Planning for computer games

Jiří Vokřínek

A4M36PAH - 14.4.2014

Planning for computer games

- Computer game
  - Illusion of intelligence in the behaviour of computer characters
  - Player experience have to “feel” natural
  - Uses path planning, decision making, planning
  - Sensing in complex environment (cheating?)
  - ~50% of game project time is spent on building AI

Planning for computer games

- Planning as playing
  - The hard job does the player

Planning for computer games

- Planning for game AI
  - Make the game fun – reasonable challenge
  - “Real-time” application
  - Limited processor/memory resources
  - Cheating as a balance: more damage, more health, precise aiming, faster movement vs. full information, environment relaxation, etc.
  - Modern games – complex 3D environment with high level of player freedom = hard problem for AI
  - Modern computers – more power for planning/AI

Planning for computer games

Sim City (1989)

Sim City (2013)

Planning for computer games

Duke Nukem (1991)

Duke Nukem Forever (2011)

Planning for computer games

Fallout (1997)

Fallout: New Vegas (2010)

Planning for computer games

StarCraft (1998)

StarCraft II: Wings of Liberty (2010)

Game AI Challenges

- Resources
  - Games are focused on graphics
  - Advanced environment needs CPU power
  - “Animation” synchronized computing
- Environment representation
  - Game environment have to be fixed before AI design
  - Compromise of time vs. effect, hard to make guaranties
  - Environment changes strongly affects AI

Game AI Challenges

- Over Intelligence
  - No one wants to be beaten by perfect AI
  - Fun is based on believability
  - Characters must react realistically (context)
- Research and Development
  - Low cohesion between AI research and game development
  - AI in computer games seems trivial to AI researchers
  - State-of-the-art research AI is far from deployment in computer games

AI in FPS-style Games

- Bottom layer – path finding
  - Predetermined navigation graphs + A*
- Top layer – reasoning and behaviour
  - Story and experience -> scripts
- Event-based actions
- Animation system
  - Appropriate sequence for selected action
  - Inverse kinematics

AI in RTS-style Games

- Bottom layer – path finding
  - Open environment map + A*
  - Dynamic environment with collisions
- Top layer – reasoning and behaviour
  - Goal based priorities and planning
  - Terrain and game situation
  - Building infrastructures, producing resources, spending resources …
  - Combination of location sensitive many-purpose units/structures

Human level AI

- Human intelligence is not easy replicable in computer program
- Turing test problem
- Context and episodic memory
- Emotions and attitudes
- Dialog systems
- Believable reactions

Game AI Challenges

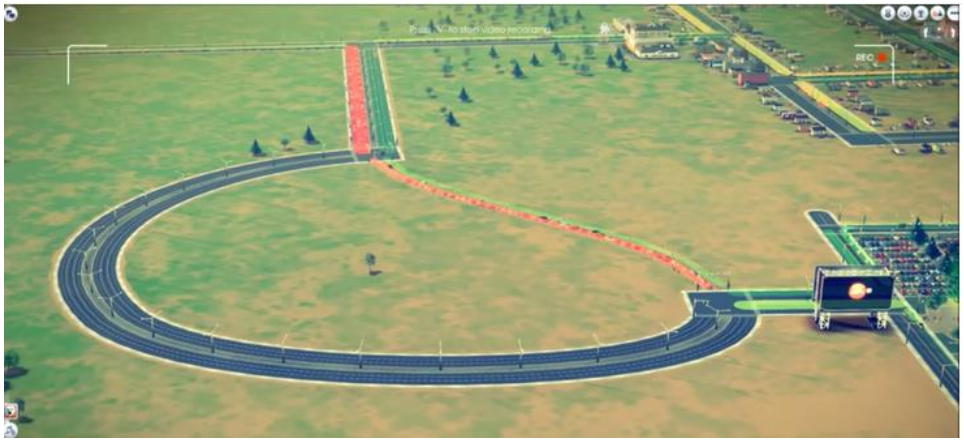
- It is extremely hard to created perfect game AI
- It is extremely easy to produce a bug in the AI
  - Examples:

