3(1), 39-60.

Orkin (2006), [Three States and a Plan: The AI of F.E.A.R.](#) *Proceedings of the Game Developer's Conference (GDC)*. [[paper](#) | [slides](#)]

Orkin (2005), [Agent Architecture Considerations for Real-Time Planning in Games](#). *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE)*. [[paper](#) | [slides](#)]

Orkin (2004), [Symbolic Representation of Game World State: Toward Real-Time Planning in Games](#). *Proceedings of the AAAI Workshop on Challenges in Game AI*. [[paper](#) | [slides](#)]

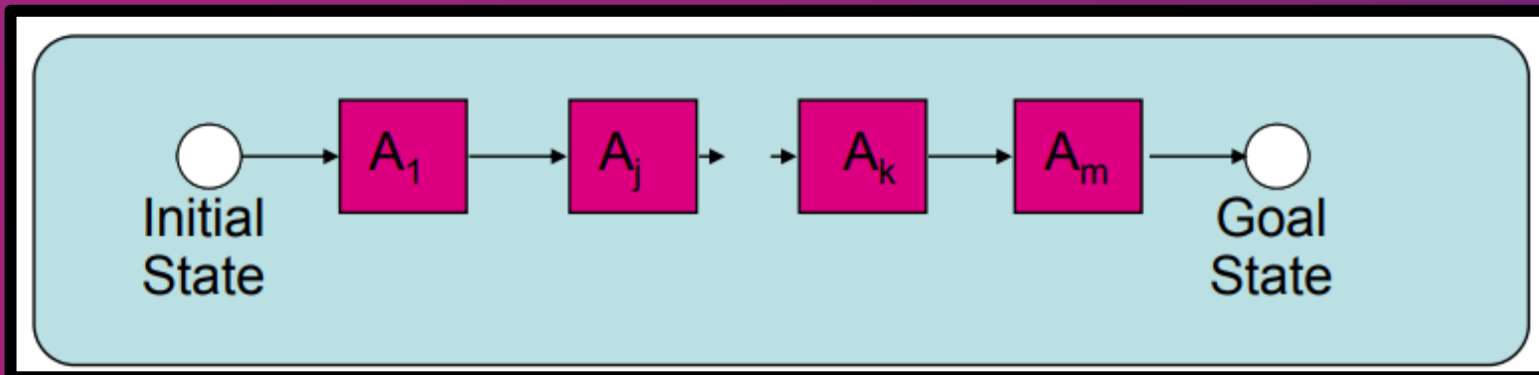
Goal Oriented Action Planning (GOAP)

Podajemy nature problemu:

- Model świata
- Stan wejściowy
- Cel(e) do osiągnięcia
- [opcjonalnie] Heurystyki dystansu między stanami

Model świata:

- Akcje
- Warunki akcji (preconditions)
- Efekty akcji (postconditions)



Reprezentacja stanu (ogólnie)

Często formalne języki podobne do *Prologa*

Formal syntax like a programming language

- Initial State (:init ...)
- Goal (:goal ...)
- Actions (:action name
:parameters (?from ?to ?dir)
:preconditions (...)
:effects (...)
)

Literals like Lisp, e.g., bot-at(area1) (bot-at area1)

Gripper task: Pick-up operator

Action/Operator:

Description: The robot can pick up x in y with z .

Precondition: BALL(x), ROOM(y), GRIPPER(z), at-ball(x , y), at-roby(y) and free(z) are true.

Effect: carry(z , x) becomes true. at-ball(x , y) and free(z) become false. Everything else doesn't change.

In PDDL:

```
(:action pick-up :parameters (?x ?y ?z)
  :precondition (and (BALL ?x) (ROOM ?y) (GRIPPER ?z)
                    (at-ball ?x ?y) (at-roby ?y) (free ?z))
  :effect (and (carry ?z ?x)
              (not (at-ball ?x ?y)) (not (free ?z))))
```

Malte Helmert

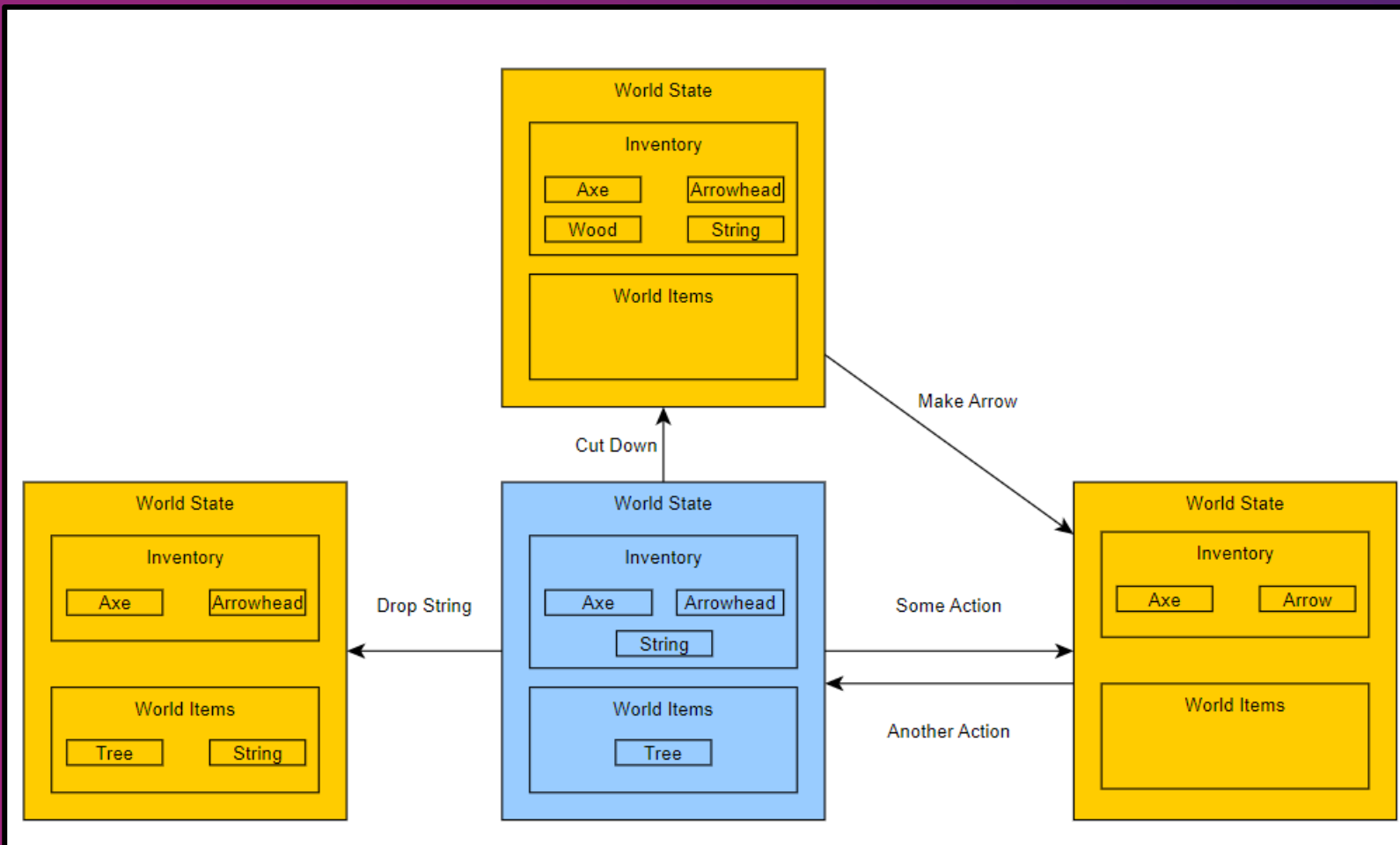
October 16th, AI Planning 12/15

Symbolic Representation of Game World State: Toward Real-Time Planning in Games

Jeff Orkin
Monolith Productions, Inc.
Kirkland, WA 98033
<http://www.lith.com>
jorkin@blarg.net

Reprezentacja stanu (Grail)

- Blackboard architecture
- Silne typowanie na zasadzie idei wariantów (any)
- Zoptymalizowane zarządzanie pamięcią

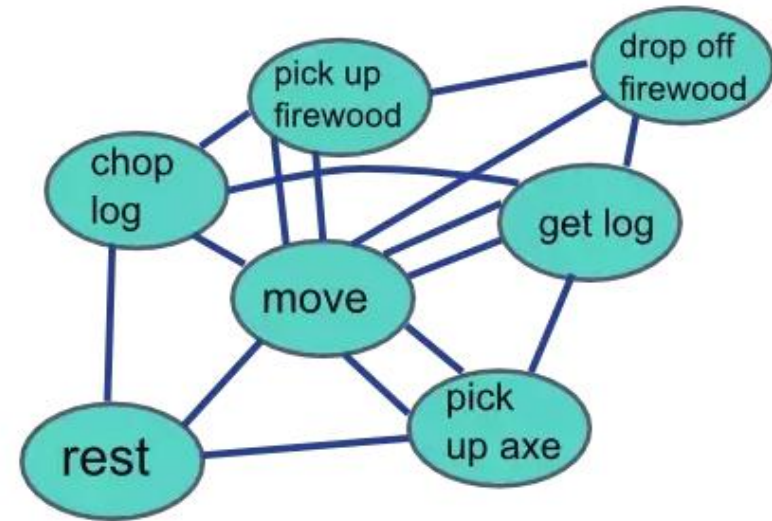


Skalowanie

Gdy gra się rozrasta

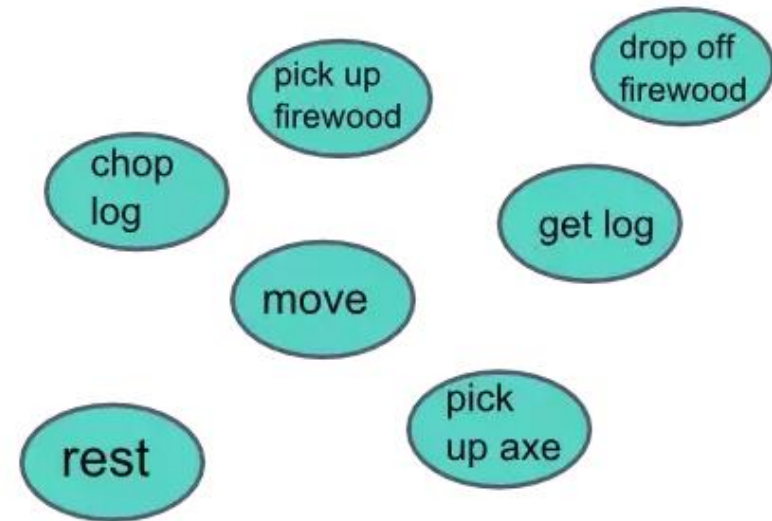
- Planer ma szansę skalować się liniowo wraz ze wzrostem możliwych zachowań
- Oprogramujemy tylko nowe zachowanie
- Drzewa behawioralne i automaty skończone nie mają tej zalety!
- Zazwyczaj reiteracja całej struktury

GOAP turn this :



Finite State Machine States : Connected Everywhere

Into this :



Algorytm planowania

Zależnie od reprezentacji, planowanie sprowadza się do problemu **wyszukiwania** (search problem) lub **rezolucji logicznej** (logical reasoning)

- Forward chaining
- Backward chaining

Scieżka od **stanu bieżącego (init)** do stanu realizującego cel (**goal**)

Najczęściej znów stosowany algorytm A^*

- opcjonalne heurystyki szacujące odległość między stanami
- można się inspirować np. odległością Levenshteina

