Applications of Artificial Intelligence

Sebastian Iwanowski
FH Wedel

Chapter 7:
Game AI

Game AI

Different types of games:

1. **Turn-based strategy games**
2. **Real-time strategy games (RTS)**
3. **Multi-user strategy games**
4. **Sport simulations**
5. **Evolving simulations**

Nähere Infos: Seminarvortrag und Ausarbeitung von Julian Huppertz, SS 2007, Nr. 1
http://intern.fh-wedel.de/mitarbeiter/iw/lv/ss-2007/seminarspieleki.html

Game AI

Definition of Game AI:

Program code for decision making of elements controlled by the computer

Types of elements controlled by the computer:

Bot: **Player controlled by the computer who acts like a human player (as a combat or an opponent)**

NPC: **Element of a game interacting with a player, but not playing itself according to the same rules as a human player**

Examples: Narrator, instructor, intelligent weapon, dynamic environment

Quelle und nähere Infos: Seminarvortrag und Ausarbeitung von Daniel Dekkers, SS 2015, Nr. 1
http://intern.fh-wedel.de/mitarbeiter/iw/lv/ss-2015/seminarspieleki.html

Game AI

Expectations to Game AI:

- Enabling fairness
- Adequate (natural!) response to actions in the game.
- Consequence for behaviour of players (good/bad value system)
- Reasonable planning (path finding, decision making, etc.)
- Dynamic and not foreseen generation of game environments

Quelle und nähere Infos: Seminarvortrag und Ausarbeitung von Daniel Dekkers, SS 2015, Nr. 1
http://intern.fh-wedel.de/mitarbeiter/iw/lv/ss-2015/seminarspieleki.html

1. Turn-based strategy games

Problem:

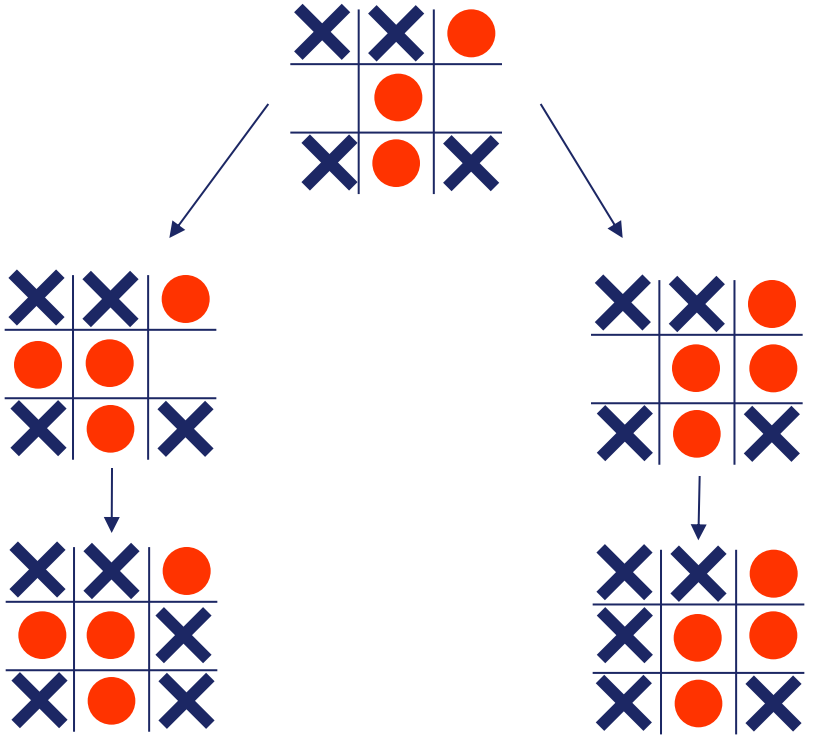
Find a strategy how to get from a starting state to a final winning state

Example: Tic-Tac-Toe

some interim state

subsequent state ●

subsequent state ✕



Quelle und nähere Infos: Seminarvortrag und Ausarbeitung von Nils Böckmann, SS 2005, Nr. 13
http://intern.fh-wedel.de/Archiv/iw/Lehrveranstaltungen/SS2005/SeminarKI.html

1. Turn-based strategy games

Chess computers

Milestone 1997:

Kasparov 2.5 – Deep Blue 3.5

further infos: http://www.research.ibm.com/deepblue

1. Turn-based strategy games

Common techniques for computer strategies:

- A finite but intractable search space is explored by a heuristic strategy.
- Interim states are evaluated by heuristic rules.
- Some indeterminism is employed.

