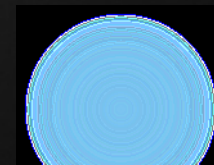


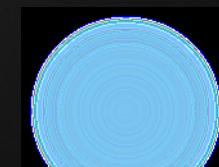
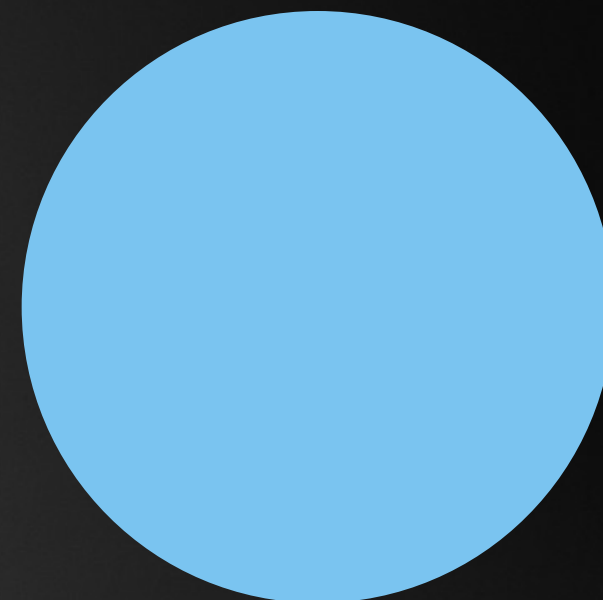
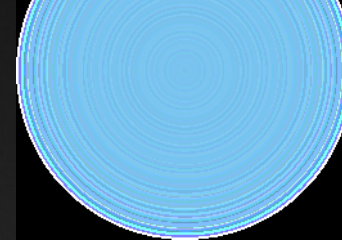
PV255 - Umělá inteligence ve hrách

Jiří Chmelík
podzim 2016



Osnova

- ▶ Úvod
- ▶ Vnímání
- ▶ Rozhodování
- ▶ Navigace
- ▶ Další
 - ▶ Skupinové chování
 - ▶ Artificial Stupidity
- ▶ Hry a výzkum v oblasti AI



Co bychom měli znát

▶ Grafy:

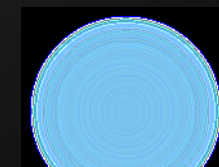
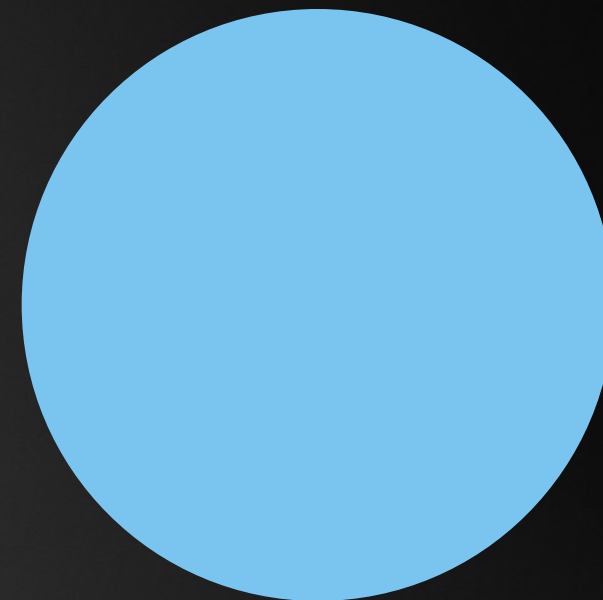
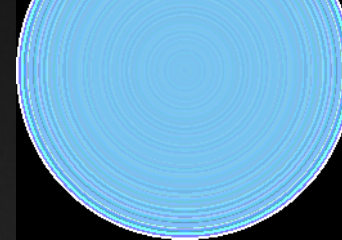
- ▶ grafové struktury,
- ▶ vyhledávání v grafu
 - ▶ nejkratší cesta – Dijkstrův al.

▶ Výpočetní geometrie

- ▶ Euklidovská vzdálenost
- ▶ Čtyř, osmi okolí

▶ Pravděpodobnost, statistika

▶ Složitost algoritmů:



AI obecně (academic AI)

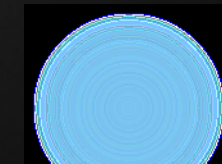
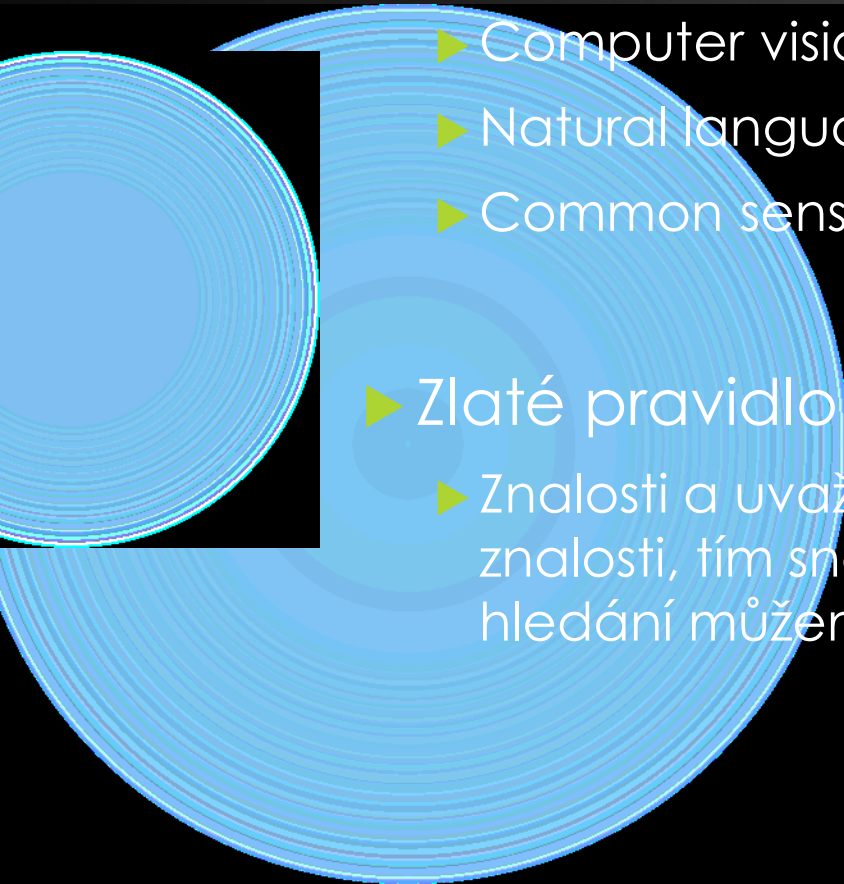
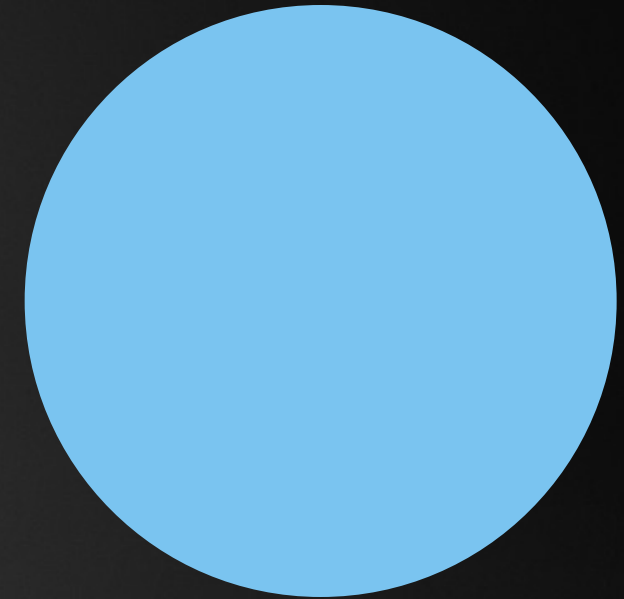
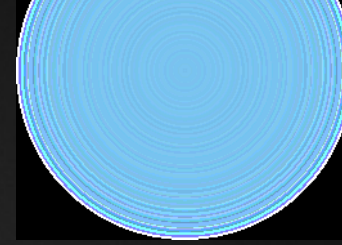
▶ Řešení problémů reálného života

▶ Témata:

- ▶ Computer vision
- ▶ Natural language processing
- ▶ Common sense reasoning

▶ Zlaté pravidlo AI:

- ▶ Znalosti a uvažování (hledání) jsou propojeny. Čím více máme znalosti, tím snadnější je najít správnou odpověď. (Čím více hledání můžeme provést, tím méně znalosti potřebujeme.)



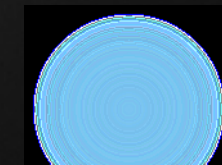
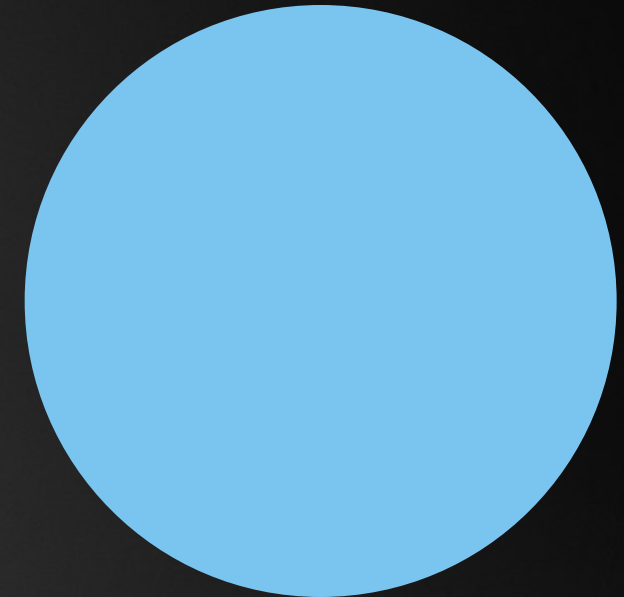
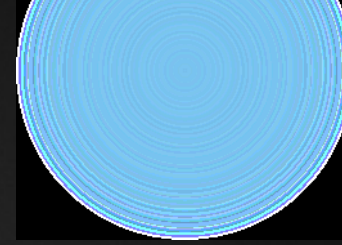
AI ve hrách

▶ Cíl:

- ▶ Podpořit zábavnost hry
- ▶ **Poskytnout správnou úroveň výzvy ve hře**
 - ▶ ani moc jednoduché,
 - ▶ ani moc náročné

▶ Hlavní omezení:

- ▶ Čas ve hře, velikost paměti
- ▶ Čas programátorů



AI ve hrách

▶ Postřehy:

- ▶ Složitější AI není vždy lepší (Black and White)
- ▶ Používáme jak přesné algoritmy, tak heuristiky, i různé hacky
- ▶ AI se těžko ladí – chybí přímý vizuální výstup (co se NPC děje „v hlavě“, pomalé testování)

▶ Různé AI pro různé charaktery

▶ Co udělá postava v temné místnosti když uslyší zvuk?