Dodatkowe czynniki

W zależności od konkretnego zastosowania:

- Częściowe planowanie
- Kiedy przeliczać istniejący plan
- Probablistyczne planowanie
- Hierarchiczne planowanie
- Planowanie wielokryterialne

Screenshot z narzędzi Graila (planowanie)

The screenshot displays the Grail tool's interface, specifically the 'Process analyzer' tab. The main area shows a task plan for 8 iterations. The plan is organized into four rows, each representing a different action type: 'cast_fireball', 'drink_potion', 'attack', and 'go to'. The actions are distributed across the iterations as follows:

- Iteration 1:** cast_fireball: monster 03
- Iteration 2:** drink_potion: potion 01
- Iteration 3:** go to: potion 01
- Iteration 4:** attack: monster 04
- Iteration 5:** cast_fireball: monster 03
- Iteration 6:** drink_potion: potion 01
- Iteration 7:** go to: potion 03
- Iteration 8:** go to: potion 01

The bottom section of the interface shows the definitions for the objects used in the plan:

potion 01		
regen	int	2
activated	bool	FALSE

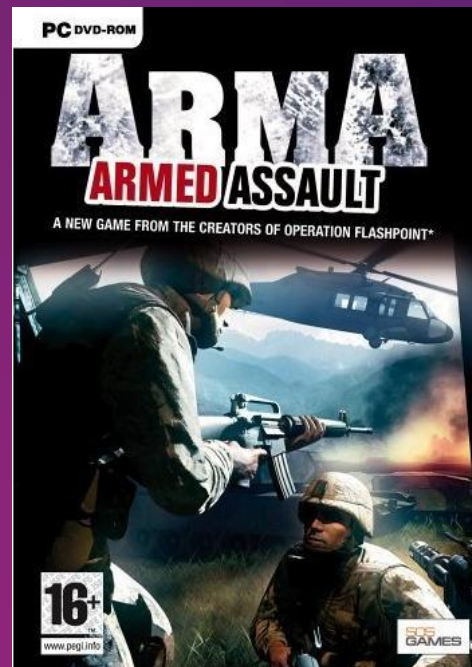
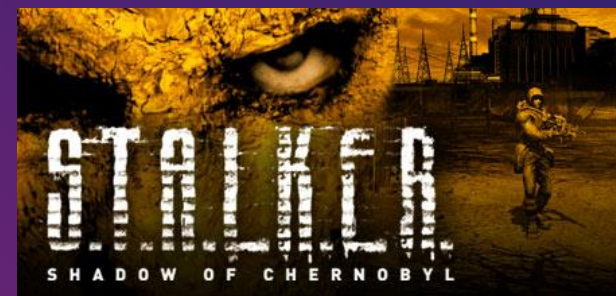
adventurer 02		
mana	int	2
damage	float	1
health	float	100

monster 03		
invincible	bool	FALSE
damage	float	10
health	float	2.5

monster 04		
invincible	bool	FALSE
damage	float	10
health	float	2.5

The interface also includes a search bar and a 'Console' tab at the bottom.

Wybrane zastosowania w grach video

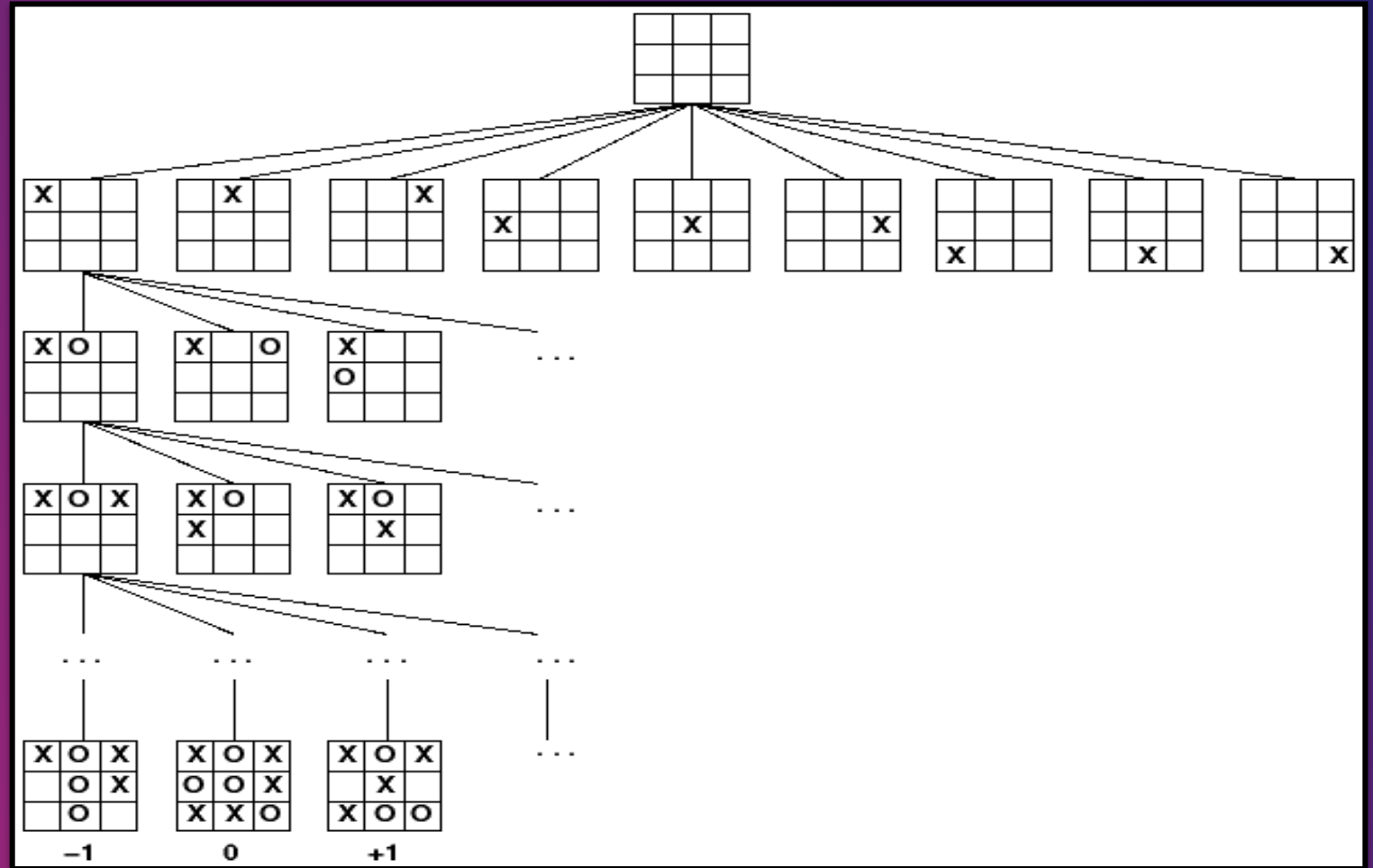


Monte Carlo Tree Search (MCTS)



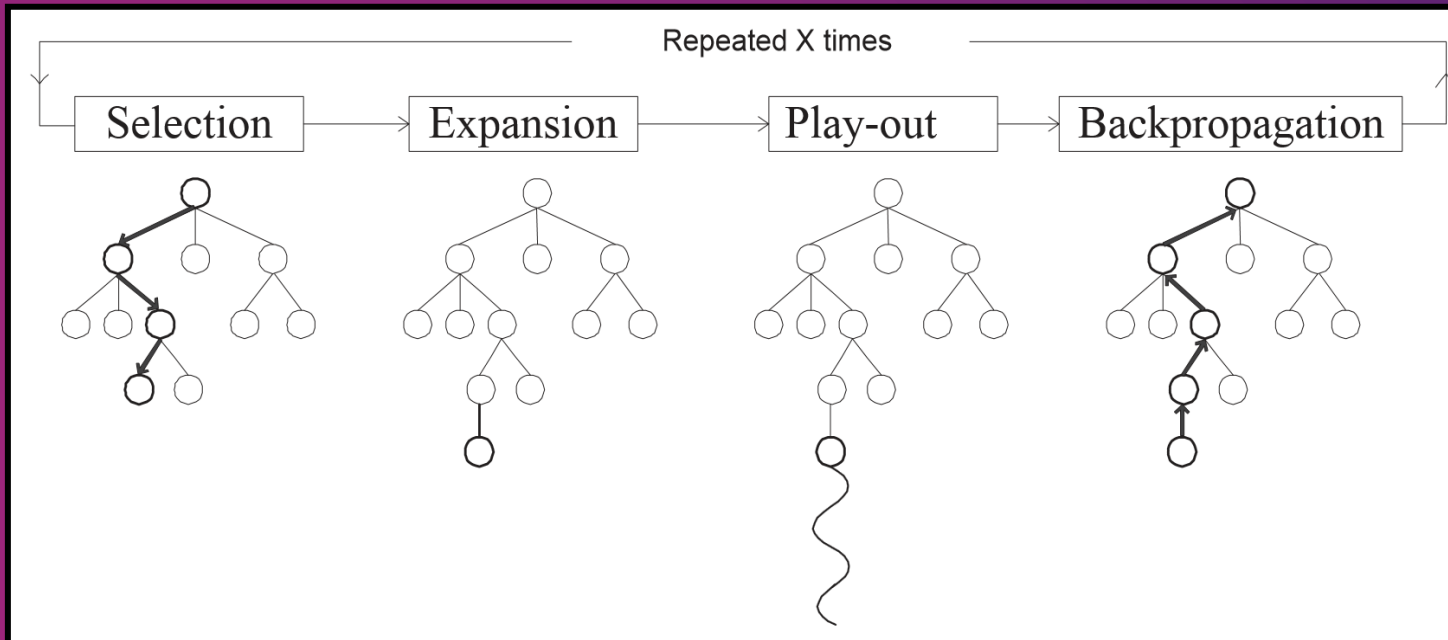
Drzewo gry

- węzły – stany gry
- krawędzie – akcje
- (przejścia między stanami)