Current state-of-the-art (since 2010):

- Chess is too simple: Modern computers beat any chess master.
- Human players use chess computers for training.
  For this sake, the intelligence of the computer may be adjusted.
- Other chess-like games are invented to make it harder for computers, e.g. Arimaa.

1. Turn-based strategy games

Latest developments (2017):

- Neural networks are used for playing Go which is much more complex than chess but very prone to pattern recognition.
- DeepMind‘s AlphaGo beat the leading Go master‘s of the world.
  It was trained by a lot of samples of previous Go matches.
- Milestone AlphaGo Zero:
  This software set up the training cases by itself (using rules and probabilistic decisions) and learnt from the experience of these cases.

see seminar presentation on this (WS 2019/20)

New features of neural networks:

- modified weight learning algorithms (since 2014)
- specialised neural network architectures
- better hardware enables the use of millions of training samples

Reference: Ian Goodfellow, Yoshua Bengio, Aaron Courville: Deep Learning, MIT Press 2016

2. Real-time strategy games

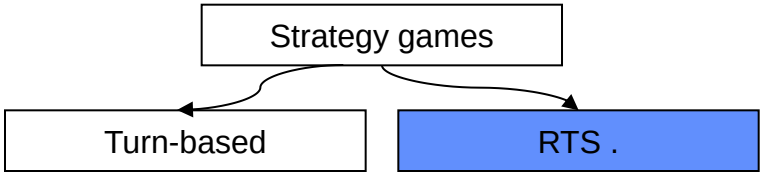
Examples (2007):

More recent examples will be given in subsequent slides

Half Life

Command + Conquer

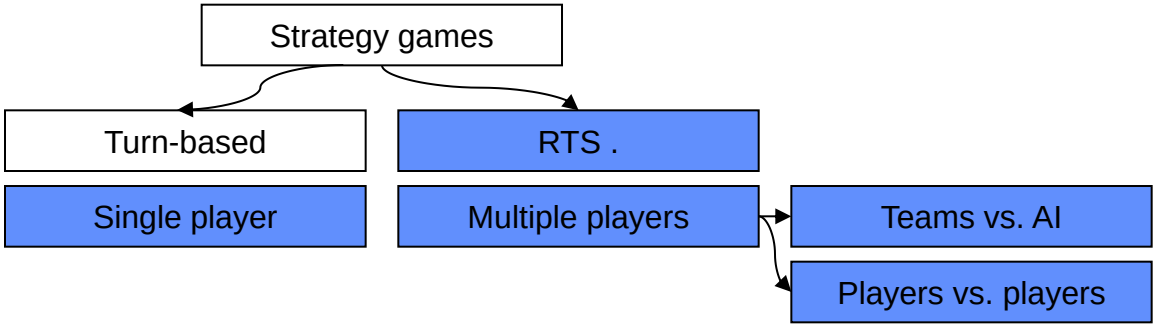
2. Real-time strategy games



- Players make their actions concurrently.
- This requires a fast reaction time.
- Complexity may be reduced.
- „Real time“ is not to be understood literally: 1 day in game ≠ 1 day in reality
- Examples: Starcraft, Warcraft, Command and Conquer, Age of Empires

Quelle: Seminarvortrag und Ausarbeitung von Jan Lipski, SS 2007, Nr. 3

2. Real-time strategy games



Single player

- always vs. AI
- scenarios, story books

Multiple player

- Teams of players vs, AI
- PvP: players vs. player (teams vs. teams)
  - needed in LAN games or internet games

Quelle: Seminarvortrag und Ausarbeitung von Jan Lipski, SS 2007, Nr. 3

2. Real-time strategy games

Typical AI tasks:

a) Path finding and terrain analysis

b) Resource planning

c) Combat tactics and strategies

Quelle: Seminarvortrag und Ausarbeitung von Julian Huppertz, SS 2007, Nr. 1

2. Real-time strategy games

a) Path finding and terrain analysis

Limited scope (mainly rectangular)