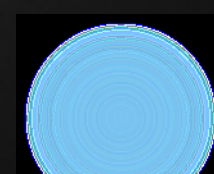
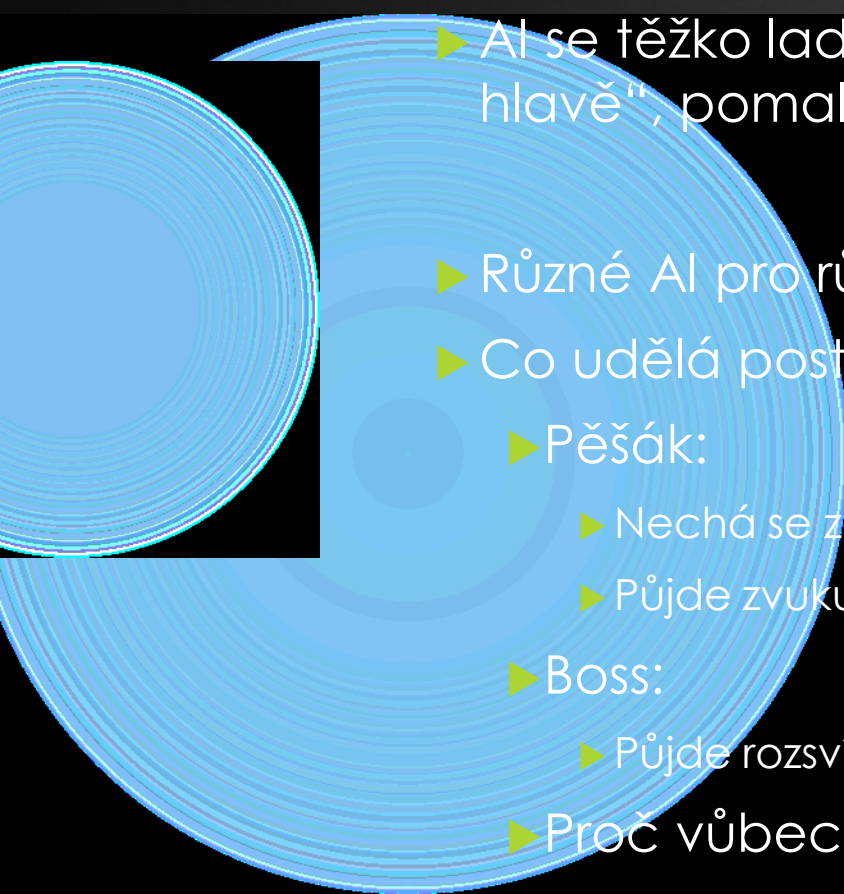
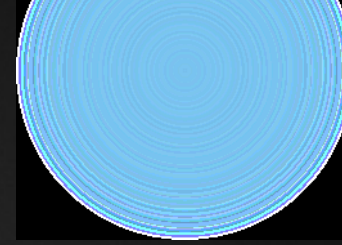
▶ Pěšák:

- ▶ Nechá se zastřelit
- ▶ Půjde zvuku naproti

▶ Boss:

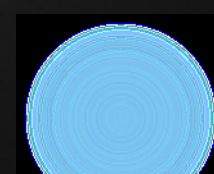
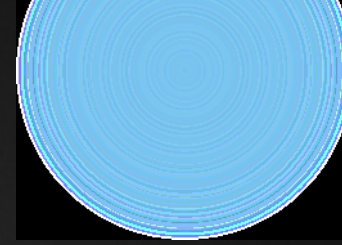
- ▶ Půjde rozsvítit

▶ Proč vůbec stojí postava v místnosti ve tmě?



Základní pojem – NPC

- ▶ non-player characters (NPC) – jakákoli samostatně se chovající entita:
 - ▶ MMORPG charaktery
 - ▶ „boti“ (FPS)
 - ▶ Auta, tanky, letadla, vesmírné lodě neřízené hráčem
 - ▶ Obranné věže, útočící jednotky
 - ▶ ...
 - ▶ Virtuální kameraman (Tomb Raider, Uncharted)
 - ▶ AI hrající RTS – rozhoduje, těží, staví, bojuje



AI ve hrách

Game Genres

Action
Role playing
Adventure
Strategy games
God games
Team sports
Individual sports



AI Entity Roles

Tactical enemies
Partners
Support characters
Story directors
Strategic opponents
Units
Commentators

AI Research Problems

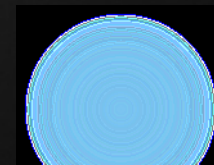
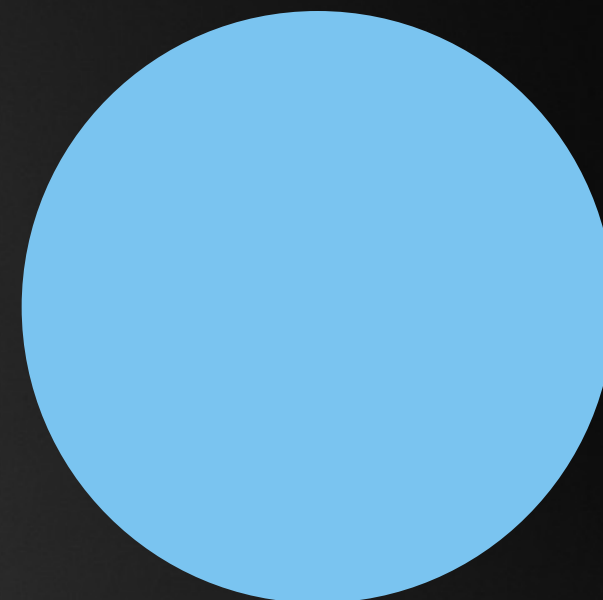
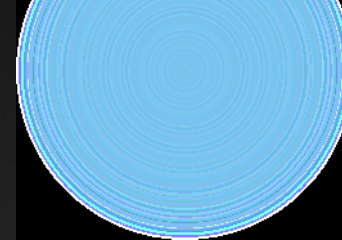
Interact with environment
Fast response
Realistic sensing
Adapt to environment
Interact with humans
Adapt to human player
Difficulty adaptation
Strategic adaptation
Interact with other AIs
Coordinate behavior
Navigation
Use tactics and strategies
Allocate resources
Understand game flow
Humanlike responses
Reaction times
Realistic movement
Emotions
Personalities
Low computational overhead
Low development overhead

AI Research Areas

High level perception
Commonsense reasoning
Natural language
Speech processing
Gesture processing
Planning & counterplanning
Cognitive modeling
Plan recognition
Soft real-time response
Reactive behavior
Teamwork
Scheduling
Path planning
Spatial reasoning
Temporal reasoning
Opponent modeling
Learning
Knowledge acquisition

Osnova

- ▶ Úvod
- ▶ **Vnímání**
- ▶ Rozhodování
- ▶ Navigace
- ▶ Další
 - ▶ Skupinové chování
 - ▶ Artificial Stupidity
- ▶ Hry a výzkum v oblasti AI



Vnímání okolí (Perception; sensor system)

„If two guards are standing talking to each other and you shoot one down, the other guard shouldn't carry on the conversation.“

[AI for Games]

- ▶ NPC potřebují vědět, co se děje kolem nich:
 - ▶ Neměli by narážet do zdí, do sebe navzájem, padat z útesů, ...
 - ▶ Neměli by podvádět („vidět“ hráče za zdí)
- ▶ Příklad: FPS aréna – vidíte hráče, který vbíhá do malé místnosti s jediným vchodem. Máte v ruce raketomet...

Vnímání okolí (Perception; sensor system)

- ▶ Hráč vnímá hru skrze rozhraní
 - ▶ Největší část informace se přenáší skrze obrazovku
 - ▶ Informace o herním světě získáváme pomocí technik zpracování obrazu
 - ▶ Řada technik a fíglů, které ovlivňují „vidění“:
 - ▶ Informace o objektech (perspektiva, motion parallax, ...)
 - ▶ Změna „kvality“, způsobu vidění:
 - ▶ Přírozené: tetelení vzduchu, přechod mezi světlou a tmavou oblastí, po zásahu, po výbuchu granátu, počasí, ...
 - ▶ Umělé – herní mechaniky: baterka, night vision, sonar, ...

Vnímání okolí (Perception; sensor system)

- ▶ Hráč vnímá hru skrze rozhraní
- ▶ NPC vnímají hru skrze vlastní rozhraní (simulated perception)
 - ▶ Co všechno (jaké informace) jim toto rozhraní poskytne?
 - ▶ Jak rychle, v jaké kvalitě?
- ▶ Game-state, game-manager
 - ▶ Ví „všechno“ o hře (až na vstupy reálných hráčů)
 - ▶ Poskytuje pouze něco

Vnímání okolí (Perception; sensor system)

▶ Základní techniky jak NPC získávají data:

▶ Dotazování (Polling)

- ▶ NPC se pravidelně (každý snímek) ptá herního světa...
- ▶ jednoduché na implementaci,
- ▶ může být náročné na zdroje,
- ▶ špatně škálovatelné

▶ Zasílání zpráv (Events System)

- ▶ složitější na implementaci,
- ▶ může být pomalejší,
- ▶ Jednodušší na debug
 - ▶ můžu si všechny zprávy vypsat
 - ▶ Můžu zprávy vyvolávat uměle a jednoduše kontrolovat, jestli NPC reagují správně

▶ Příklad: `Update ()` vs. `OnMouseButtonDown ()`

Dotazování vs. zasílání zpráv

- ▶ V praxi – obojí:

- ▶ Teoreticky nejrychlejší,
- ▶ vyžaduje extra kód,

- ▶ Příklad – NPC hlídač:

- ▶ Dokud hráč není v (místnosti / oblasti) – NPC čeká na zprávu
- ▶ Jakmile je v dohledu, NPC dostává zprávu a začne se pravidelně dotazovat

Dotazování vs. zasílání zpráv

▶ Rozšíření Polling: Polling stations

▶ Mezipaměť pro informace, které:

- ▶ je složité spočítat

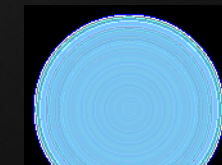
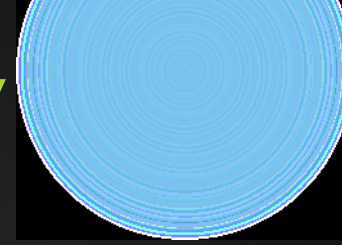
- ▶ hodí se často, hodí se více hráčům

▶ Při dotazování:

- ▶ Pokud je informace v mezipaměti a není příliš stará – použij ji

- ▶ Jinak – spočítej a ulož do mezipaměti.