MCTS

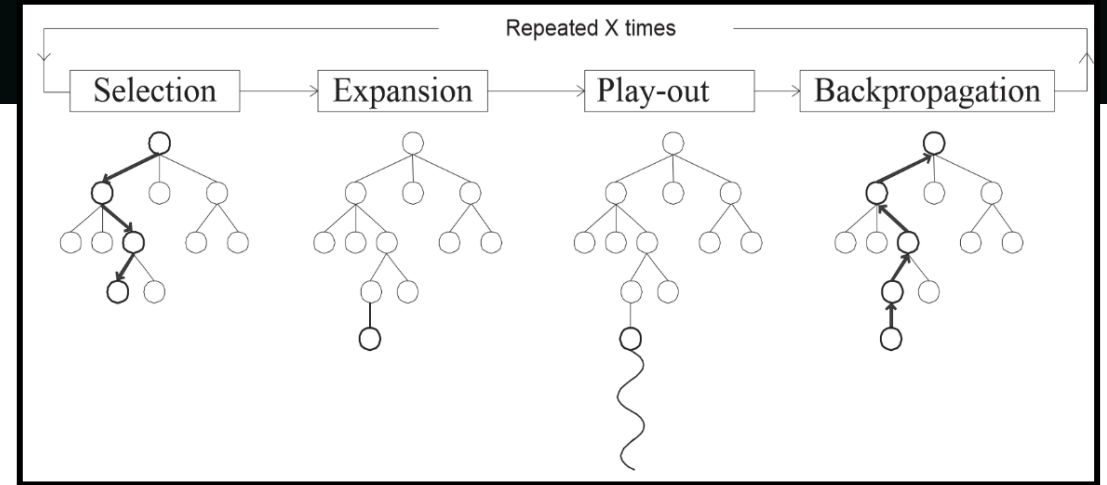
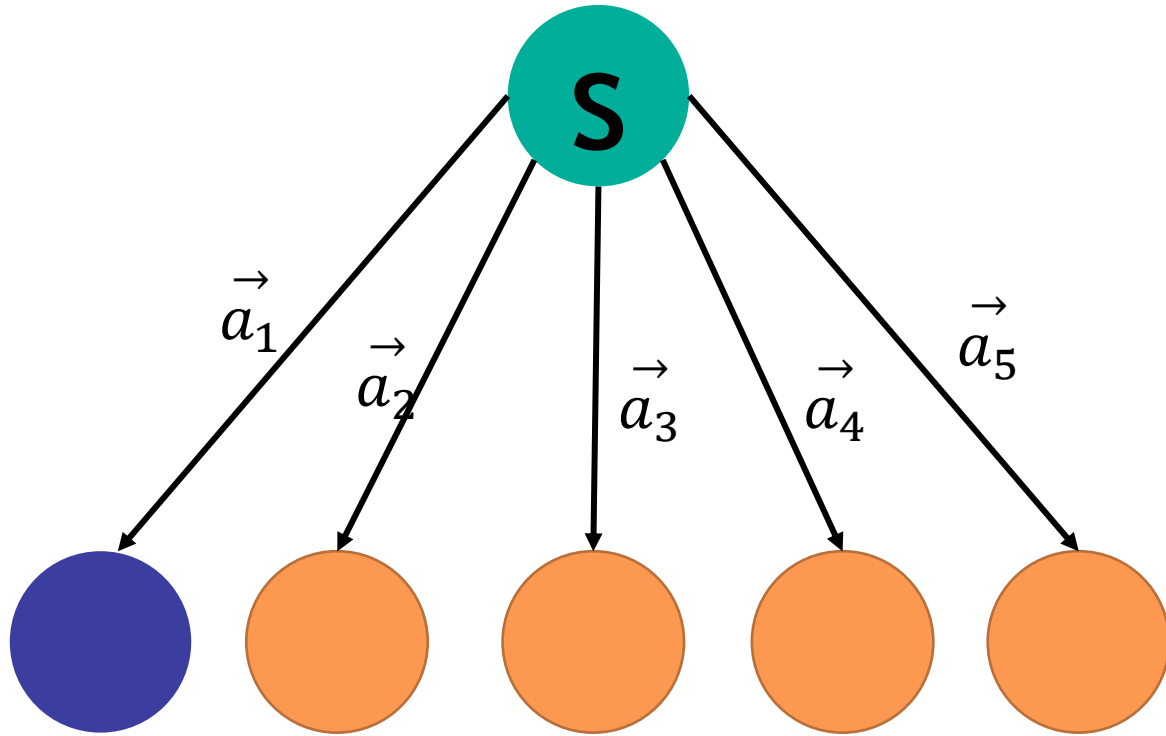
- Pierwsze zastosowanie w 2006 roku (Go)
- Znaczny przełom w sile gry
- AlphaGo vs. Lee Sedol (2016)



[Monte Carlo Tree Search: A Review of Recent Modifications and Applications]

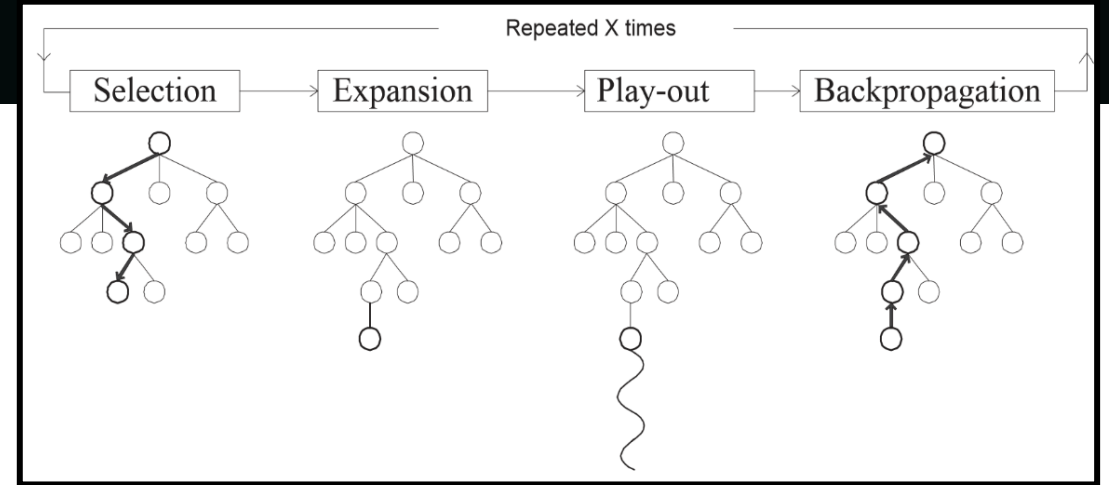
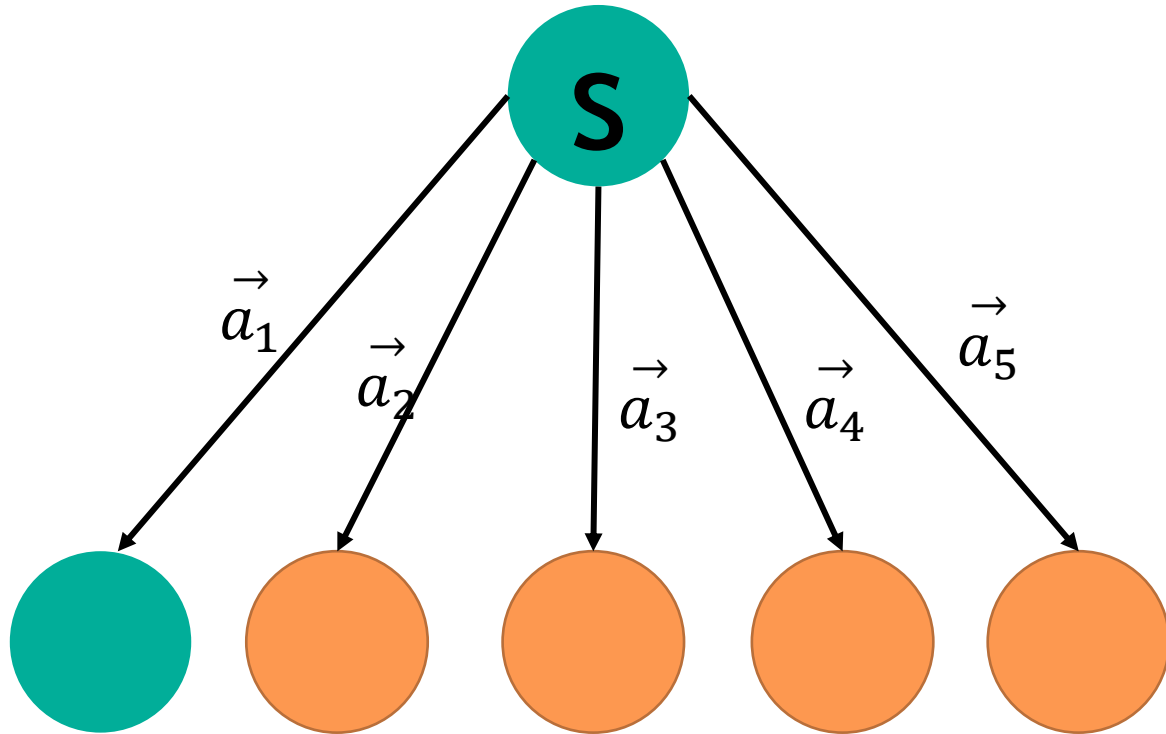
<https://arxiv.org/abs/2103.04931>

Selection



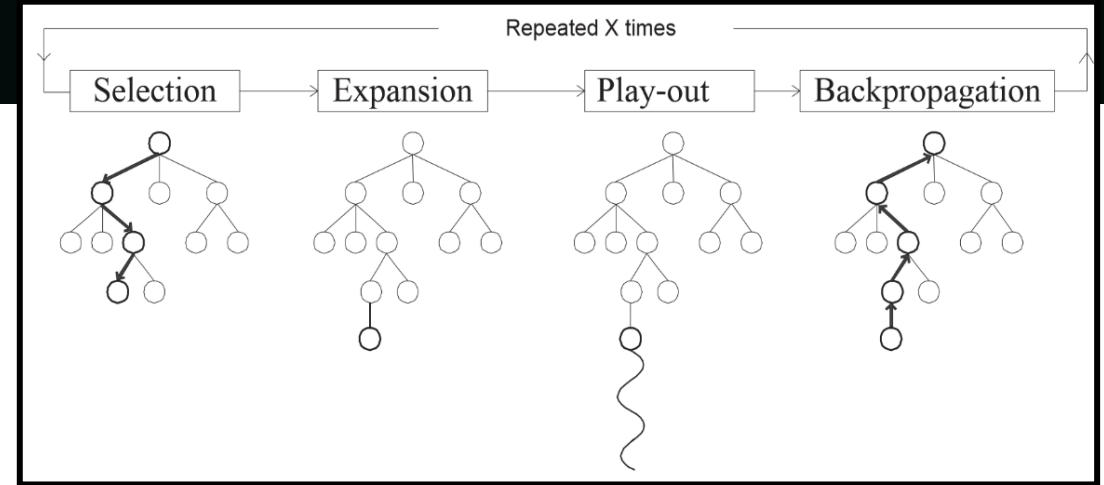
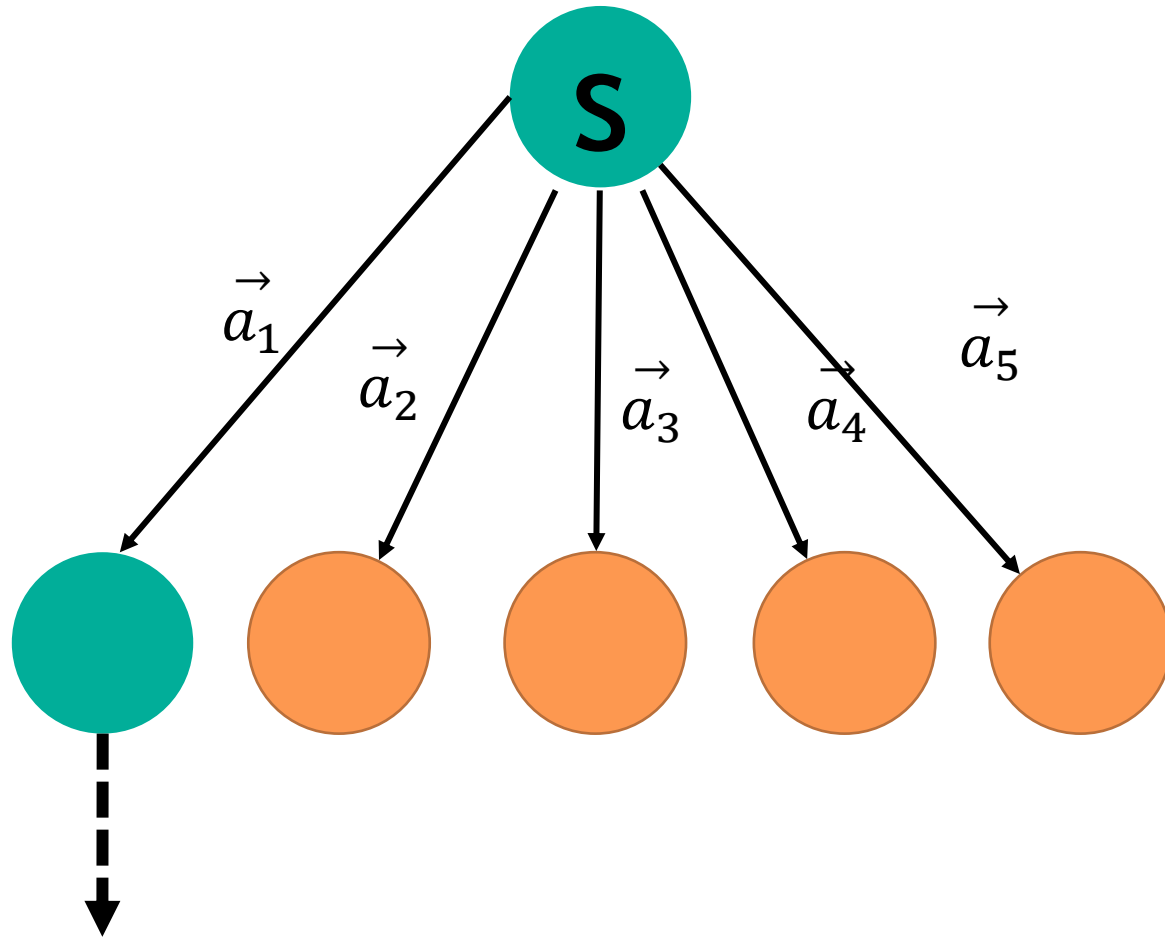
- always start from the root node
- get available actions
- select the next node until we fall out of the cached portion of the tree