Different levels of elevation

Different view to the map (mostly irrelevant for bots)

Different types of terrain

- passable
- impassable
- passable under certain conditions

Quelle: Seminarvortrag und Ausarbeitung von Jan Lipski, SS 2007, Nr. 3

2. Real-time strategy games

a) Path finding and terrain analysis

Example: Warcraft

different levels of elevation

impassable

minimap

units commands

Quelle: Seminarvortrag und Ausarbeitung von Jan Lipski, SS 2007, Nr. 3

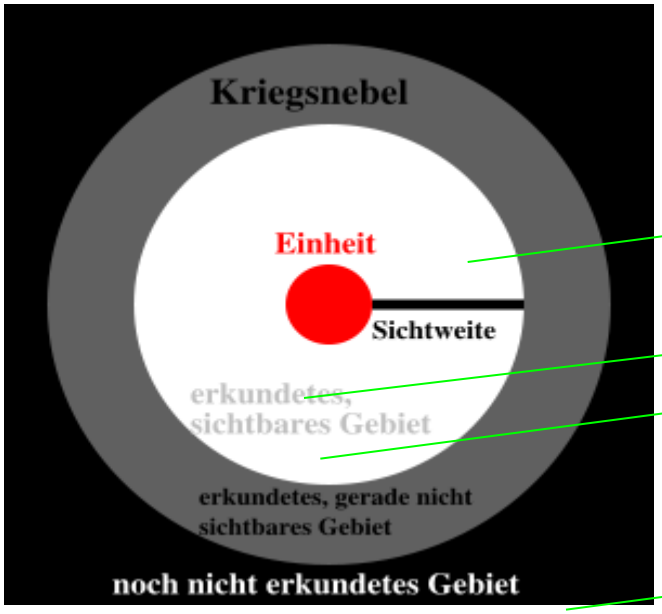
2. Real-time strategy games

a) Path finding and terrain analysis

Example: Warcraft

Visibility zones for units

- „Fog of war“ (first implementation: *Warcraft 2*)
- shows visibility ranges of units



Quelle: Seminarvortrag und Ausarbeitung von Jan Lipski, SS 2007, Nr. 3

2. Real-time strategy games

a) Path finding and terrain analysis

(Example: Command + Conquer)

- How does the whole army get from home base to enemy base ?
- How do all troops traverse bridges without impeding each other ?
- Are there any obstacles on the route planned ?
- Is there terrain which should better not be traversed?
  (Example in AI critique of 2008: Collector who picks tiberium in the shooting range of enemy)
- Where should I erect what kind of buildings? – e.g. for being safe against ion storms

Quelle: Seminarvortrag und Ausarbeitung von Julian Huppertz, SS 2007, Nr. 1

2. Real-time strategy games

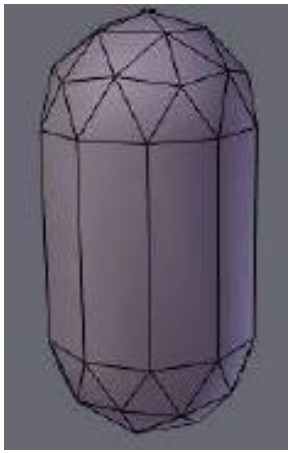
a) Path finding and terrain analysis in detail

Collision protection for 3d solids

Problem is simplified in games.
In most cases, no rotation is possible.
Exact results are not important as long as graphics makes no errors evident.