- ▶ Umožňuje snazší monitorování a debug



Dotazování vs. zasílání zpráv

▶ Zasílání zpráv: manažer událostí (event manager)

- ▶ Centralizované řešení pro zpracování událostí a rozesílání zpráv zainteresovaným NPC

▶ Části:

▶ Checking engine

- ▶ Game-state, detekce kolizí,
- ▶ Některé události nemusí být kontrolovány – generují se přímo:
 - ▶ uživatelský vstup, ...
 - ▶ Změna stavu ve „FSM“ (dále)

▶ Event queue

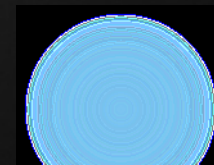
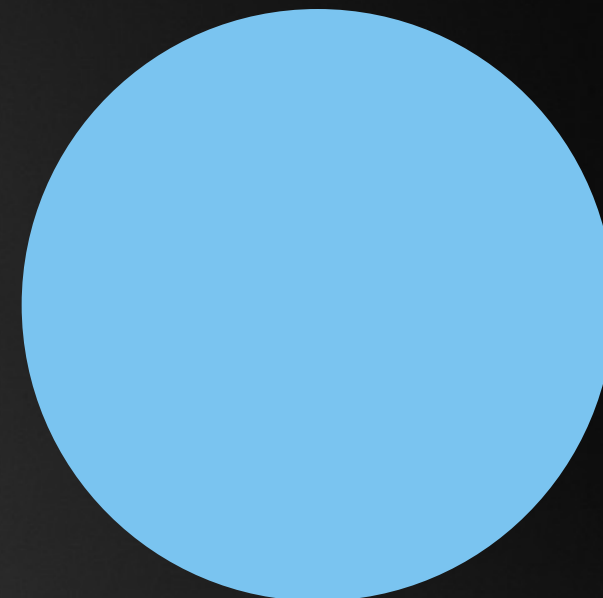
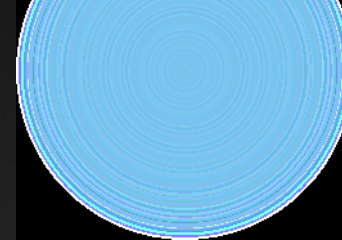
- ▶ Vyvažování zátěže

▶ Registry of event recipients

▶ Event dispatcher

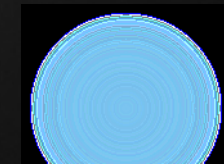
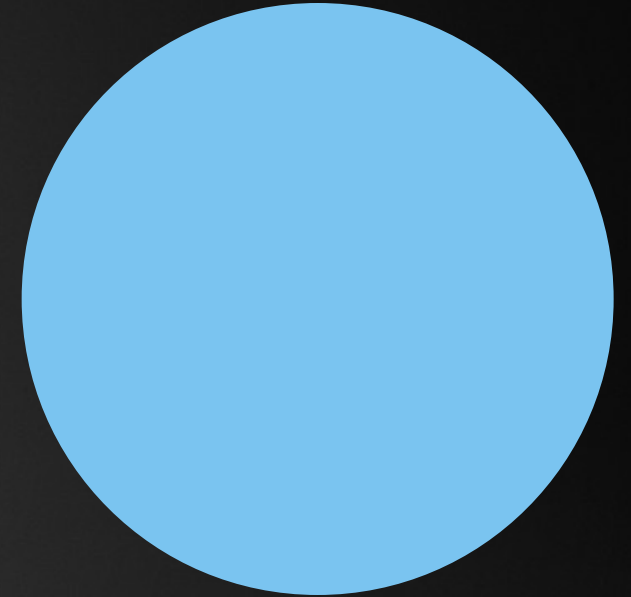
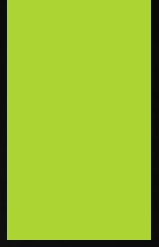
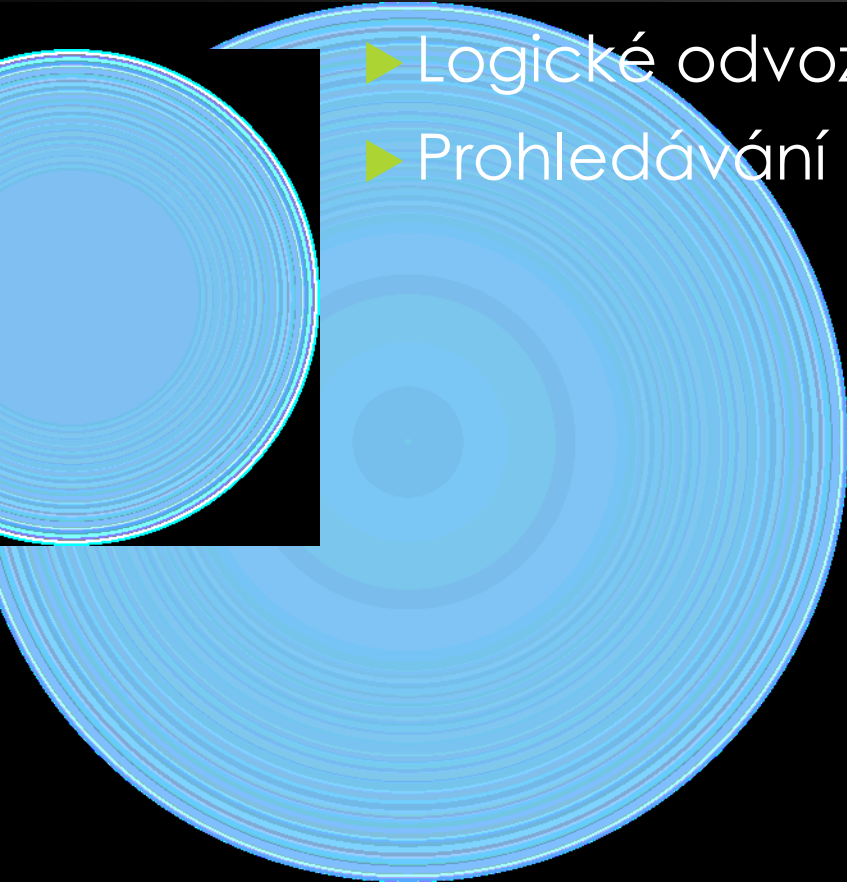
Osnova

- ▶ Úvod
- ▶ Vnímání
- ▶ **Rozhodování**
- ▶ Navigace
- ▶ Další
 - ▶ Skupinové chování
 - ▶ Artificial Stupidity
- ▶ Hry a výzkum v oblasti AI



Základní rozhodovací techniky

- ▶ Rozhodovací strom
- ▶ Konečný stavový automat
- ▶ Produkční systém
- ▶ Logické odvozování
- ▶ Prohledávání



Konečné automaty (FSM)

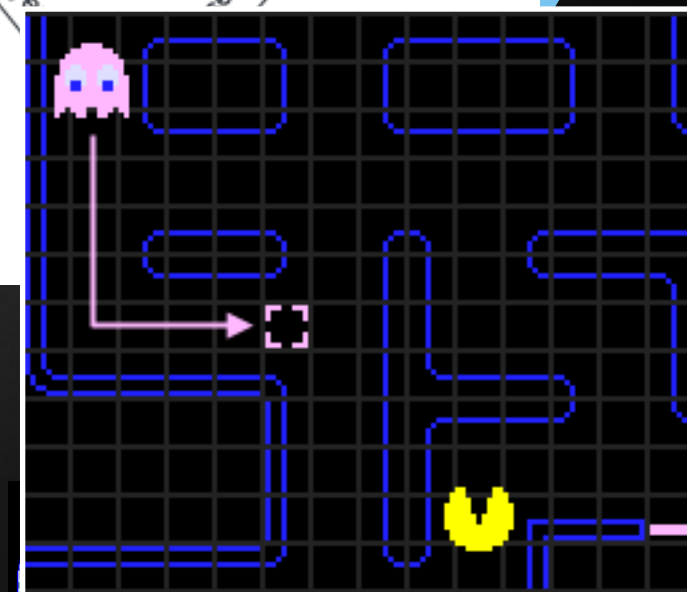
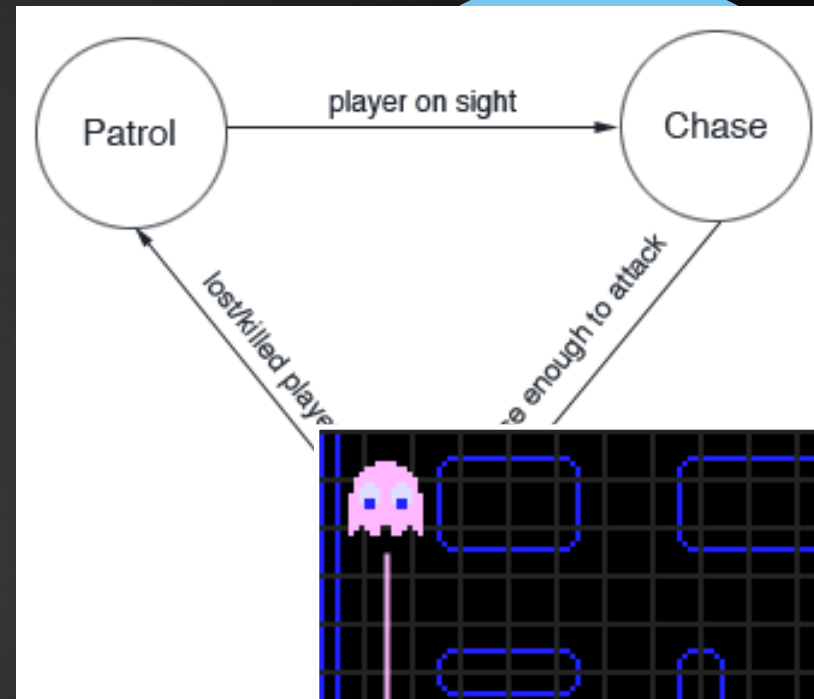
▶ Entita je v právě jednom z konečného množství stavů

▶ FSM:

- ▶ Stavy
- ▶ Přečody
- ▶ Pravidla
- ▶ Události

▶ Pac-Man (1979)