18 Embarrassing Game AI Bugs Caught On Tape… and Fixed! (Alex J. Champandard on April 28, 2009)

http://aigamedev.com/open/article/bugs-caught-on-tape/

Game AI Challenges

- Shortest path example – SimCity (2013)



- Path planning is one of the most visible problems!

http://www.youtube.com/watch?v=g418BSF6XBQ
http://www.youtube.com/watch?v=ABJjdpxeMtE
https://www.youtube.com/watch?v=CXpQYngyQRI

Planning in Computer Games

- Path finding, navigation maps

  vs

- Scripted behaviour (rule based)
- State machines
- Strips based planners
- Hierarchical task networks
- Behaviour trees
- Utility systems
- Massive crowd behaviour

Path Finding and Navigation Maps

http://images.bit-tech.net/content_images/2009/03/how-ai-in-games-works/fear2seg2sm.jpg

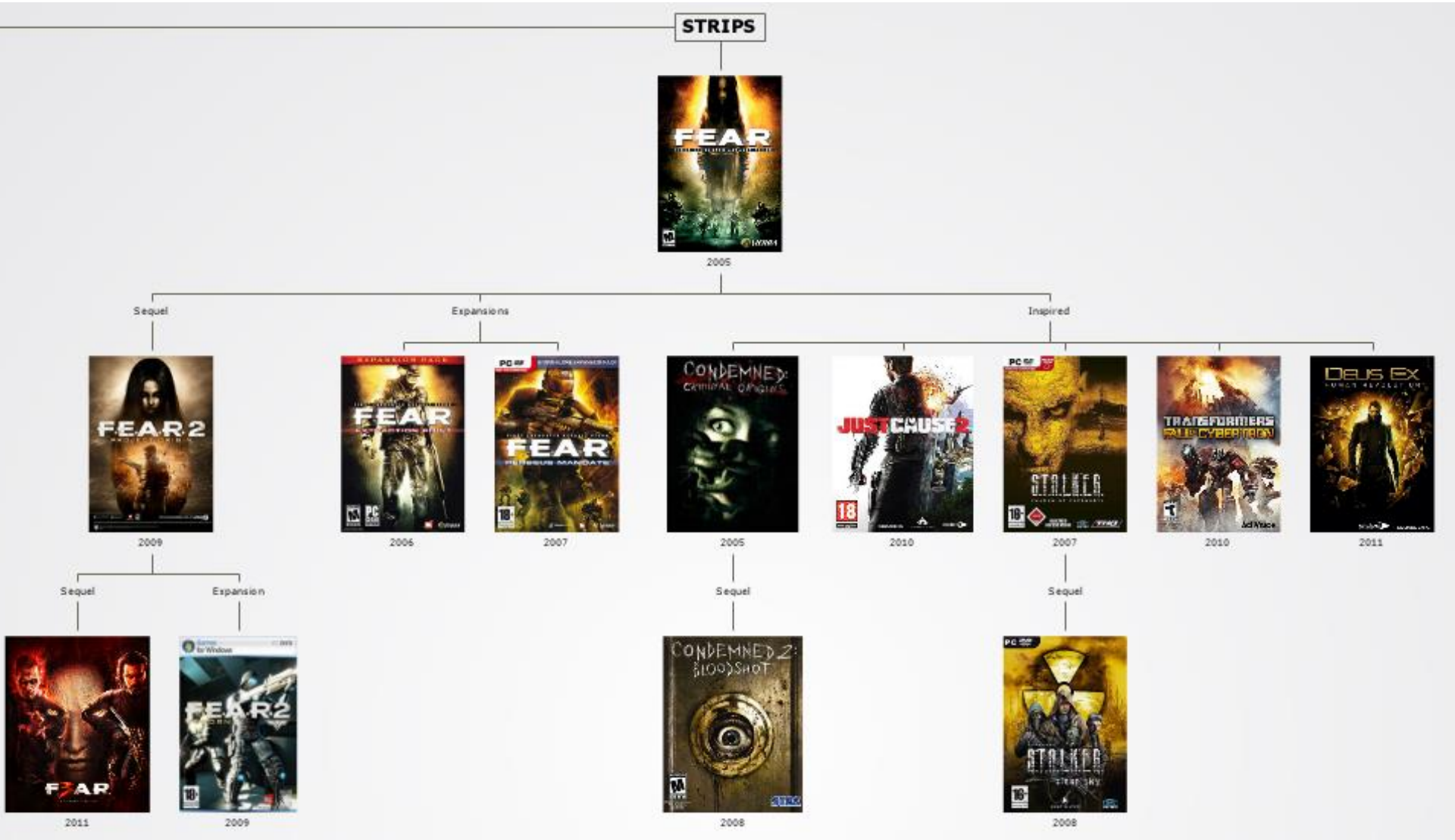
State Machines

- Every actor has a state (shooting, seeking, …)
- Triggers to make state transactions
- Each state is represented by scripted behaviour
- Decades proven AI technique
- Many states and many actors lead to extremely large state machines
- Examples are Crysis, Dawn of War 2

STRIPS Based Planners

- STRIPS searches through states (game situations), typically backward from goal
- F.E.A.R. – first STRIPS-like AI
- Also in S.T.A.L.K.E.R. series, Condemned, Just Cause 2
- Better player acceptance than scripted or linear story-driven

STRIPS Based Planners



http://aigamedev.com/open/review/planning-in-games/

STRIPS Based Planners

Left: action plan in F.E.A.R.