Axis aigned
bounding box
in Half-Life 2

Capsule



Quelle und nähere Infos: Seminarvortrag und Ausarbeitung von Willy Schinmeyer, SS 2015, Nr. 2
http://intern.fh-wedel.de/mitarbeiter/iw/lv/ss-2015/seminarspieleki.html

2. Real-time strategy games

a) Path finding and terrain analysis im Detail

Regular meshes

Graph is defined implicitely by discrete game environment

Hexagon areas
in King's Bounty

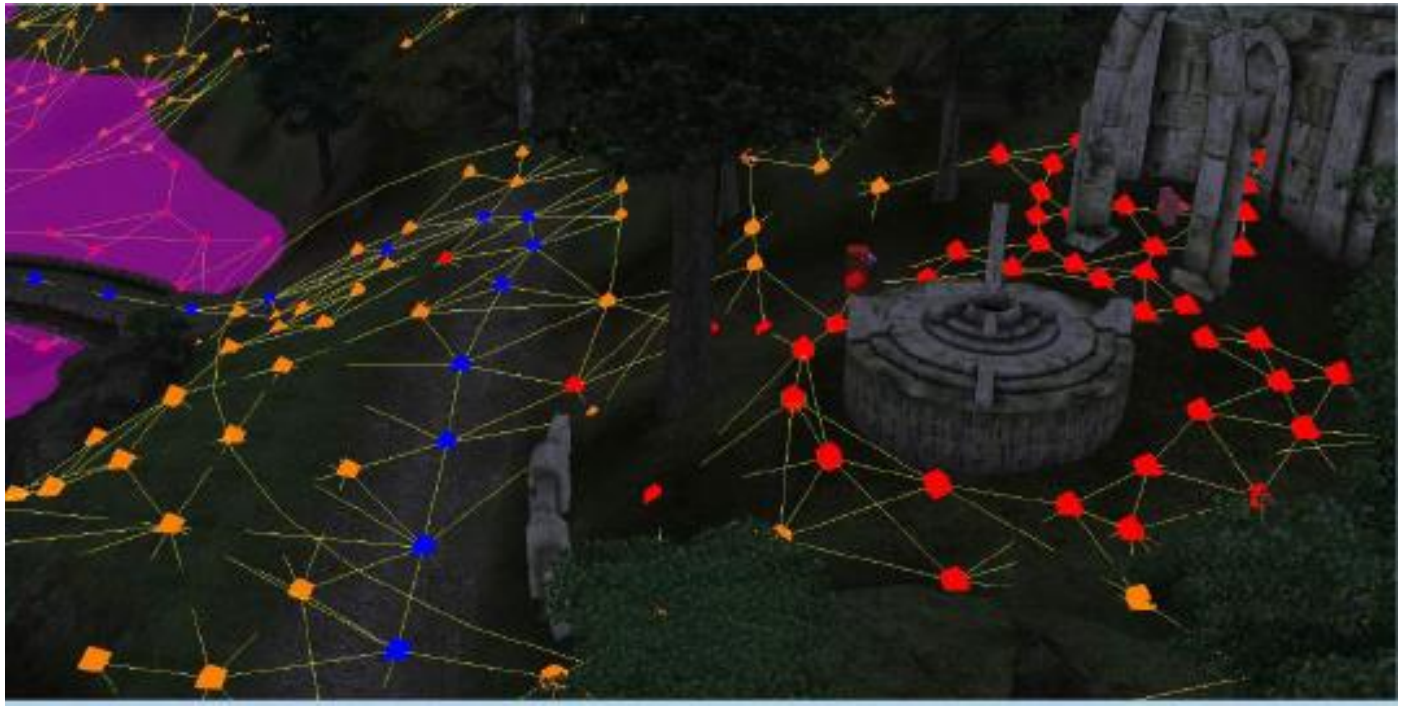
Quelle und nähere Infos: Seminarvortrag und Ausarbeitung von Willy Schinmeyer, SS 2015, Nr. 2
http://intern.fh-wedel.de/mitarbeiter/iw/lv/ss-2015/seminarspieleki.html

2. Real-time strategy games

a) Path finding and terrain analysis im Detail

Way points

Different weights are possible.