- ▶ Tři stavy, každý z duchů má jiné AI



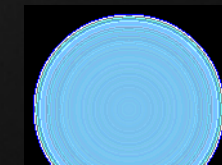
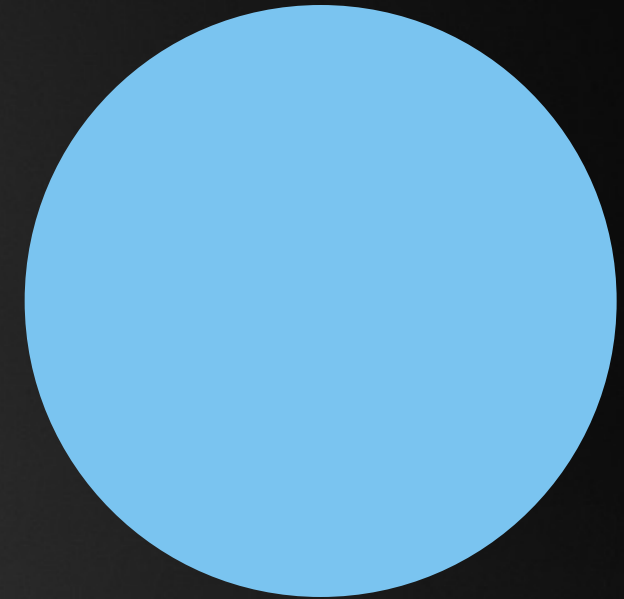
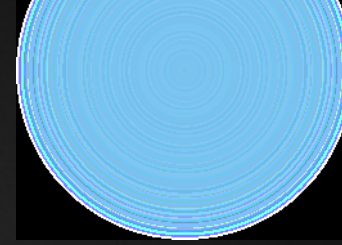
Konečné automaty (FSM)

▶ Výhody

- ▶ Jednoduše implementovatelné
- ▶ Nástroje pro snazší návrh (i implementaci)

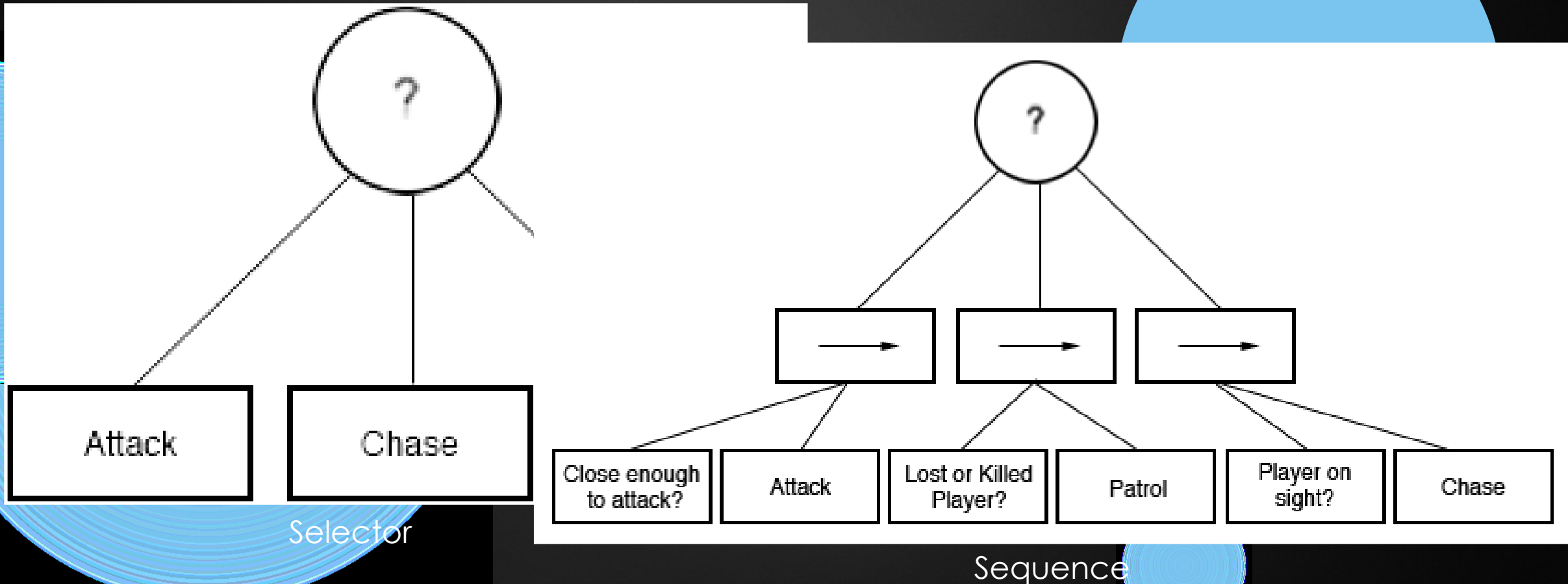
▶ Nevýhody

- ▶ Špatně škálovatelné řešení
- ▶ Ideální pro menší, jednodušší systémy
- ▶ Možné rozšíření – hierarchie FSM



Rozhodovací stromy (Behavior Trees)

- ▶ Snadno škálovatelné
- ▶ Složitější implementace



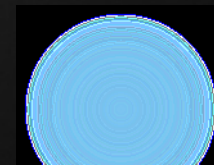
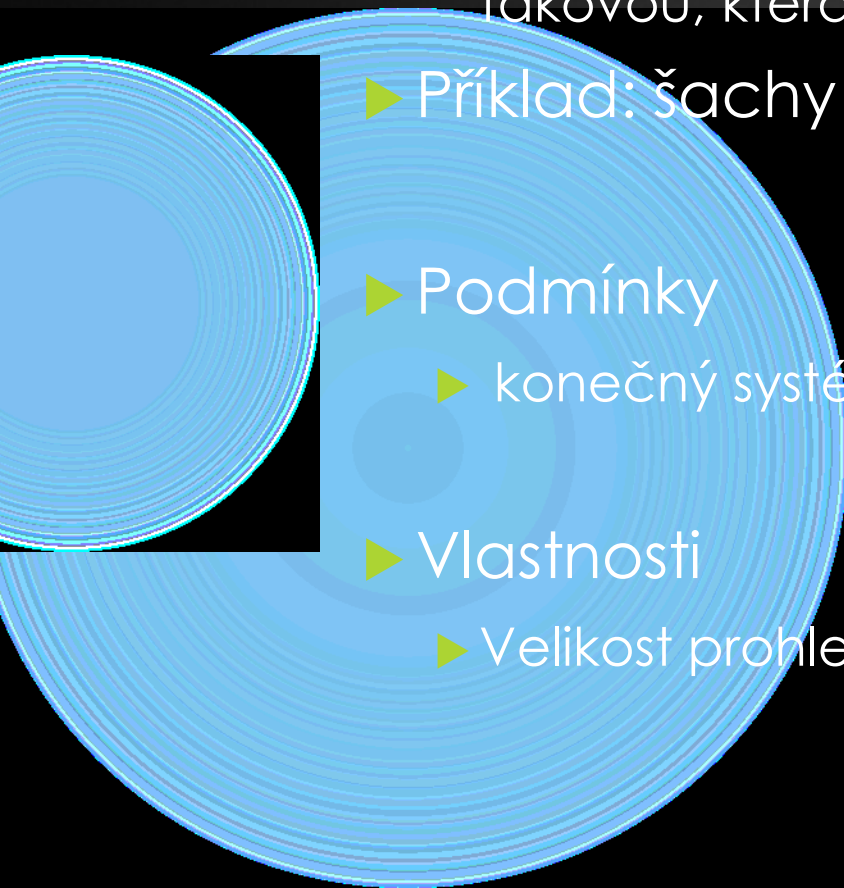
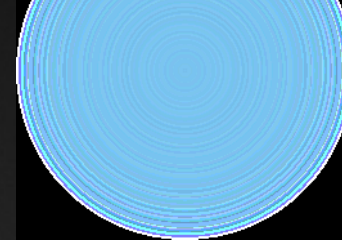
Rozhodovací stromy (Behavior Trees)

▶ Existující implementace:

- ▶ Cryengine,
- ▶ Unity (plug-iny: Behave, Behavior Designer), Unity 5
- ▶ Unreal Engine
- ▶ Rozšíření – Modulární Behaviorální stromy
 - ▶ Kingdom Come: <https://youtu.be/QewizbnbPZg>

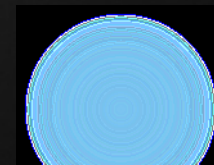
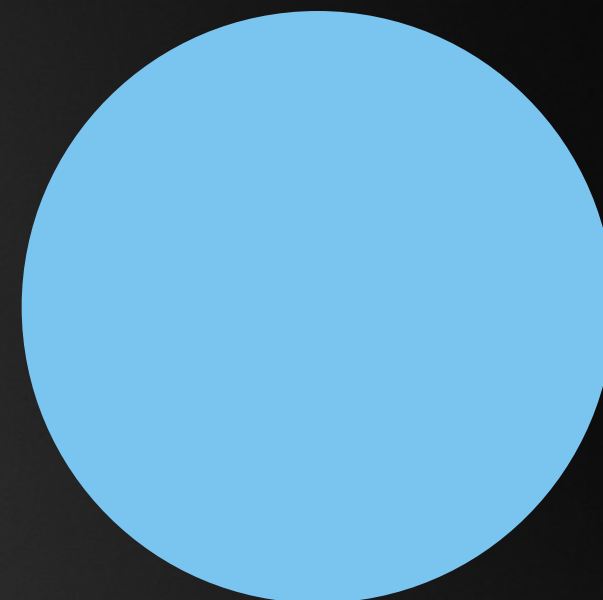
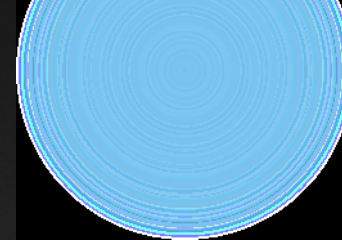
Prohledávání (Searching)

- ▶ Dopředu otestuji všechny možné (bezprostřední) akce a vyberu tu nejlepší
 - ▶ Pokud jsem nenašel cíl, testuji sekvencí více akcí a hledám takovou, která vede k cíli
- ▶ Příklad: šachy
 - ▶ Podmínky
 - ▶ konečný systém
 - ▶ Vlastnosti
 - ▶ Velikost prohledávacího stromu roste exponenciálně