Expansion

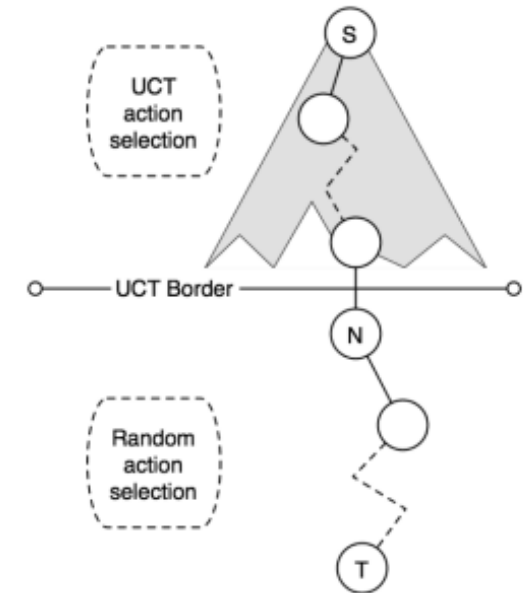


- add one* new node to the tree

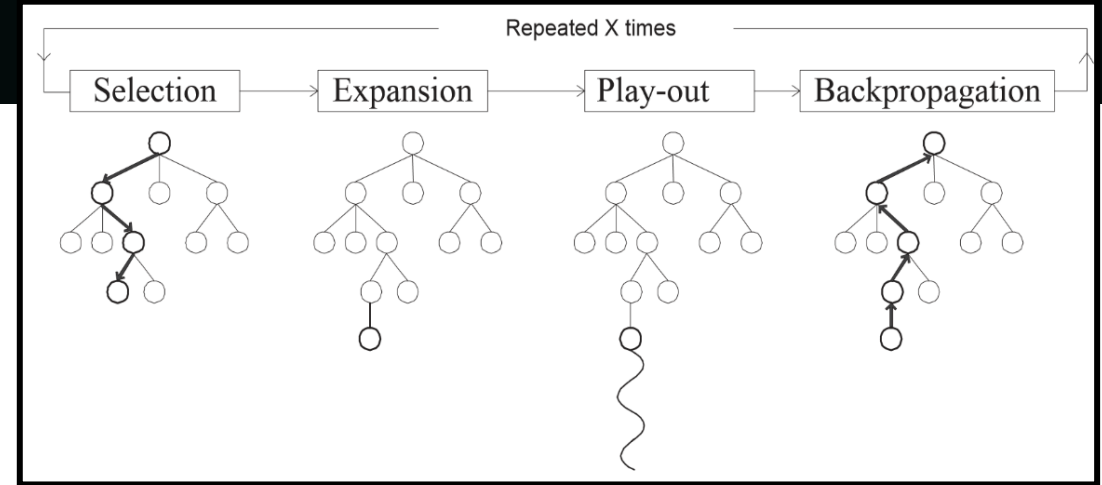
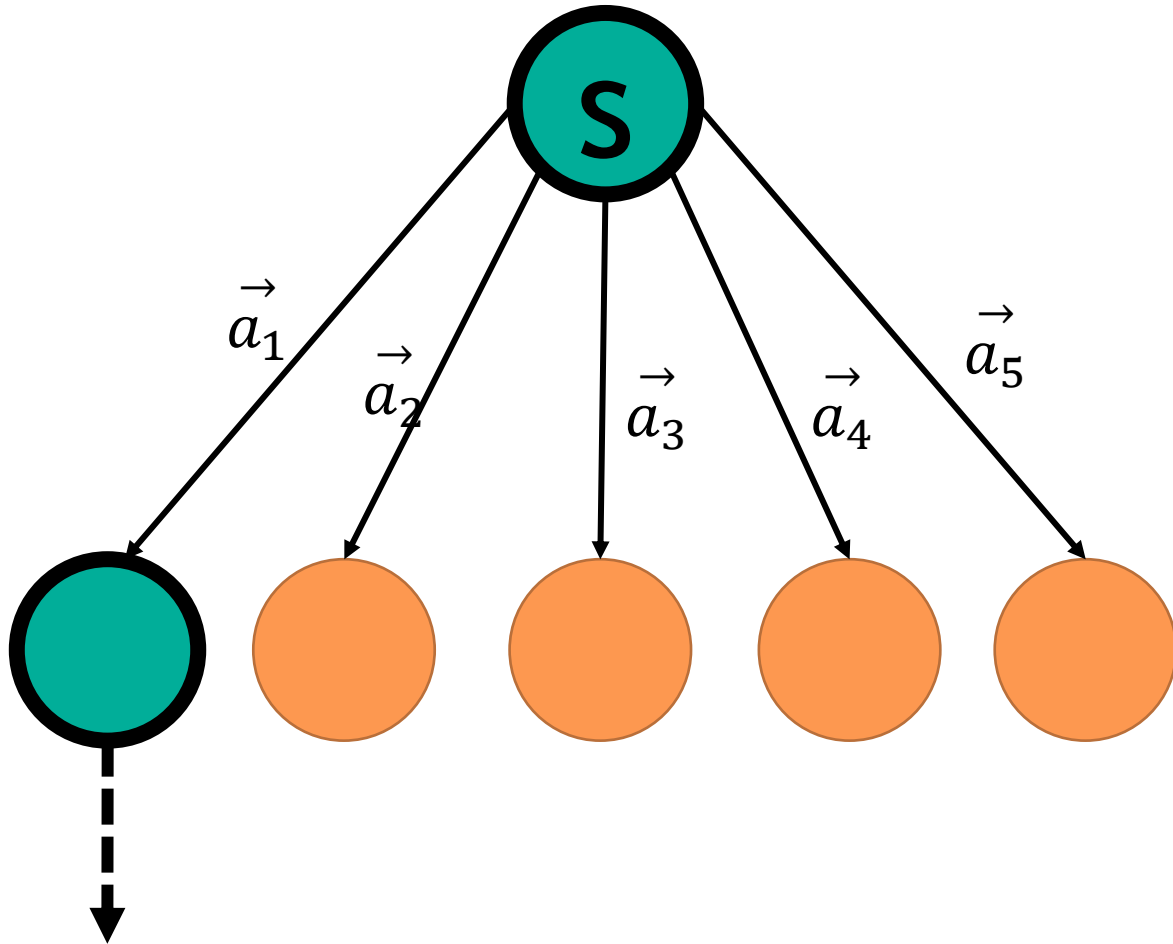
Expansion



- random or quasi-random simulation to the terminal state

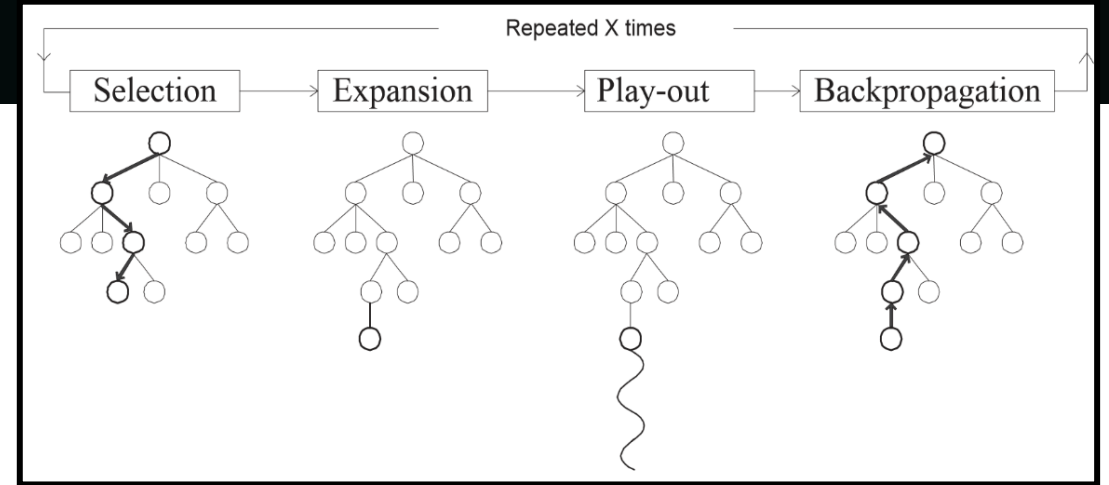
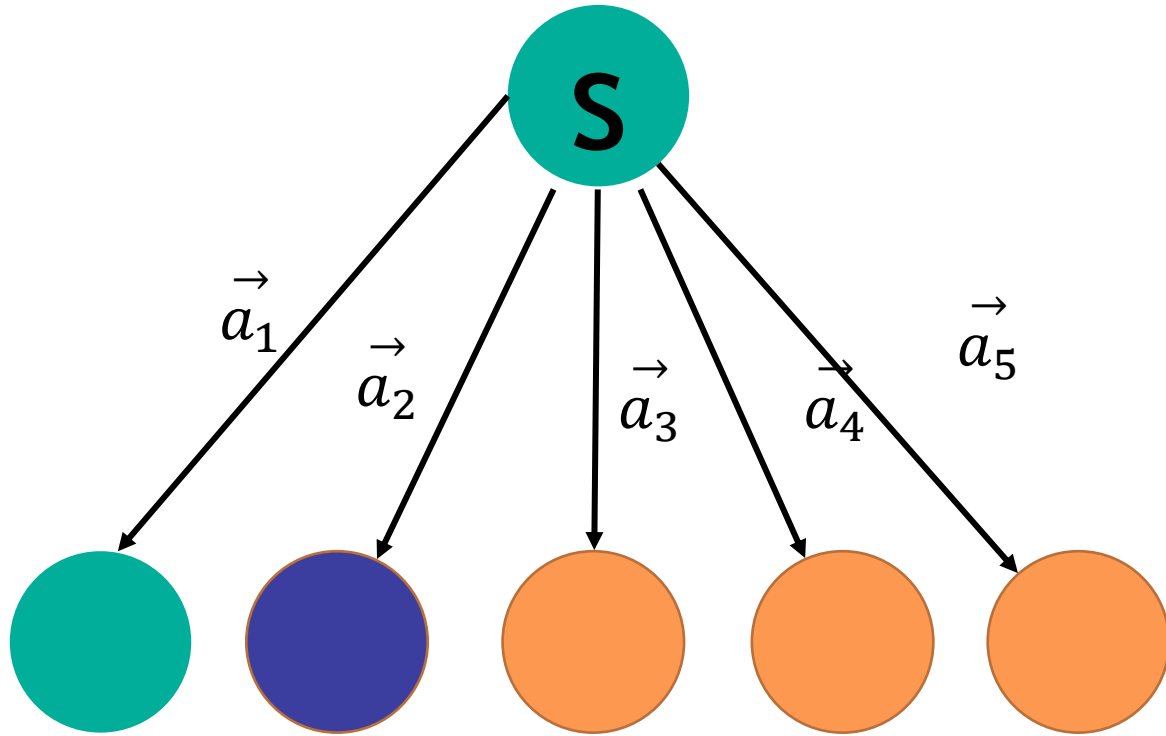