The Elder Scrolls IV: Oblivion

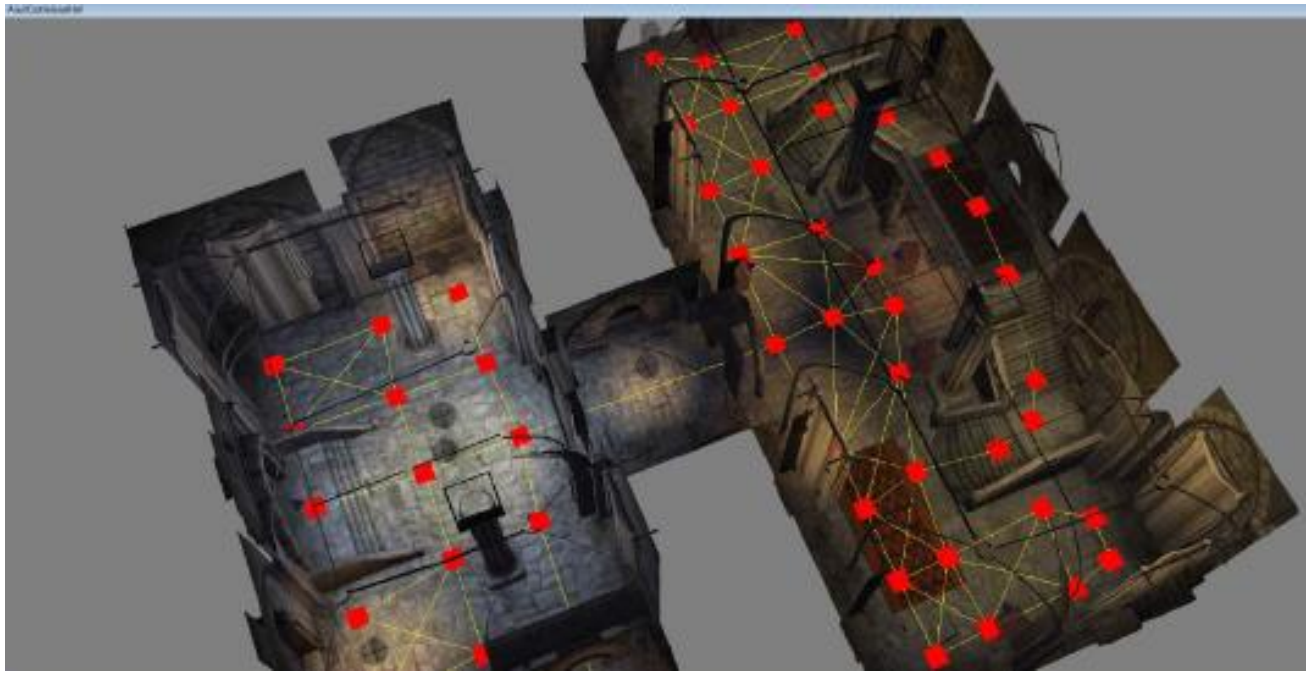
Quelle und nähere Infos: Seminarvortrag und Ausarbeitung von Willy Schinmeyer, SS 2015, Nr. 2
http://intern.fh-wedel.de/mitarbeiter/iw/lv/ss-2015/seminarspieleki.html

2. Real-time strategy games

a) Path finding and terrain analysis im Detail

Way points

... may also be used for a precise modeling of floors.



The Elder Scrolls IV: Oblivion

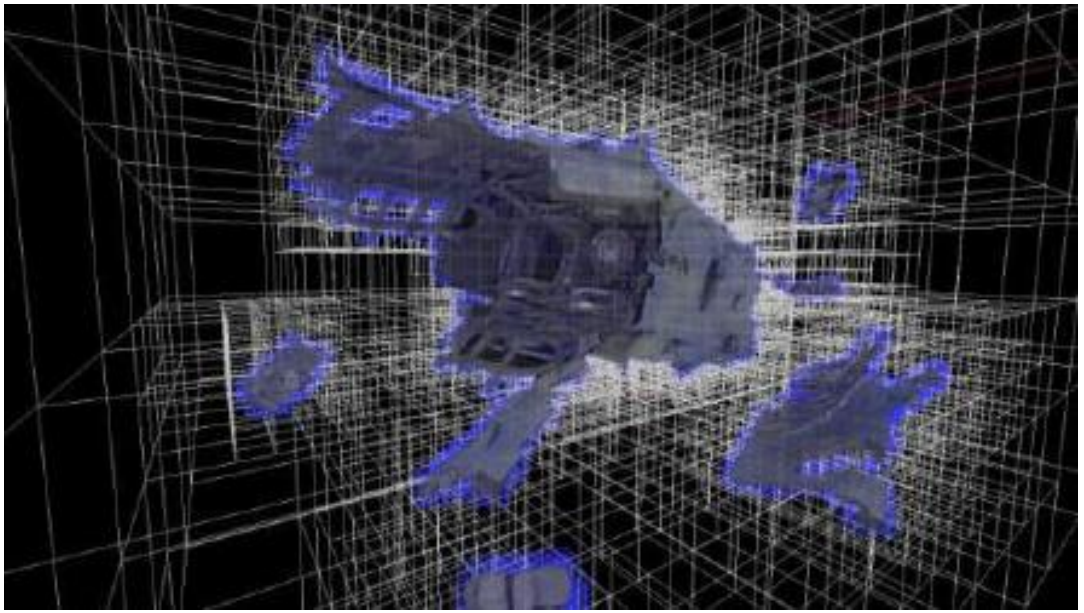
Quelle und nähere Infos: Seminarvortrag und Ausarbeitung von Willy Schinmeyer, SS 2015, Nr. 2
http://intern.fh-wedel.de/mitarbeiter/iw/lv/ss-2015/seminarspieleki.html