Osnova

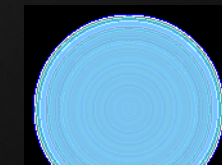
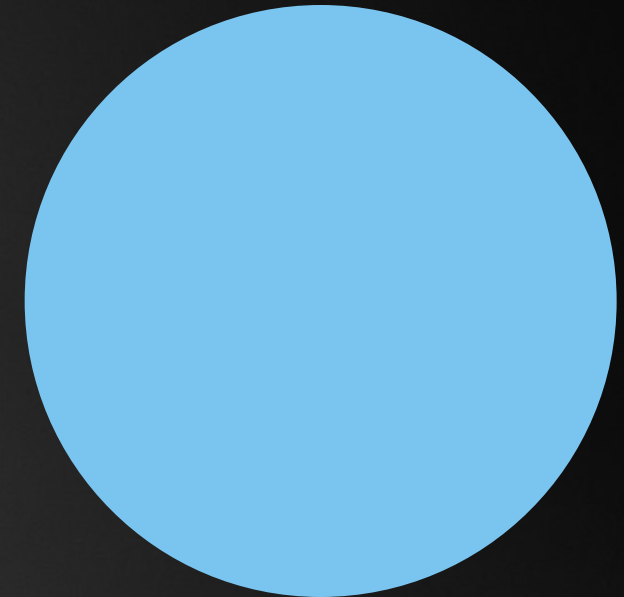
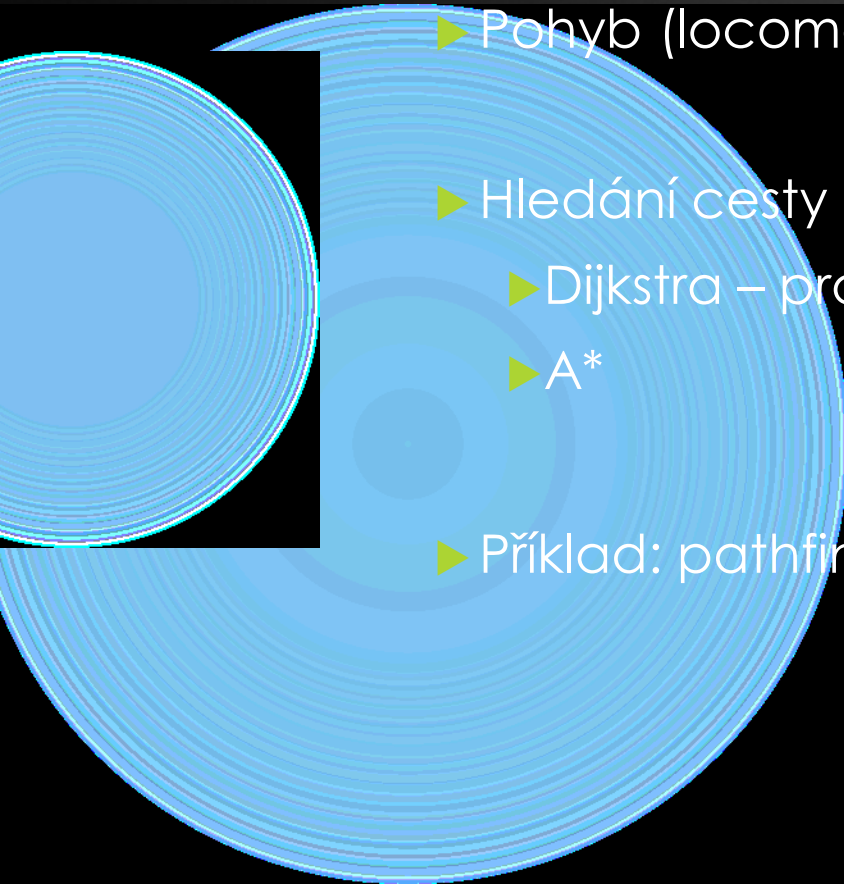
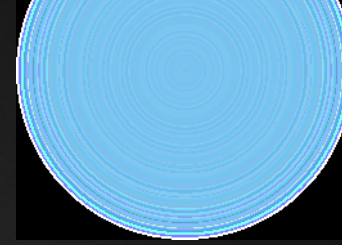
- ▶ Úvod
- ▶ Vnímání
- ▶ Rozhodování
- ▶ **Navigace**
- ▶ Další
 - ▶ Skupinové chování
 - ▶ Artificial Stupidity
 - ▶ Hry a výzkum v oblasti AI



Navigace

Více-úrovňový problém:

- ▶ Výběr akce (strategie, taktika, cíle, plánování)
- ▶ Řízení (steering) – výběr akce
- ▶ Pohyb (locomotion)
- ▶ Hledání cesty (Pathfinding)
 - ▶ Dijkstra – proč ne?
 - ▶ A*
- ▶ Příklad: pathfinding vs. fog of war (Age of Empires)



Algoritmus A*

▶ [: ei stár :]

▶ Jednoduchý princip, implementace může být zapeklitá

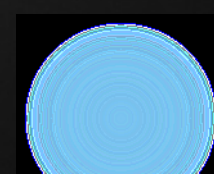
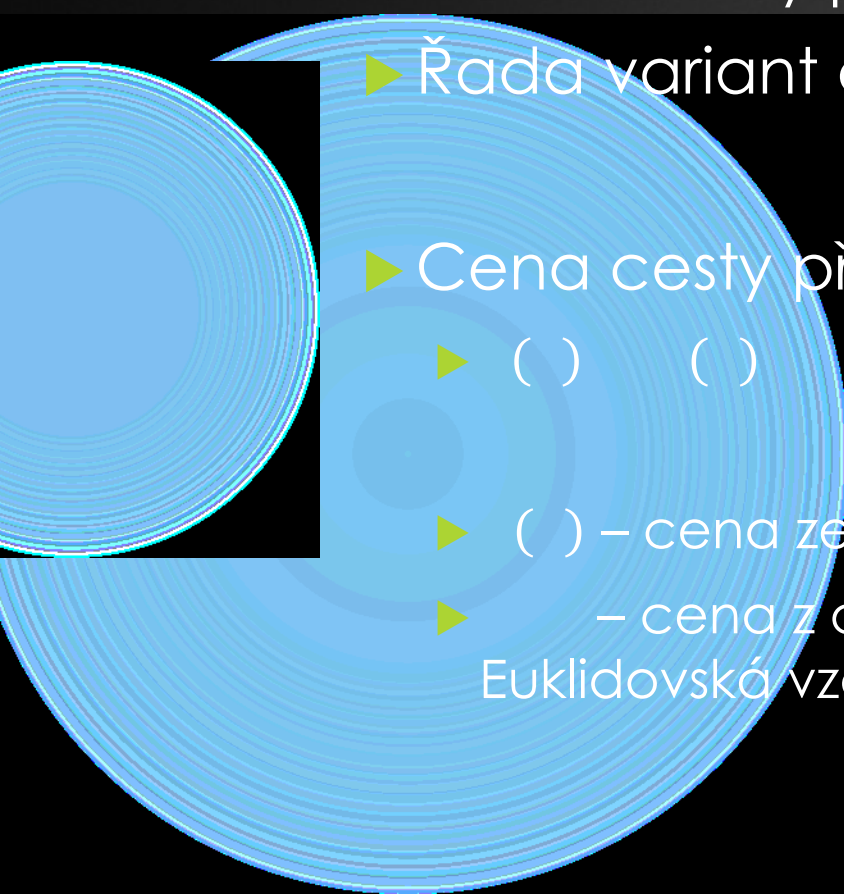
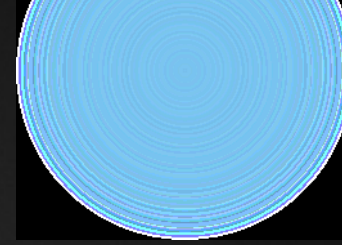
▶ Řada variant algoritmu, řada existujících řešení

▶ Cena cesty přes uzel x

▶ $f(x)$ $g(x)$ $h(x)$

▶ $g(x)$ – cena ze startu do aktuálního uzlu – lze spočítat přesně

▶ $h(x)$ – cena z aktuálního uzlu do cíle – nutno odhadnout (např. Euklidovská vzdálenost)

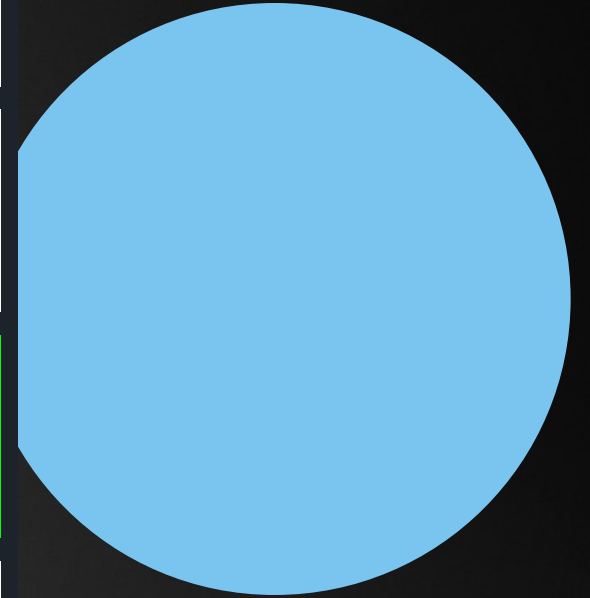
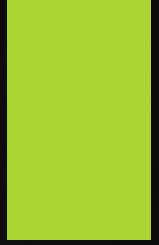
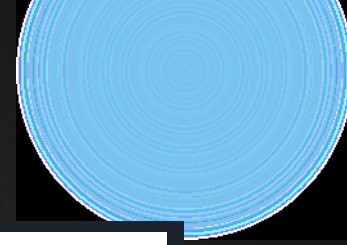
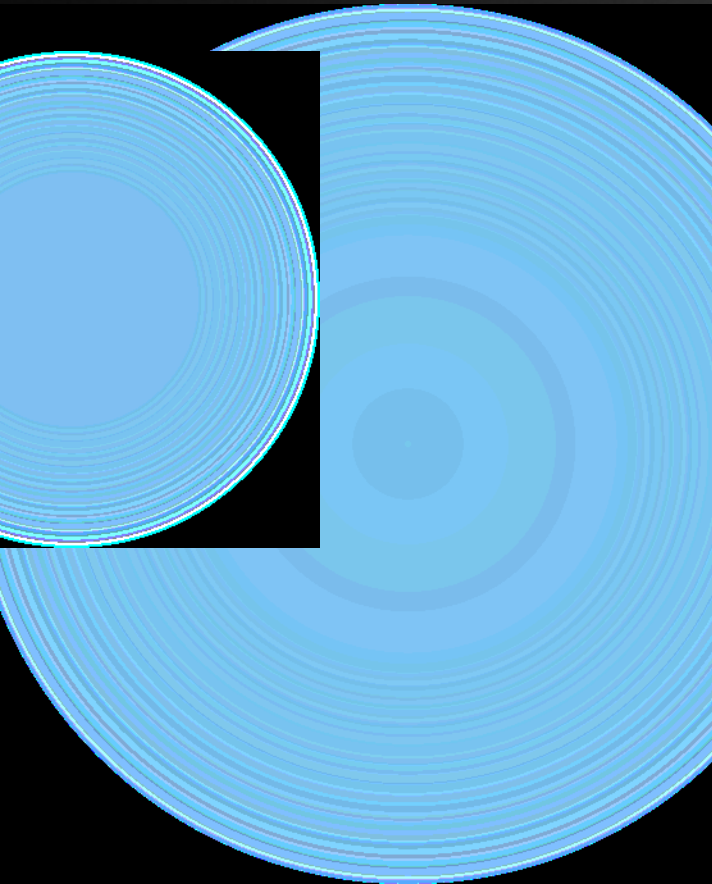
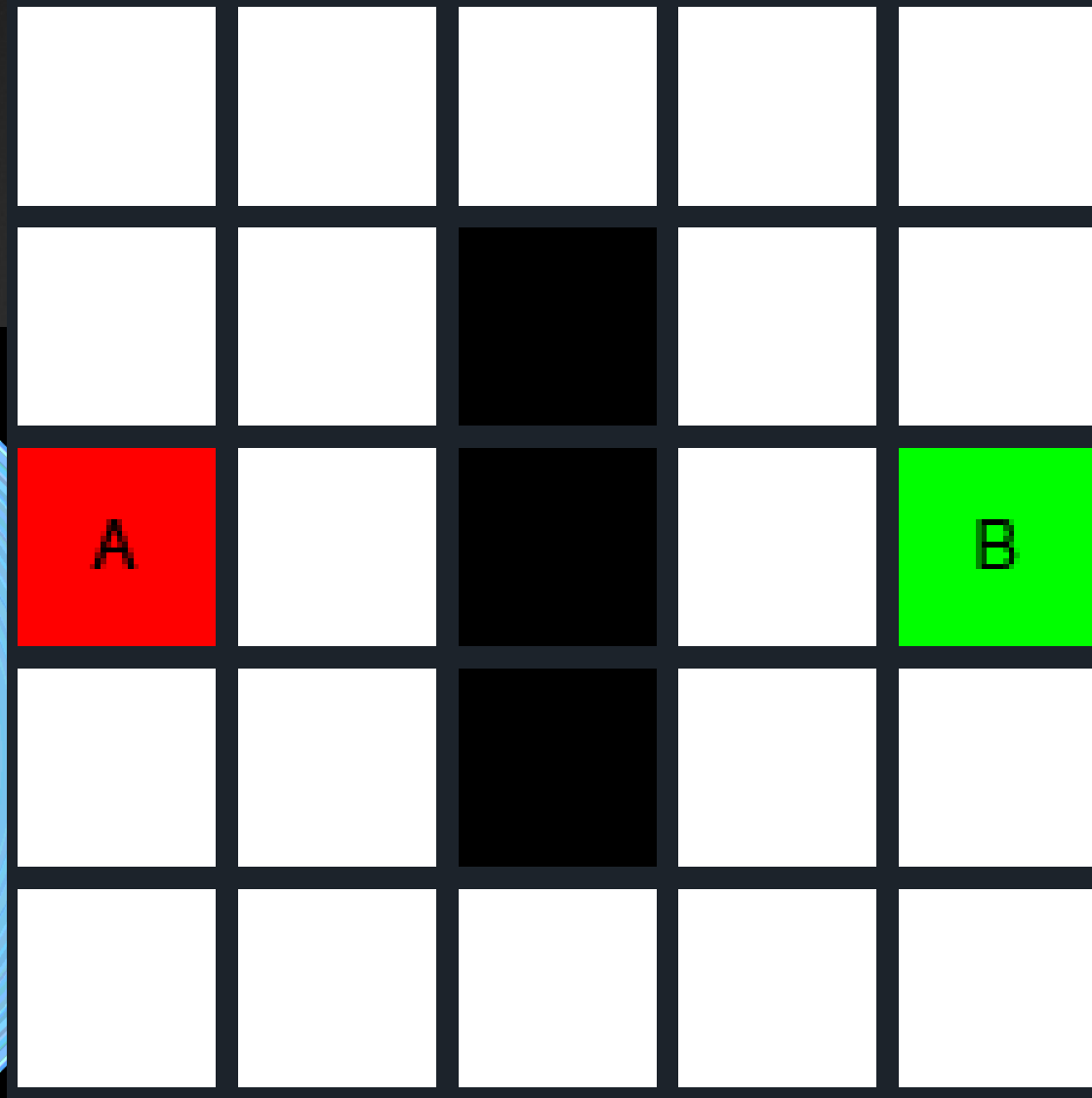