Down: Fallout 3 Super Mutants loot weapons from dead bodies

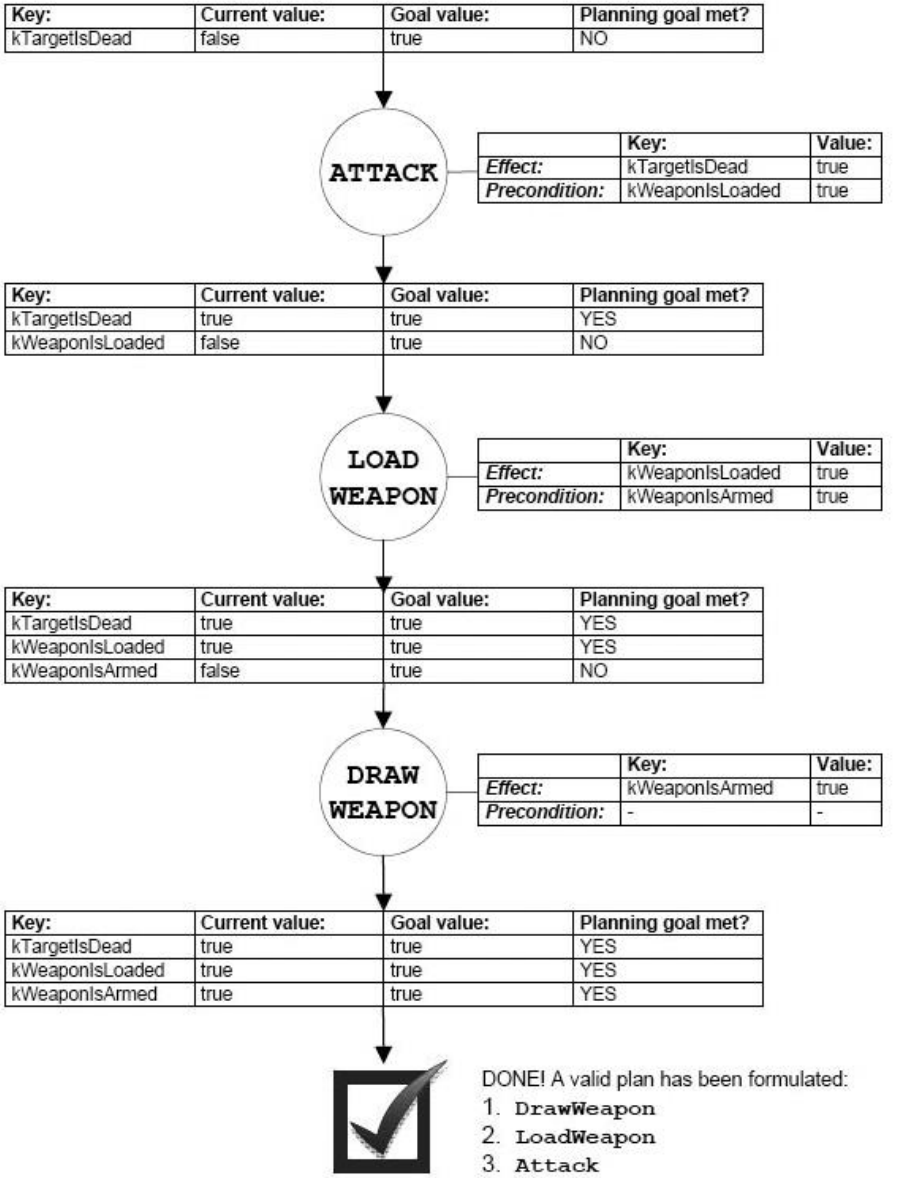


Figure 6. The planner finds a valid plan by a