Backpropagation

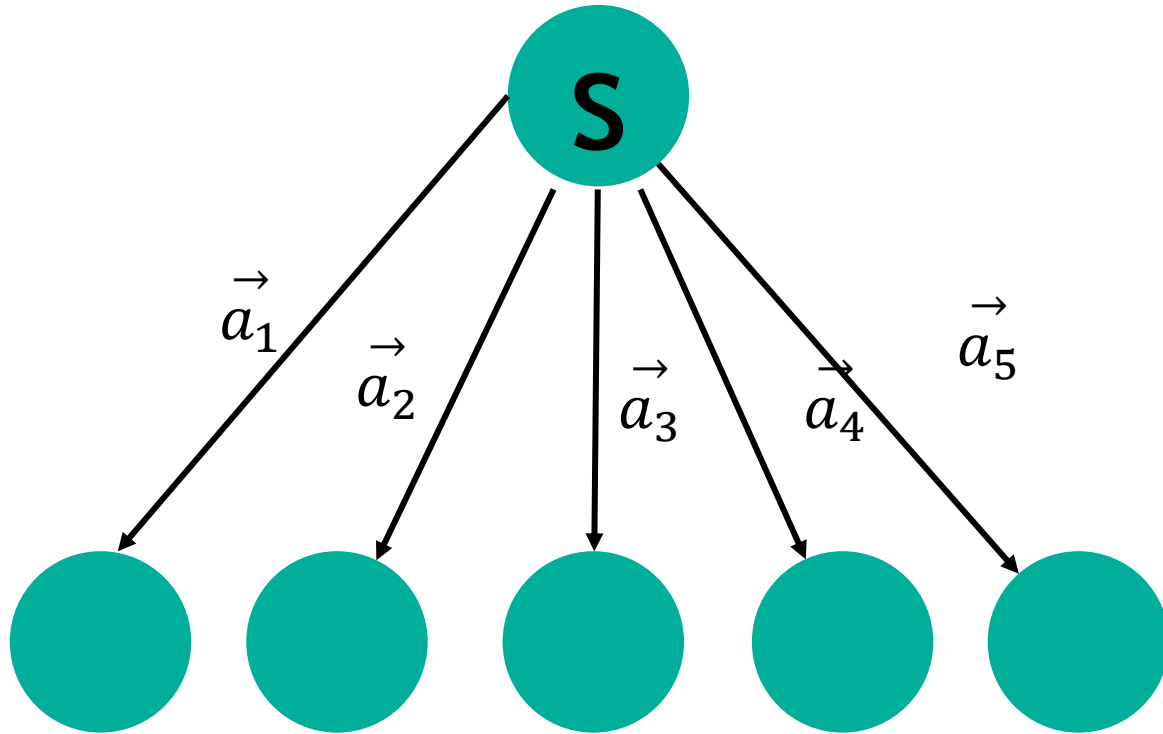


- update statistics (**total score per player, number of visits**)
- on the whole path of nodes visited in the selection phase

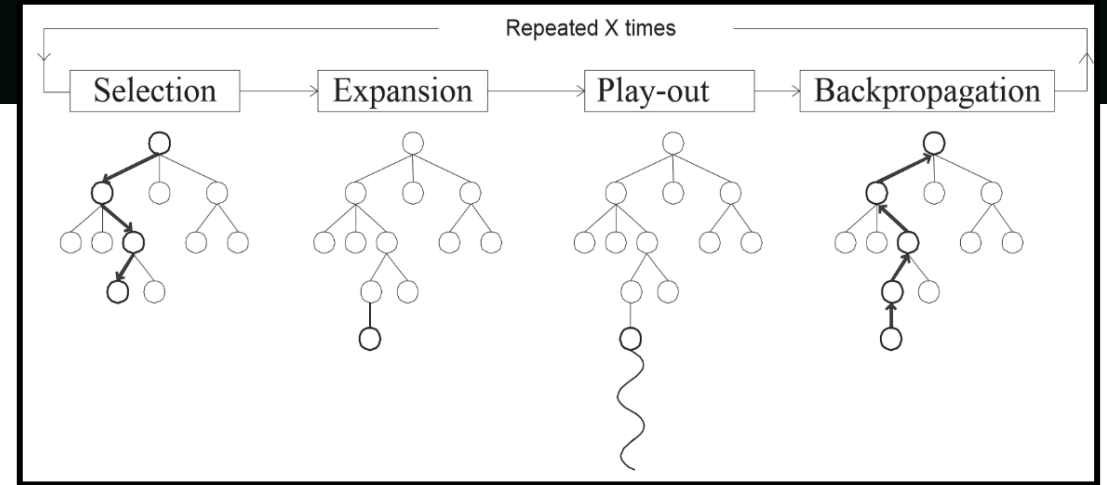
Selection



Selection



we have visited each node once already



Upper Confidence Bounds Applied for Trees (UCT)

$$a^* = \mathop{\text{arg max}}_{a \in A(s)} \left\{ Q(s, a) + C \sqrt{\frac{\ln N(s)}{N(s, a)}} \right\}$$

Dla każdego gracza w grze:

a – rozpatrywana akcja

s – bieżący stan

$Q(s, a)$ – historyczny średni wynik a w s

$N(s)$ – liczba wizyt do s (zaobserwowań a)

$N(s, a)$ – liczba wyborów a w s



Alternatywy dla UCT

Comparison of Different Selection Strategies in Monte-Carlo Tree Search for the Game of Tron

Pierre Perick, David L. St-Pierre, Francis Maes and Damien Ernst

Temporally-Extended ϵ -Greedy Exploration

Will Dabney¹ Georg Ostrovski¹ André Barreto¹

Modifications of UCT and sequence-like simulations for Monte-Carlo Go

Yizao Wang
Center of Applied Mathematics
Ecole Polytechnique, Palaiseau, France
yizao.wang@polytechnique.edu

Sylvain Gelly
TAO (INRIA), LRI, UMR (CNRS - Univ. Paris-Sud)
University of Paris-Sud, Orsay, France
sylvain.gelly@lri.fr

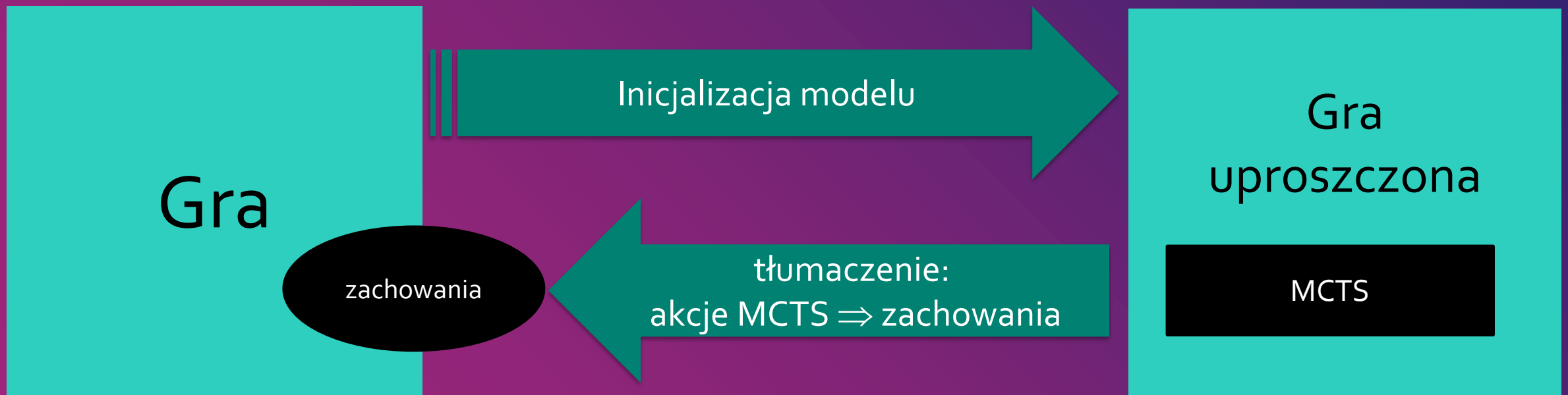
- Note that a Bernoulli random variable with $p = 0.5$ is the reward distribution that will give the highest variance (which is $\frac{1}{4}$).
- We can also compute the sample variance σ_j for each action.
- Then use the upper confidence bound for action j of:

$$\sqrt{\frac{\ln n}{n_j} \min\left(\frac{1}{4}, \left(\sigma_j + \frac{2 \ln n}{n_j}\right)\right)} \quad (5)$$

MCTS w grze video

Developer gry definiuje uproszczony model

- operowanie na dokładnym symulatorze => prawie nigdy niemożliwe w praktyce



Popularne uproszczenia gier

- Limit dostępnych opcji (np. samplowanie punktów z mapy)
- Granulacja (kubekowanie) zmiennych
- Zrezygnowanie z mniej ważnych zmiennych
- Grupowanie akcji w hierarchie
- Zachłanne oskryptowane akcje
- Zrezygnowanie z losowości (np. zastąpienie wartością oczekiwaną)

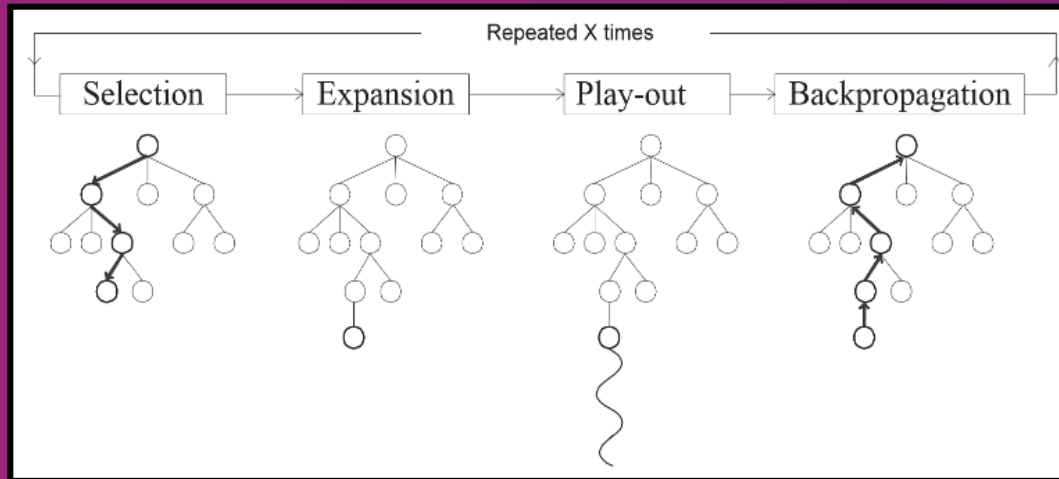
Celem zmniejszenie **złożoności kombinatorycznej** gry i zwiększenie **wydajności obliczeniowej**