Algoritmus A*

Cena cesty:

- ▶ Vzdálenost (metrika Manhattan)
- ▶ Obtížnost terénu (brodit se přes řeku nebo oběhnout přes vzdálený most?)
- ▶ Přítomnost obranných věží, snadné cíle pro odstřelovače,
- ▶ Adaptivní – počet mrtvých členů týmu na cestě
- ▶ ...

Algorithmus A*



Algorithmus A*

0		0		0		0		0	
1	0	1	0	1	0	1	0	1	0
6		6		0		0		0	
1	5	2	4	1	0	1	0	1	0
0		4		0		0		0	
1	0	1	3	1	0	1	0	1	0
6		6		0		0		0	
1	5	2	4	1	0	1	0	1	0
0		0		0		0		0	
1	0	1	0	1	0	1	0	1	0

Algorithmus A*

8	0	0	0	0	0	0	0		
2	6	1	0	1	0	1	0	1	0
6	6	0	0	0	0	0	0		
1	5	2	4	1	0	1	0	1	0
0	4	0	0	0	0	0	0		
1	0	1	3	1	0	1	0	1	0
6	6	0	0	0	0	0	0		
1	5	2	4	1	0	1	0	1	0
0	0	0	0	0	0	0	0		
1	0	1	0	1	0	1	0	1	0

Algorithmus A*

8	8	8	8	8	8	8	8		
2	6	3	5	4	4	5	3	6	2
6	6	6	0	0	0	8	8	8	8
1	5	2	4	1	0	6	2	7	1
0	4	4	0	0	0	8	8	8	8
1	0	1	3	1	0	7	1	8	0
6	6	6	0	0	0	8	8	8	8
1	5	2	4	1	0	6	2	7	1
8	8	8	8	8	8	8	8	8	8
2	6	3	5	4	4	5	3	6	2

Algorithmus A*

8		8		8		8		8	
2	6	3	5	4	4	5	3	6	2
6		6		0		8		8	
1	5	2	4	1	0	6	2	7	1
0		4		0		8		8	
1	0	1	3	1	0	7	1	8	0
6		6		0		8		8	
1	5	2	4	1	0	6	2	7	1
8		8		8		8		8	
2	6	3	5	4	4	5	3	6	2

Graf zastávek (Waypoint Graph)

- ▶ Zjednodušení mapy na graf zastávek

- ▶ Hledání nejlepší cesty v grafu

- ▶ Málo uzlů:

- ▶ Nuda

- ▶ Riziko kolize

- ▶ Hodně uzlů – větší nároky:

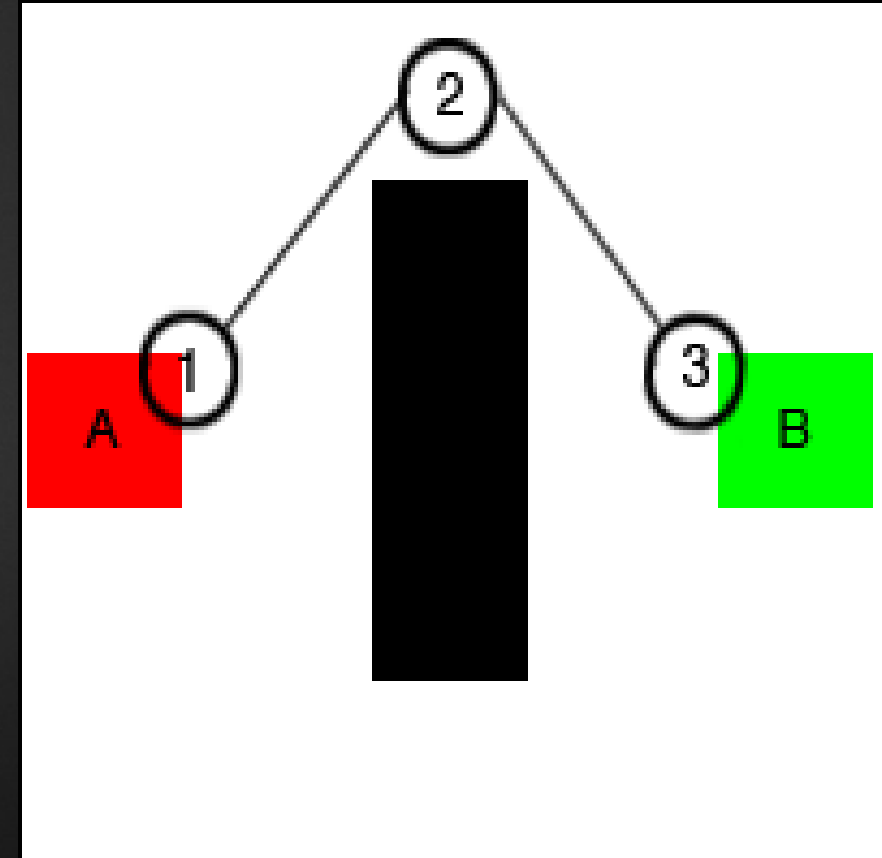
- ▶ Na tvorbu

- ▶ Na paměť

- ▶ Na procesor

- ▶ Neřeší velikost NPC

- ▶ Pěšák vs. tank



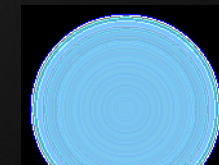
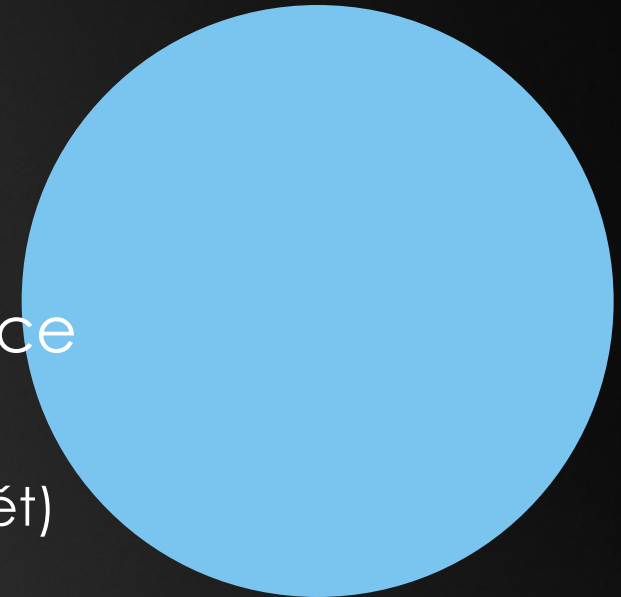
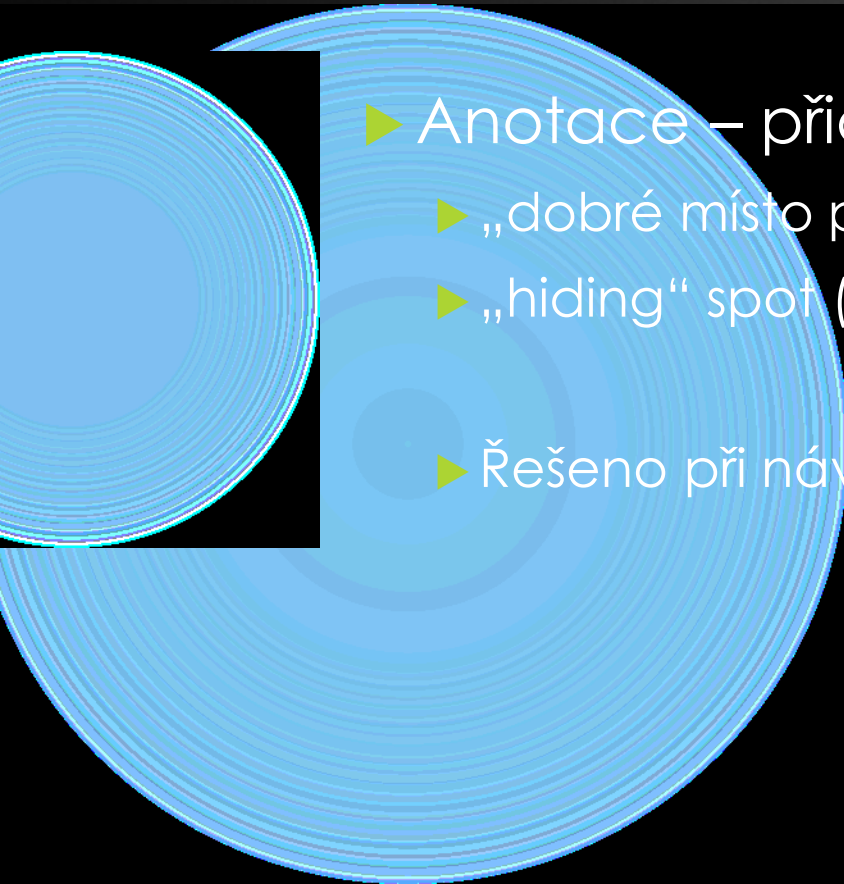
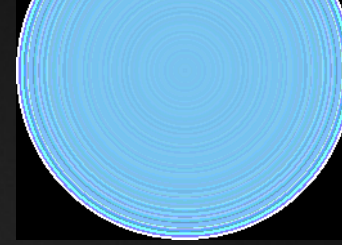
Waypoints (zastávky?)