regressive search (after Orkin, 2004c, pp. 225).

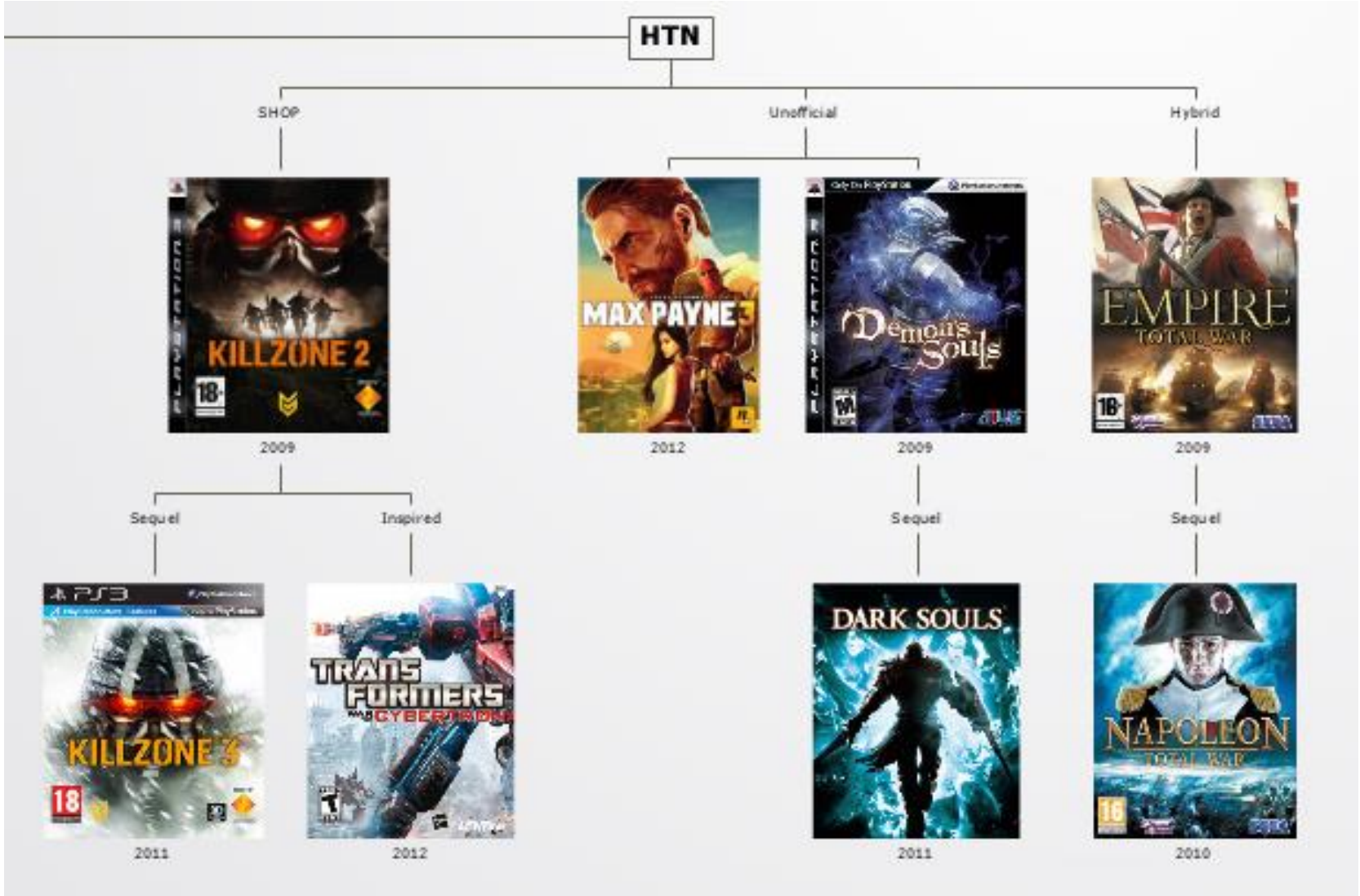
http://www.bit-tech.net/gaming/2009/03/05/how-ai-in-games-works/3