Uczenie offline



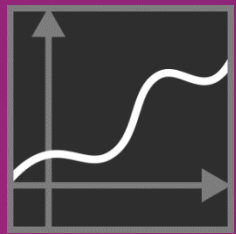
Uczenie offline

- Uruchamianie MCTS w fazie produkcji gry
- Używanie tylko botów
- Zwolnienie czasu gry
np. 50 razy => więcej rzeczywistego czasu na wybór akcji

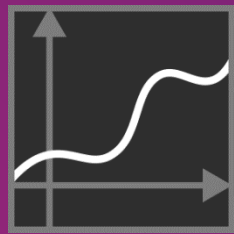


Wektoryzacja

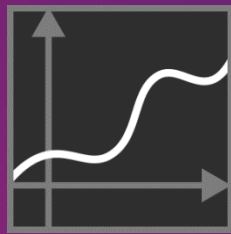
Twórca gry dostarcza miary – aspekty stanu gry, które będą brane pod kątem decyzji:



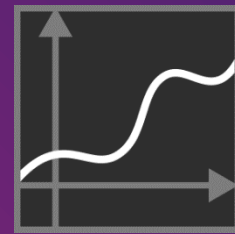
MyHP



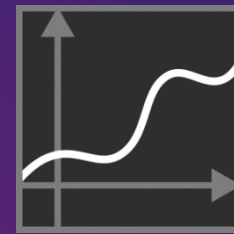
EnemyHP



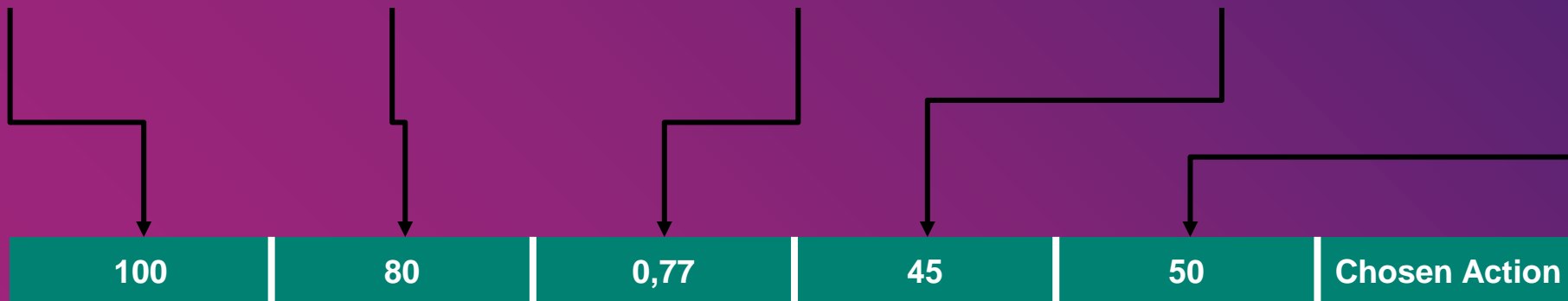
AttackAngle



MyWeaponDmg



EnemyWeaponDmg

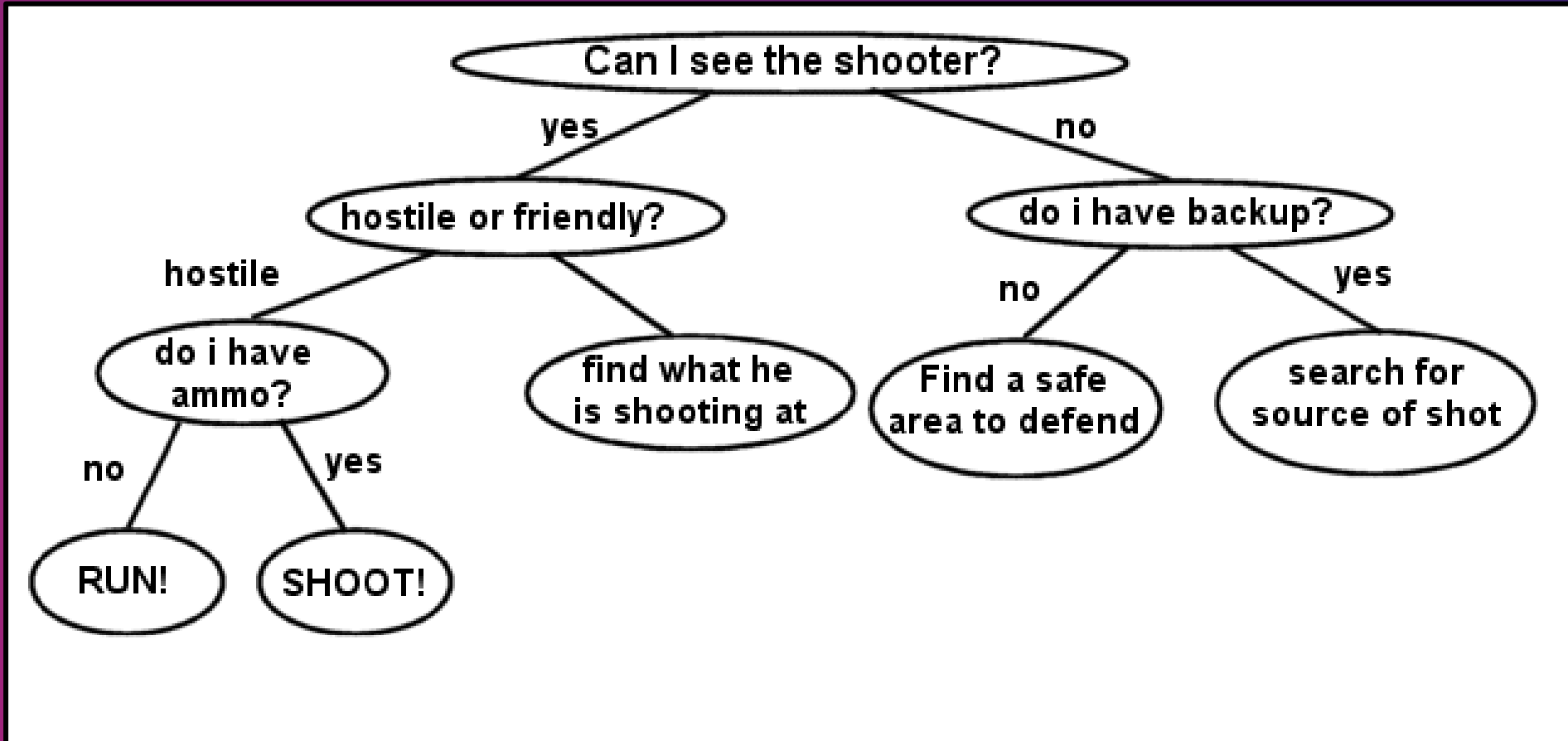


Metoda generowania próbek treningowych

- Wiele możliwości realizacji zadania
- Jedna wektoryzacja może mapować się wiele węzłów MCTS
- Problem identyfikacji właściwej akcji => pomocne statystyki MCTS (liczba wizyt)
- Uwzględnianie tylko najlepiej przesymulowanych węzłów
- Zapewnienie odpowiedniego treningu
 - dużo gier
 - dywersyfikacja sytuacji
 - uniknięcie przeuczenia

Uczenie offline

- Na podstawie próbek pobranych z rozgrywek uczone jest drzewo decyzyjne:



Czas obliczeń w docelowej grze (online)

- Znaczne przyspieszenie działania online
- Największy problem to poprawne wyuczenie modelu ML
- Metody search-based vs. learning-based

TABLE XV

THE FIRST COLUMN CONTAINS THE NUMBER OF MCTS ITERATIONS ALLOTTED PER EACH MOVE. THE REMAINING COLUMNS CONTAIN TIMES, IN MILLISECONDS, THAT THE RESPECTIVE AGENTS TAKE TO MAKE A SINGLE MOVE.

Experiment with 2000 repeats					
MCTS Iterations	Game 1			Game 2	
	MCTS	DT(90,30)	DT(1,1)	MCTS	XGB(33)
25K	220ms	<1ms	<1ms	256 ms	<1ms
50K	420ms	<1ms	<1ms	489 ms	<1ms
100K	792ms	<1ms	<1ms	1048 ms	<1ms
200K	1598ms	<1ms	<1ms	2144 ms	<1ms

Inne metody

Automaty skończone (Finite State Machines)

Ręczne skryptowanie

Dynamiczne skryptowanie (dynamic scripting)

Metody zachowań tłumu (crowd behavior)

Analityka

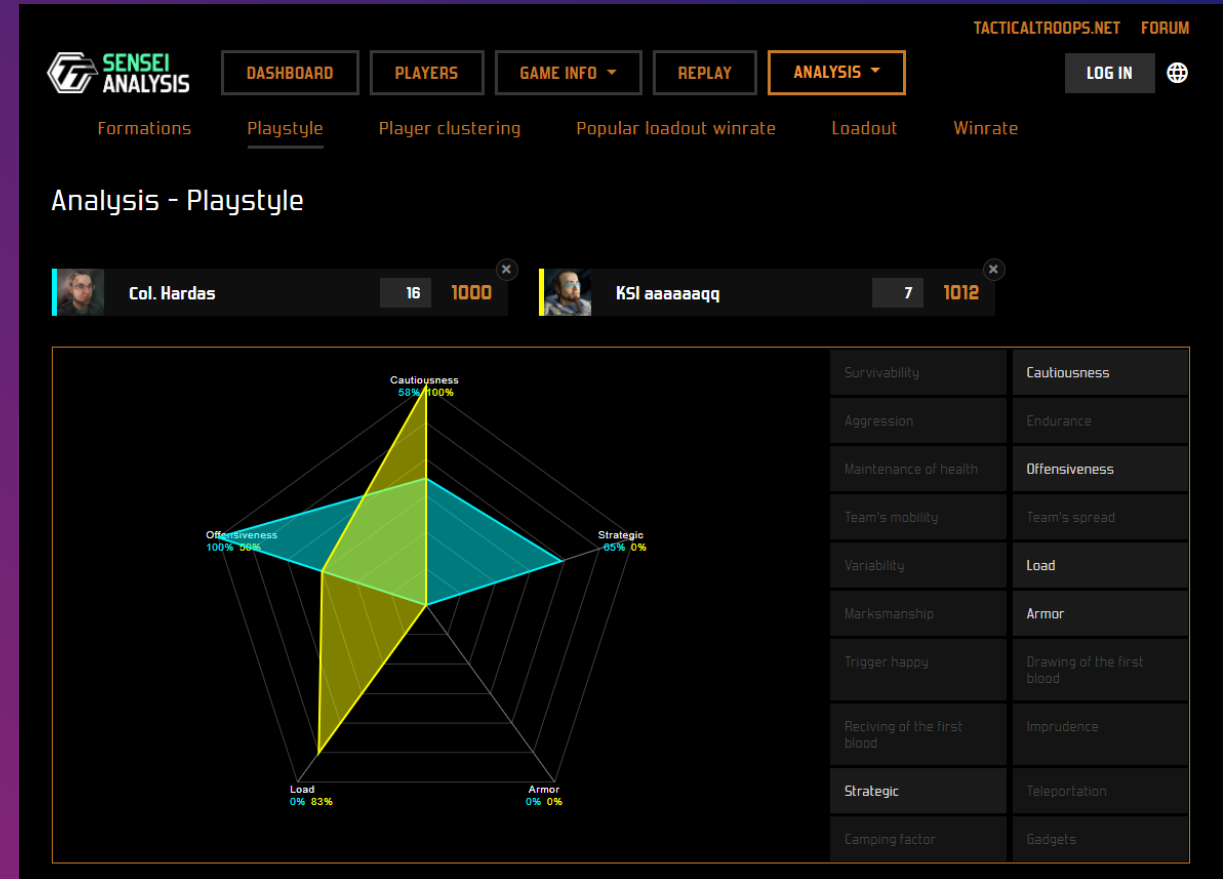


Analityka: slajd 1/3

- Klastrowanie graczy
- Wizualizacja podobieństw
- Określanie “profilu” gracza – stylu gry