▶ Vytváření:

- ▶ Obvykle návrhář úrovní
- ▶ Automatické systémy

▶ Anotace – přidání dalších informací k zastávce

- ▶ „dobré místo pro odstřelování“
- ▶ „hiding“ spot (z kterých jiných zastávek je/není vidět)
- ▶ Řešeno při návrhu => zjednodušuje/urychluje rozhodování v real-time



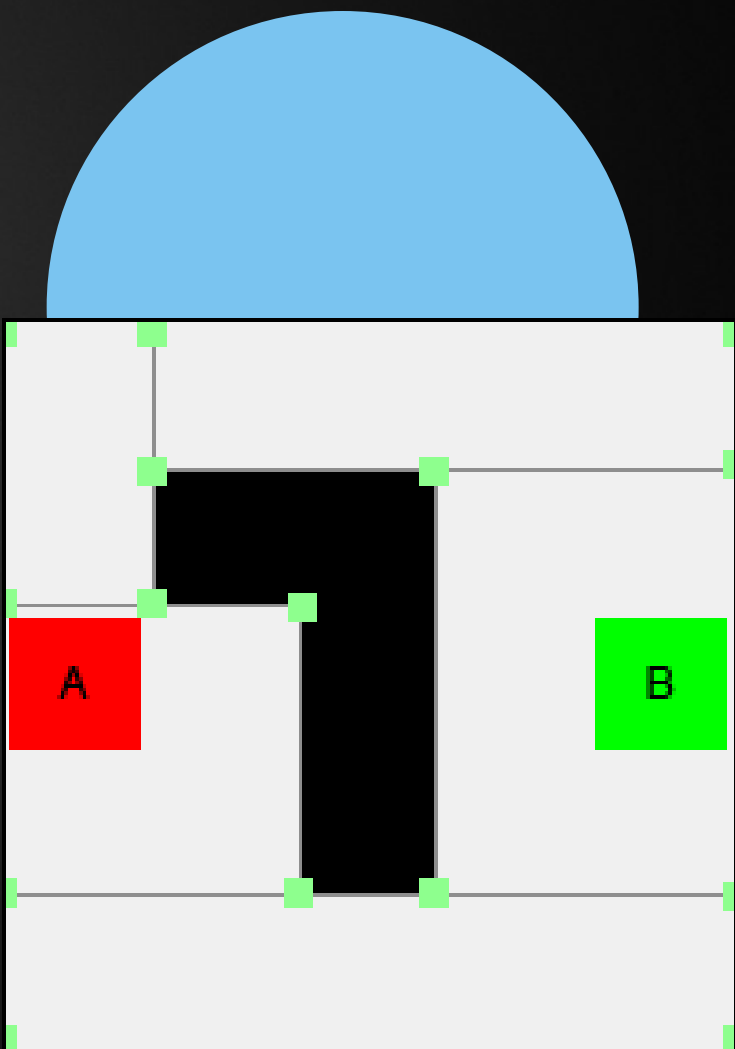
Navigační síť (Navigation Mesh)

Grafová struktura pro zjednodušení „mapy“

- ▶ Rozdělení na konvexní polygony
- ▶ Poskytuje více informací než graf zastávek

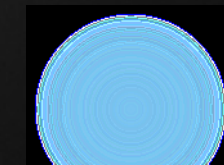
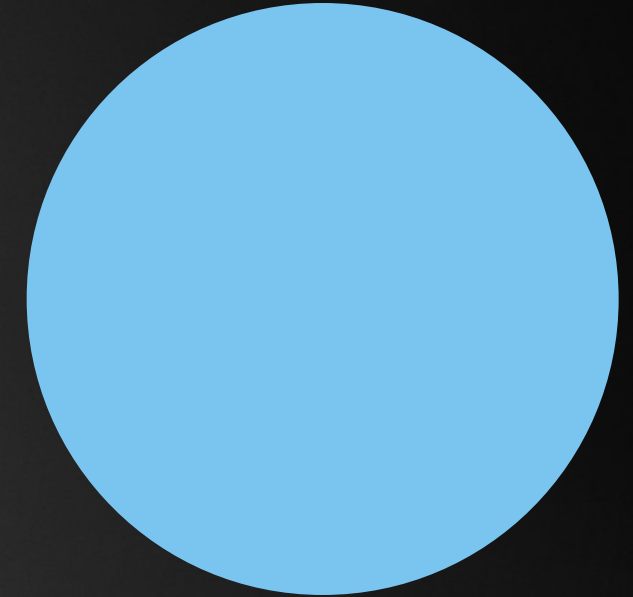
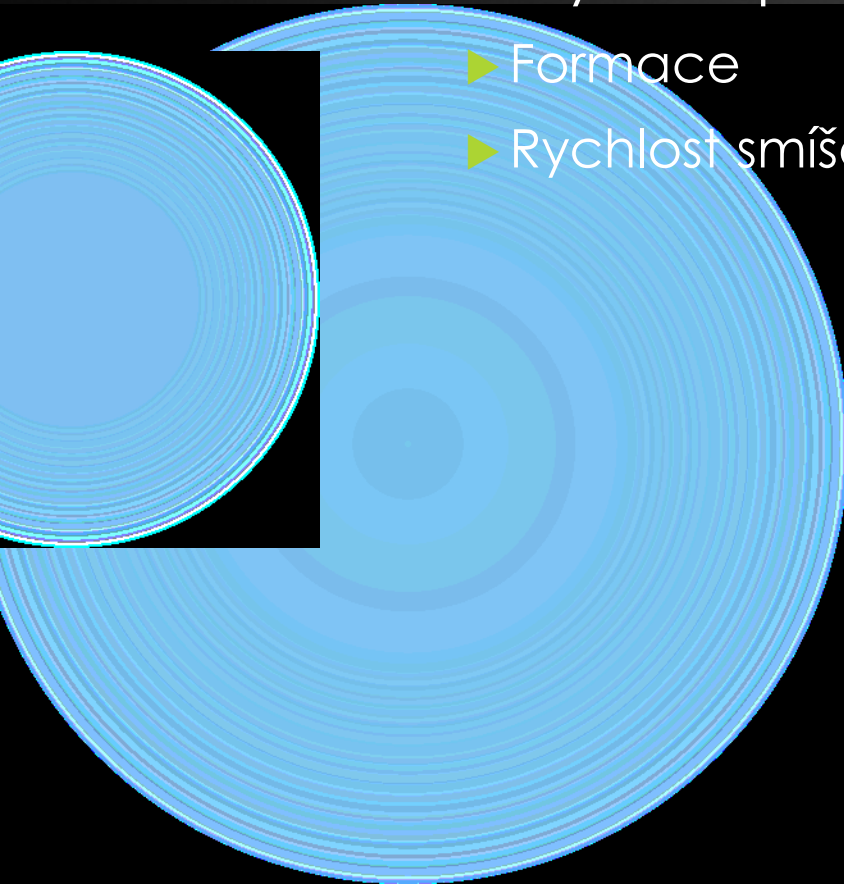
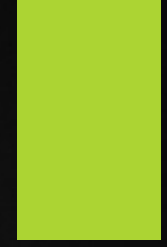
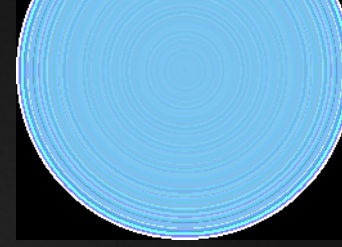
▶ Hotová implementace v:

- ▶ Cryengine
- ▶ Unity
- ▶ Unreal Engine



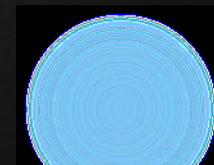
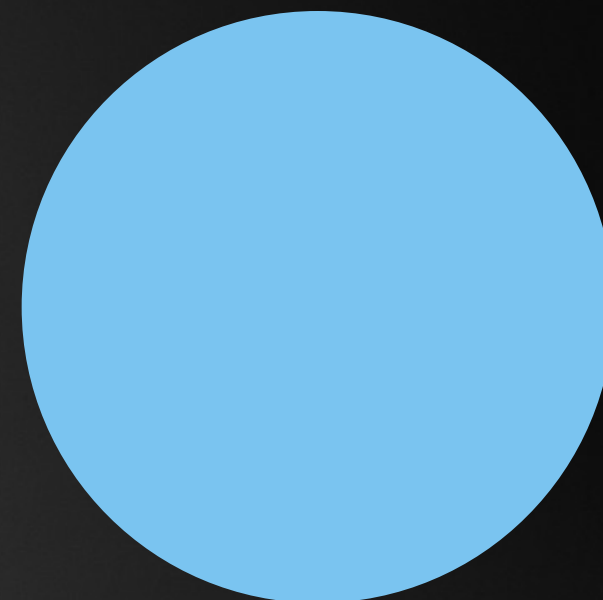
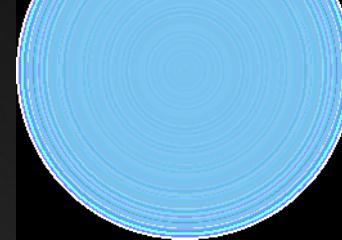
Pohyb NPC po mapě

- ▶ Fog of war
- ▶ Jak projít skrz vlastní linii jednotek
- ▶ Pohyb skupiny jednotek
 - ▶ Formace
 - ▶ Rychlost smíšené skupiny (jízda plus pěší)



Osnova

- ▶ Úvod
- ▶ Vnímání
- ▶ Rozhodování
- ▶ Navigace
- ▶ **Další**
 - ▶ Skupinové chování
 - ▶ Artificial Stupidity
 - ▶ Výzkum v oblasti AI



Skupina – smečka, stádo, hejno, ...