HTN Based Planners

- Recursively broken down hierarchies of tasks
- Ordered HTN is more reliable and robust
- Partial order approach is closer to STRIPS
- SHOP-like planner in KILLZONE 2
- More popular than STRIPS planers in games

- Similarity with Behaviour trees

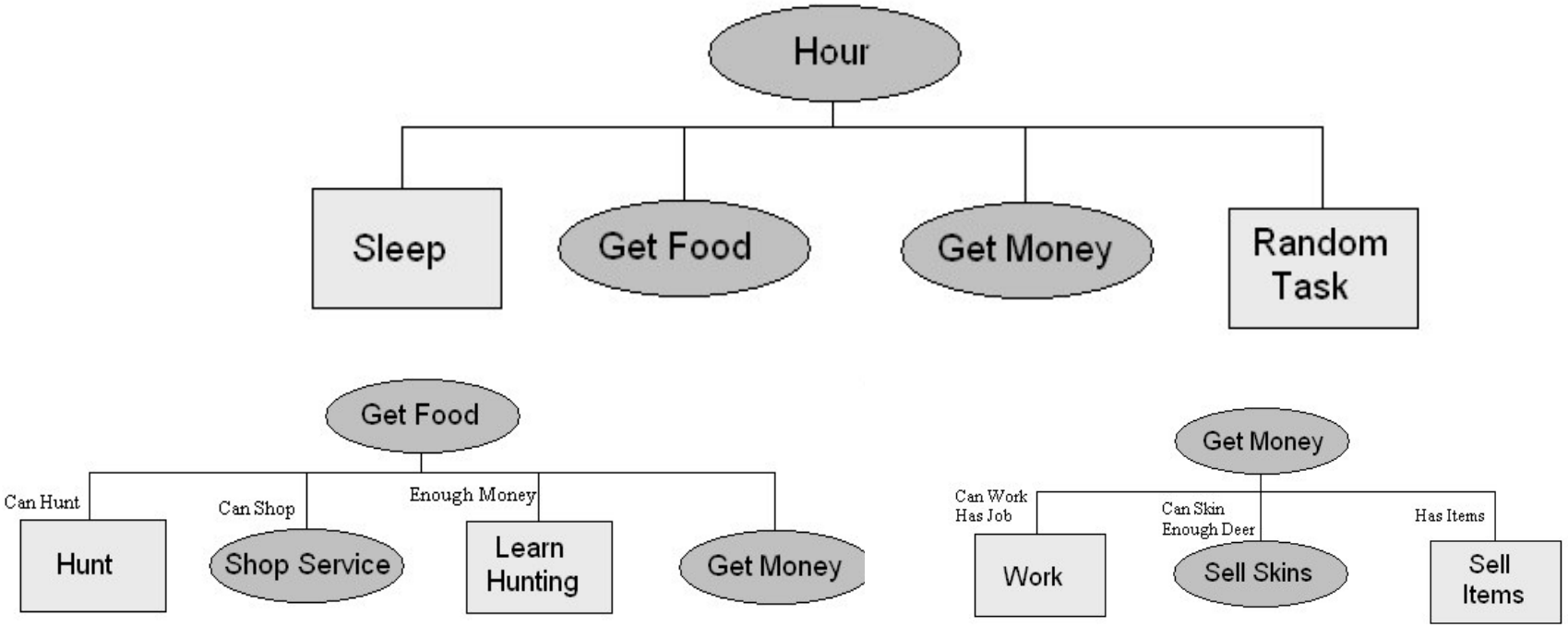
HTN Based Planners



http://aigamedev.com/open/review/planning-in-games/

HTN Based Planners

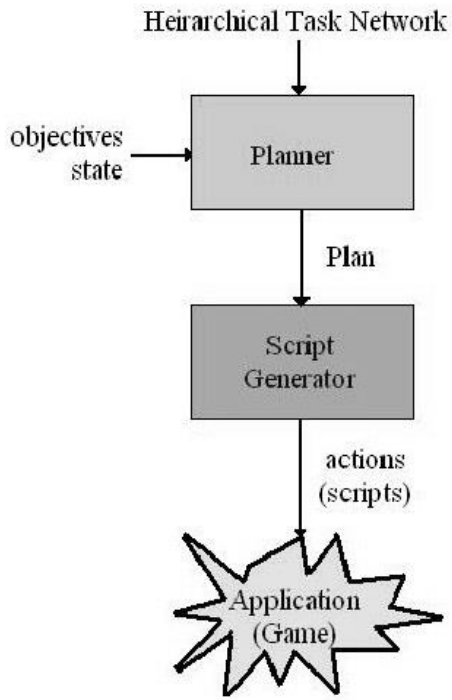
- Oblivion example



Kelly, John-Paul, Adi Botea, and Sven Koenig. "Planning with hierarchical task networks in video games." *Proceedings of the ICAPS-07 Workshop on Planning in Games*. 2007.