Kolekcyjne gry karciane: (Clash Royale, Hearthstone)

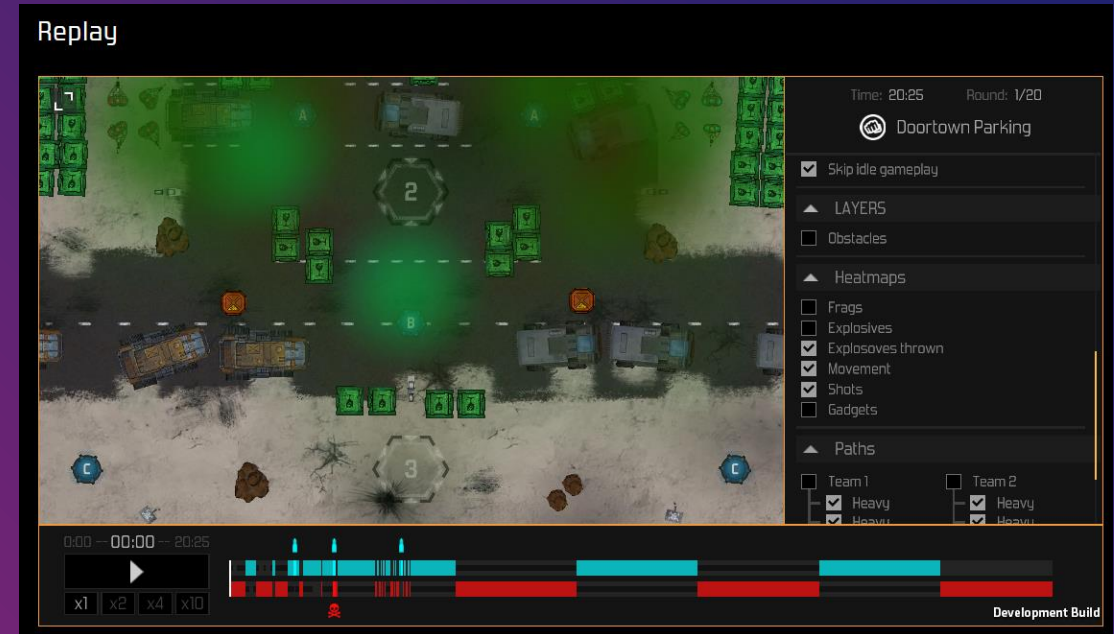
- Rekomendacje kart do nabycia
- Rekomendacje zamian kart
- Analiza metagame i wykrywanie archetypów
- Predykcja szans na wygraną (winrate model)



Analityka: slajd 2/3

Tactical Troops:

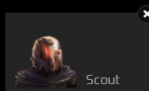
- Profilowanie graczy, obliczanie różnych TOP N
- Wszelkie proste statystyki
 - skuteczności, % wyborów poszczególnych rzeczy
- Odtwarzacz powtórek z gier
- Wykrywanie “game-changerów”
- Heatmapy
- Rekomendacje wyboru jednostek, broni i ekwipunku w grze (loadout) na podstawie preferencji gracza
- Sparametryzowana predykcja szans na wygraną
- Analiza metagame loadoutów
- Analiza wzajemnego grupowania jednostek na mapach
- Rozwój gracza w czasie: określanie głównych powodów zmiany rankingu gracza



SELECT MAP



UNITS

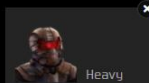


Scout

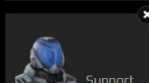
EQUIPMENT



cM400 Assault



Heavy



Support

Recommend Next

Recommend All

Analysis - Winrate

SELECT MAP



PLAYER 1



Rafaka Zabijaka

2

1085

PLAYER 2



So Drama

22

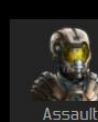
1000

GAME MODE

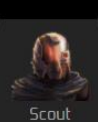
Domination

Devastator

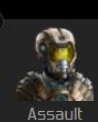
Predict



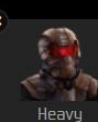
Assault



Scout



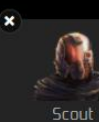
Assault



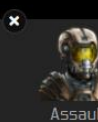
Heavy



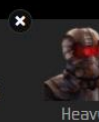
Support



Scout



Assault

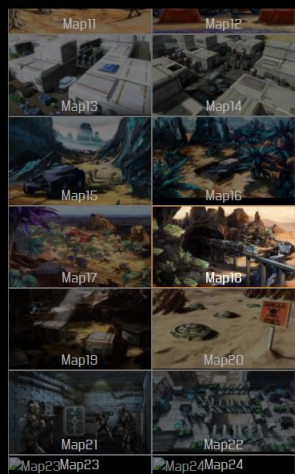


Heavy

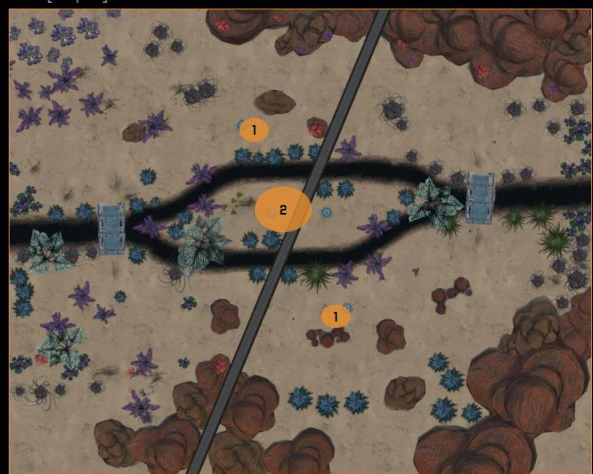
63% chance of Player 1 to win

Felix Barbarosa 18 1000

MAP LIST



MAP [Map18]



Tom Gunner

Look through Tom Gunner's stats and games



Tom Gunner

3

1068

Playing since:
2020-11-18

Games played:

24

Most played mode:

Domination

Kills / Deaths:

51 / 48 (2.13)

Avg. kills per game:

2.13

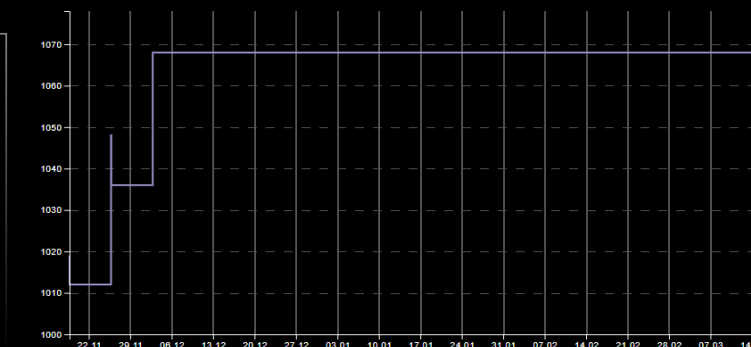
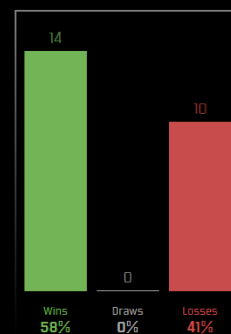
Damage dealt:

10219

Avg. deaths per game:

2.00

Total: 24



Analitika: slajd 3/3

Wybrane stosowane metody:

- UMAP
- kNN
- Sieci Neuronowe
- Logistic Lasso
- Metody NLP, Word2Vec, Bag-of-words, Latent Semantic Analysis
- Maximum Relevance Minimum Redundancy (MRMR)
- DBSCAN
- Lasy Losowe i XGBoost
- (TODO) Analiza koszykowa (affinity analysis)
- Tagowanie i Active Learning
- Proste statystyki

Podsumowanie



Podsumowanie - zastosowania AI

6 głównych zastosowań AI w grach video

- podejmowanie decyzji / sterowanie postaciami
- analityka gry
- analityka biznesowa
- PCG
- QA i testowanie
- rozwiązywanie określonych problemów np. nawigacji (pathfinding)

Podsumowanie

- AI w nauce vs. gamedev – trochę inne cele i oczekiwania
- Bardzo mały budżet obliczeniowy na AI w komercyjnych grach!
- Metody *state-of-the-art* z nauki raczej wciąż rzadko sterują botami
- powoli to się zmienia

Ważne aspekty:

- Spełnianie założeń gry; realizacja wizji projektantów
- Kontrola
- Testowalność
- Inżynierii oprogramowania:
łatwa implementacja, modyfikacja, utrzymanie, integracja z grą

Podsumowanie – braki komputerowych postaci

O czym nie było:

AI w grach video jest zwykle dość słabe lub oszukujące (cheating AI)

Często komputerowym postaciom brakuje:

- kreatywności
- adaptacji
- wnioskowania
- realizmu
- modelu pamięci i zmysłów

Powoli zmienia się mentalność twórców gier czym może być AI

Nowe wymagania graczy – jedna z szans na zmiany

Q & A



Bonus:
Poruszanie



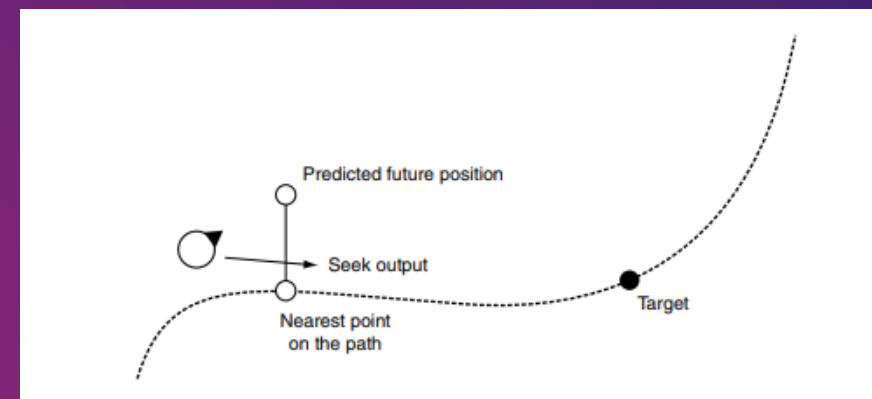
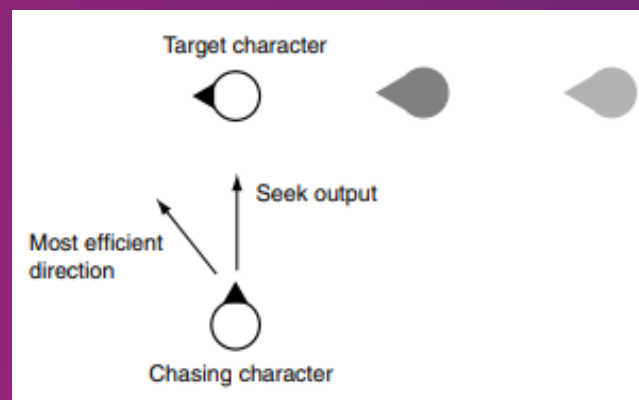
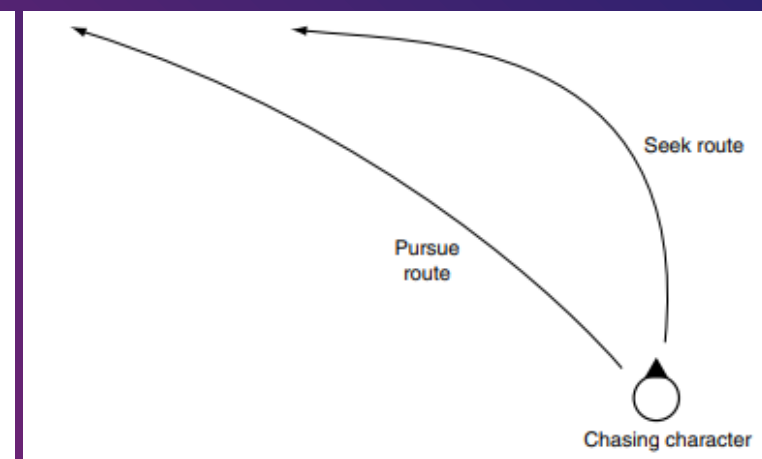
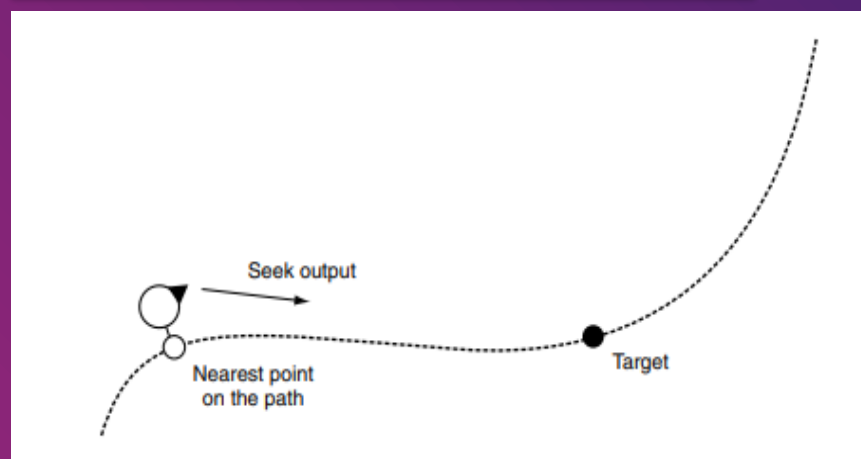
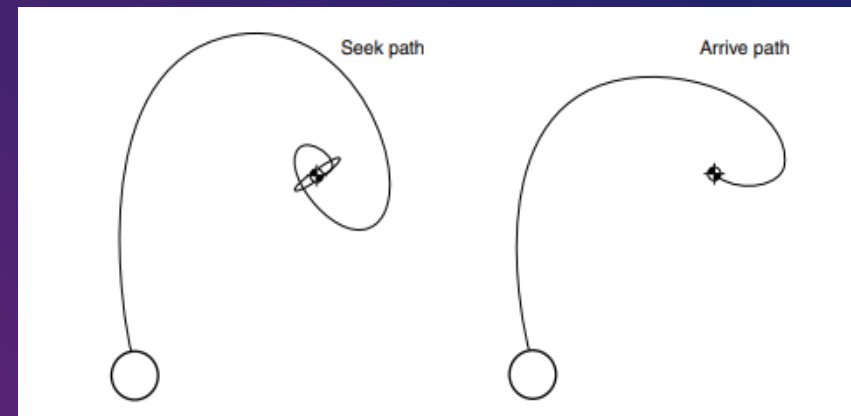
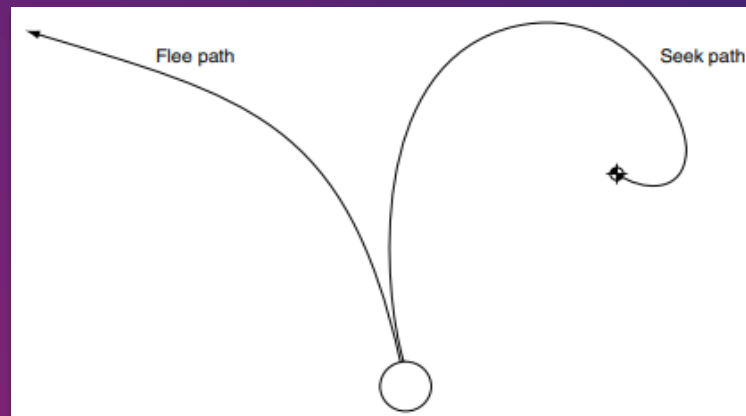
Poruszanie

- Wzorce ruchu
- Sterowanie i unikanie kolizji
- Nawigacja

Wzorce ruchu

Na przykład:

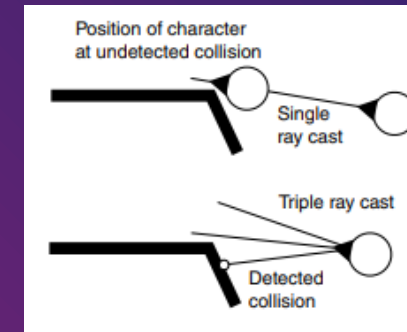
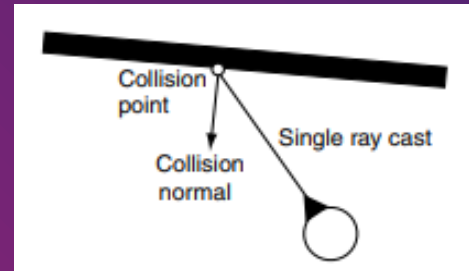
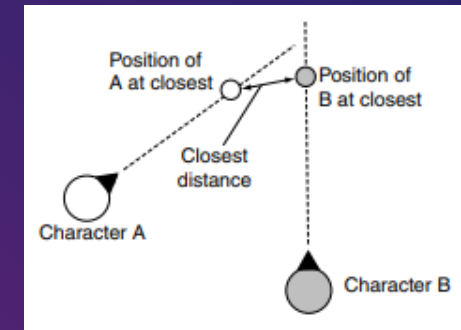
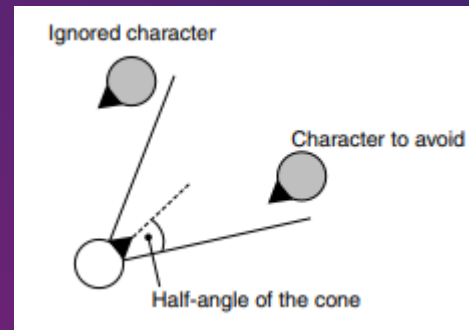
- Szukanie celu
- Uciekanie
- Podążanie za celem
- Podążanie za ścieżką



Rozwiązanie = proste skrypty

Sterowanie

- Np. unikalnie kolizji
- Czasem kontrolery PID
- Czasem Machine Learning



np. sieci neuronowe zastosowano w Colin McRae Rally i Forza Horizon 4



Nawigacja

Szukanie ścieżki z A do B

- generalnie najkrótszej

Czasem dodatkowo:

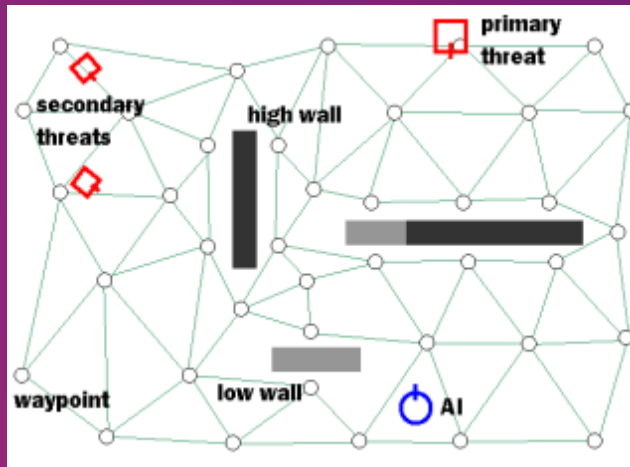
- naturalność
- bezpieczeństwo
- wykonywana dla wielu agentów jednocześnie
(sztuczki typu korzystanie ze znalezionych już ścieżek)

Nawigacja

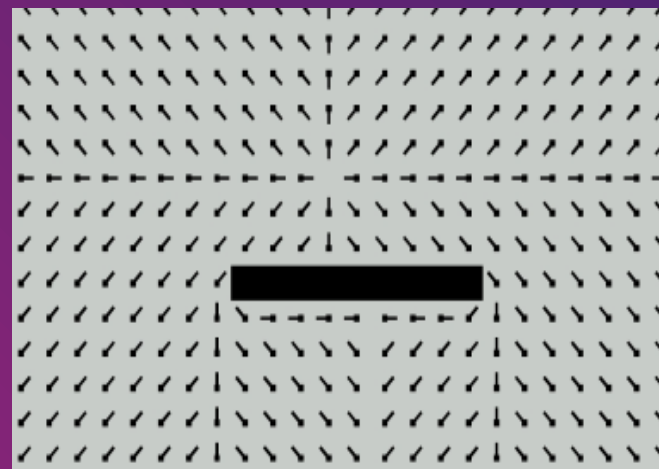
- Szukanie ścieżki z A do B
ale często uwzględnienie dodatkowych aspektów np. bezpieczeństwo
- Popularne reprezentacje:



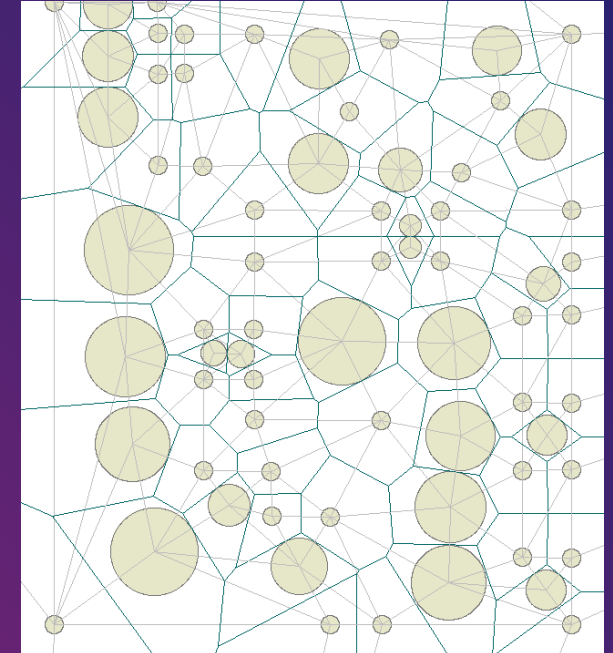
Siatka 2D



Graf punktów kontrolnych



Pole przepływów



NavMesh

Nawigacja

Algorytmy

- A*
- Hierarchical A*
- Jump Point Search (JPS)

Częste problemy:

- Związane z różnymi rozmiarami agentów
- Związane z naturalnością ruchu np. możliwość rotacji, zmiany kierunku
- Różne parametry fizyczne ruchu np. reakcja na stromość podejścia
- Różne parametry terenu np. płytkość, głęboka woda itp