- ▶ Chceme aby se skupina NPC pohybovala společně
- ▶ Základní algoritmus: „Boids“ – každý člen skupiny se snaží dodržet tři pravidla:
 - ▶ Seperace
 - ▶ Zarovnání
 - ▶ Koheze
- ▶ Craig Reynolds: *Flocks, Herds and Schools – A Distributed Behavioral Model*, SIGGRAPH, 1987

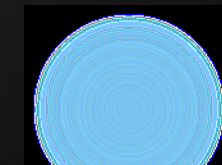
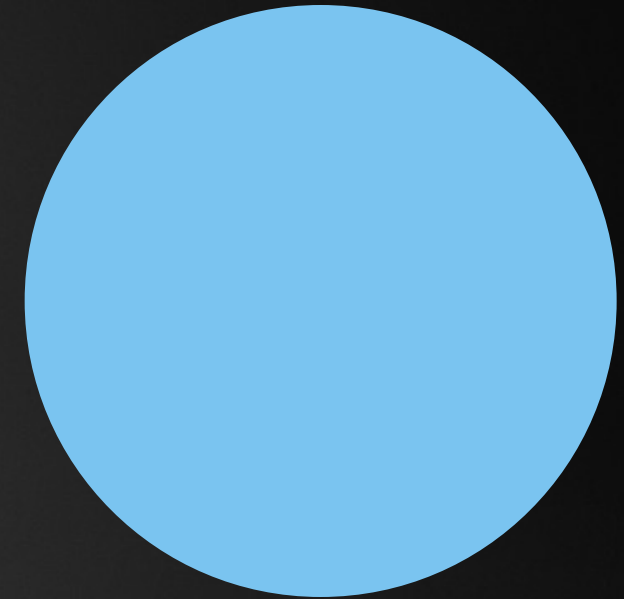
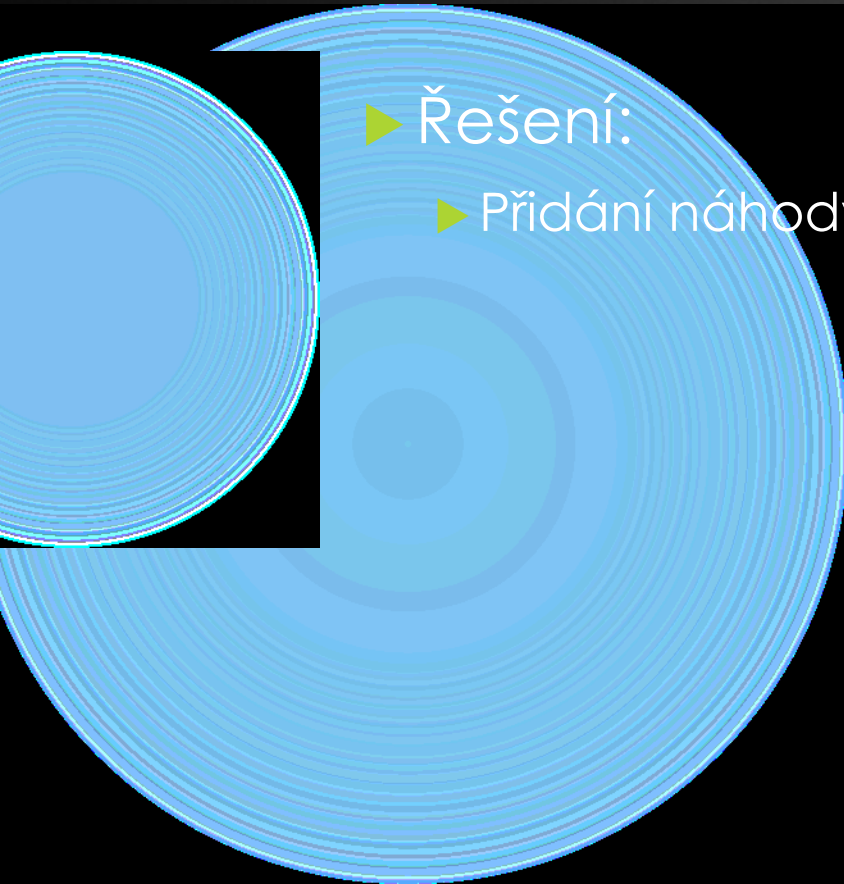
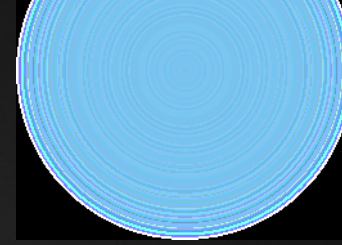
Umělá „hloupost“

▶ Nechceme:

- ▶ Příliš dobrý AI střelec, řidič,
- ▶ Příliš jednotvárný střelec, řidič

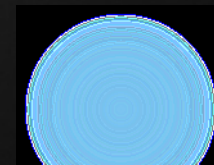
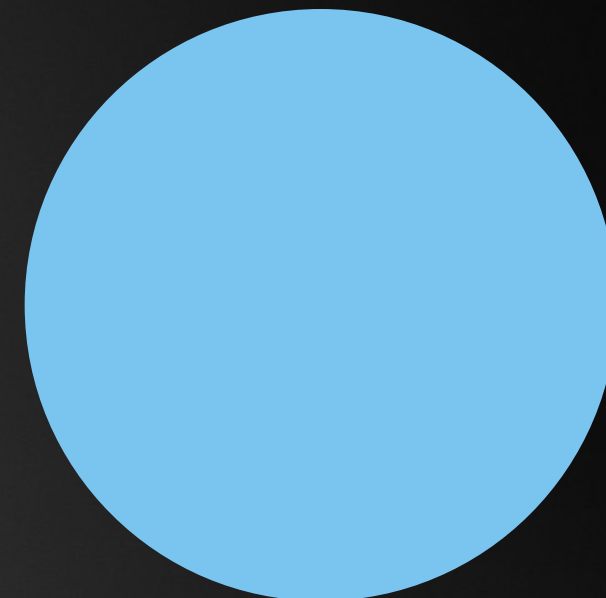
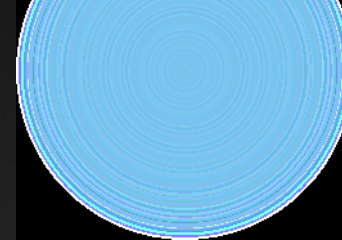
▶ Řešení:

- ▶ Přidání náhody do AI



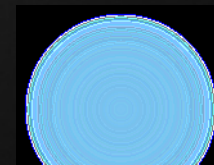
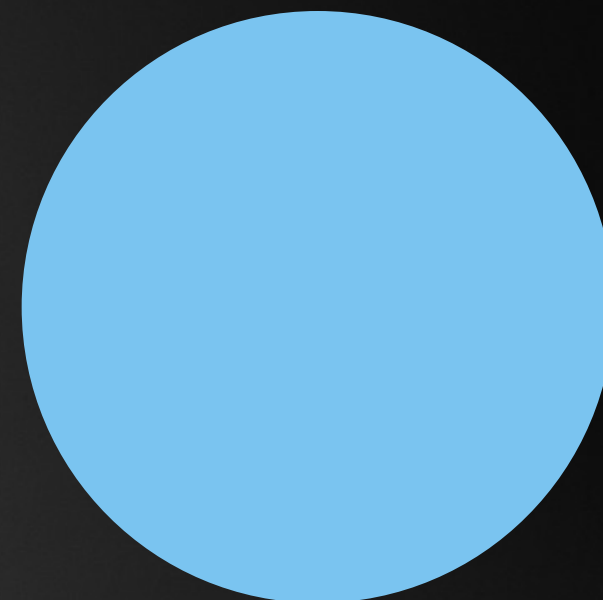
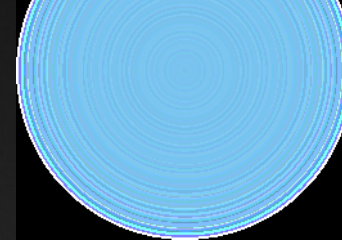
Umělá „hloupost“

- ▶ Umělé přidávání šumu na vstupu
 - ▶ Vzdálenost, rychlost hráče (ostatních aut, ...)
- ▶ Vnesení náhody na výstupu
 - ▶ Střelba:
 - ▶ Přidáme náhodnou odchylku
 - ▶ Miss-roll
 - ▶ per turn, per battle (X-Com EU)
 - ▶ Střelba:
 - ▶ Zpomalení NPC projektilů (Doom 1 a 2)
 - ▶ Záměrná střelba těsně vedle
 - ▶ Miss-roll, NPC schválně míří těsně vedle hráče



Osnova

- ▶ Úvod
- ▶ Vnímání
- ▶ Rozhodování
- ▶ Navigace
- ▶ Další
- ▶ **Hry a výzkum v oblasti AI**



Hry a výzkum umělé inteligence

- ▶ Hry tvoří výborné prostředí pro testování nových AI technik
 - ▶ Šachy
 - ▶ 1996, 1997: Deep Blue vs. G. Kasparov
 - ▶ Atari 2600 games (Pong, Breakout, Space Invaders, ...)
 - ▶ 2003: Deep Reinforcement Learning, DeepMind
 - ▶ GO
 - ▶ 2016: DeepMind (Alphabet) vs Lee Sedol
 - ▶ Starcraft
 - ▶ 2016: DeepMind + Blizzard: ohlašují Starcraft II open API

Hry a výzkum umělé inteligence

▶ Výzkum AI pro RTS:

- ▶ Otevřené API

- ▶ Soutěže

- ▶ možnost porovnat výsledky různých skupin, motivace

- ▶ Od cca 2006

- ▶ Replay – pro trénování umělých inteligencí (neuronové sítě, ...)

- ▶ Lidští oponenti – generování tréninkových dat, porovnání výkonu botů

Literatura

▶ Knihy:

- ▶ Různí: *Game Programming Wisdom I, II*
- ▶ Ian Millington, John Funge: *Artificial Intelligence for Games*, 2009
- ▶ A. S. Kyaw, T. N. Swe: *Unity 4.X Game AI Programming*, 2013

▶ Články:

- ▶ John Laird, Michael VanLent: *Human-Level AI's Killer Application: Interactive Computer Games*, AI MAGAZINE, 2001
- ▶ Marc Ponsen: *Improving Adaptive Game AI with Evolutionary Learning*, Diploma thesis, 2004