HTN Based Planners

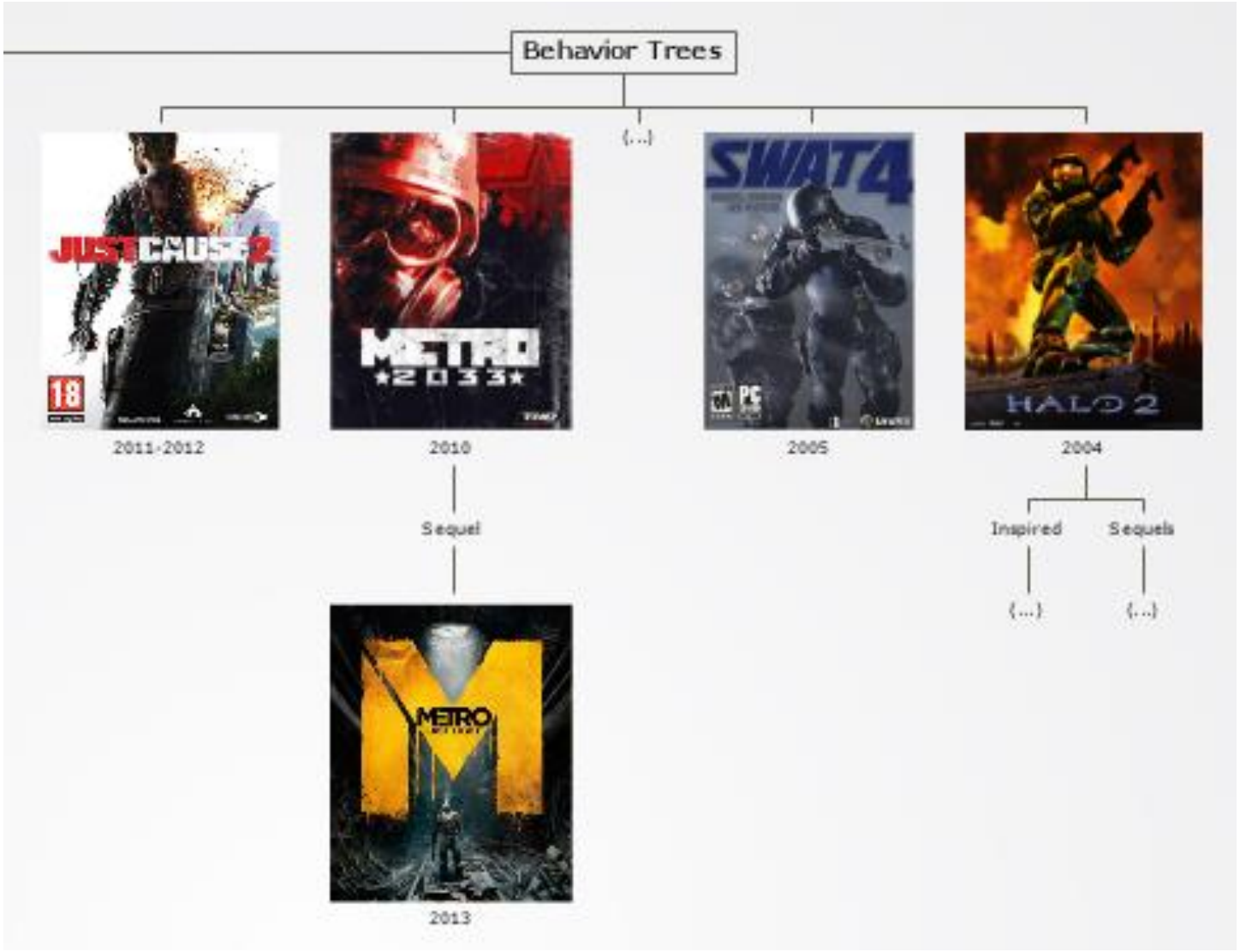
- Oblivion example




```
if (theTime >= 10 && theTime < 11&&
    finished != 10)
   if current == 0
      AddScriptPackage TravelDeerForest
      set current to current + 1
   endif
   if current == 1
      if bobVar != 60
         set bobVar to 60
         set skinning to 1
         set skinDeer to 0
         AddScriptPackage skin
      endif
   endif
   if current == 2
      AddScriptPackage TravelSkinShop
      set current to current + 1
   endif
   if current == 3
      if bobVar != 70
         set bobVar to 70
         set selling-skin to 1
         AddScriptPackage SellSkins
      endif
   endif
   if current == 4
      set current to 0
      set finished to 10
      set bobvar to 0
   endif
endif
```

Kelly, John-Paul, Adi Botea, and Sven Koenig. "Planning with hierarchical task networks in video games." *Proceedings of the ICAPS-07 Workshop on Planning in Games*. 2007.

Behaviour Trees



http://aigamedev.com/open/review/planning-in-games/

Utility Systems

- Voting/scoring system for goal/task selection
- Emergent behaviour (like STRIPS) rather than top-down design
- More “action-search” than a planner
- Examples are DEMIGOD or SIMS 3
- Good level of realism on character level
- Can be combined with higher level strategic/tactical planning

Utility Systems

http://aigamedev.com/open/review/planning-in-games/

Massive Crowd Behaviour

- Massive entities behaviour (i.e. battle system)
- From abstract units to individuals
- Movie industry inspired

Colonization (1994)
Empire Total War (2009)

Trends and Future

- Reactive techniques beat in time and quality in some cases, but …
- Planning rocks in open environments (hey! it is domain independent!)
- Goal-based game design is inevitable
- Modern computers have enough power for planning in computer games
- Shift from tweaks to data/knowledge oriented game design (animation vs. planning based game engines)

Trends and Future

- Intelligent landscape
  - Interaction with the environment
  - Realistic sensing and actions
  - Planning instead of rules scripting
- World fillers
  - Intelligent NPCs
  - Persistent living world and storytelling
  - Intelligent opposition – strategic/tactical planning instead of scripted challenges

Trends and Future

- Game design as a planning problem?