2. Real-time strategy games

a) Path finding and terrain analysis im Detail

Flying actors in 3d environment

NavMesh describes areas for walking.
Solid obstacles are modeled by waypoints, meshes and octrees.



Octree in Warframe

[http://www.gdcvault.com/play/1022017/Getting-off-the-NavMesh-Navigating]

Quelle und nähere Infos: Seminarvortrag und Ausarbeitung von Willy Schinmeyer, SS 2015, Nr. 2
http://intern.fh-wedel.de/mitarbeiter/iw/lv/ss-2015/seminarspieleki.html

2. Real-time strategy games

b) Resource planning

(Examples: Command + Conquer, Warcraft)

- Strongly simplified simulation of resource finding and production
- Limited number of different resource types (< 5)
- Resources are fundamental for building houses, units, developments
- Limited and sparse resources on the map lead to a wanted battle about recources.
- Non-militaristic applications: Strategy games with wanted business focus (*Settlers, Anno series, Rising cities, Roller coaster*)

Quelle: Seminarvortrag und Ausarbeitung von Jan Lipski, SS 2007, Nr. 3

2. Real-time strategy games

b) Resource planning

(Example: Command + Conquer)

- I have got 27.500$: How do I spend this effectively in a long run?
- In which technology branch should I invest?
  (e.g. air or ground forces)

Quelle: Seminarvortrag und Ausarbeitung von Julian Huppertz, SS 2007, Nr. 1

2. Real-time strategy games

c) Combat tactics and strategies

(Example: Command + Conquer)

- How do I organise my base in order to defend myself effectively?
- Which formation is most effective for attack?
- Analysis of enemy behaviour
- Should I form many small units or one big unit?

Quelle: Seminarvortrag und Ausarbeitung von Julian Huppertz, SS 2007, Nr. 1

2. Real-time strategy games

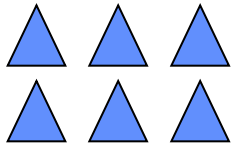
c) Combat tactics and strategies

Typical feature of RTS games:

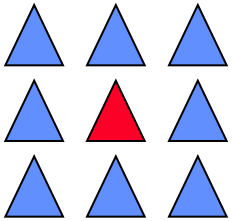
- Groups of units

Examples for formations:

- Line



- Box



- Wedge / V

Quelle: Seminarvortrag und Ausarbeitung von Jan Lipski, SS 2007, Nr. 3

2. Real-time strategy games

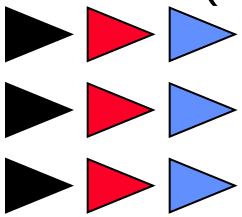
c) Combat tactics and strategies

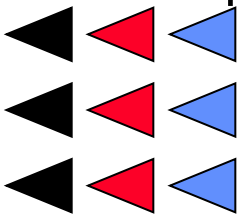
Important features of formations:

- Order of different unit types, effectivity
- Orientation of units, depth of formation
- Mutual impediment, collision?
- Scope of attack, speed of movement

  Shooter:  Equestrian:  foot soldier: 

- Who wins? (often rock-paper-scissors principle)



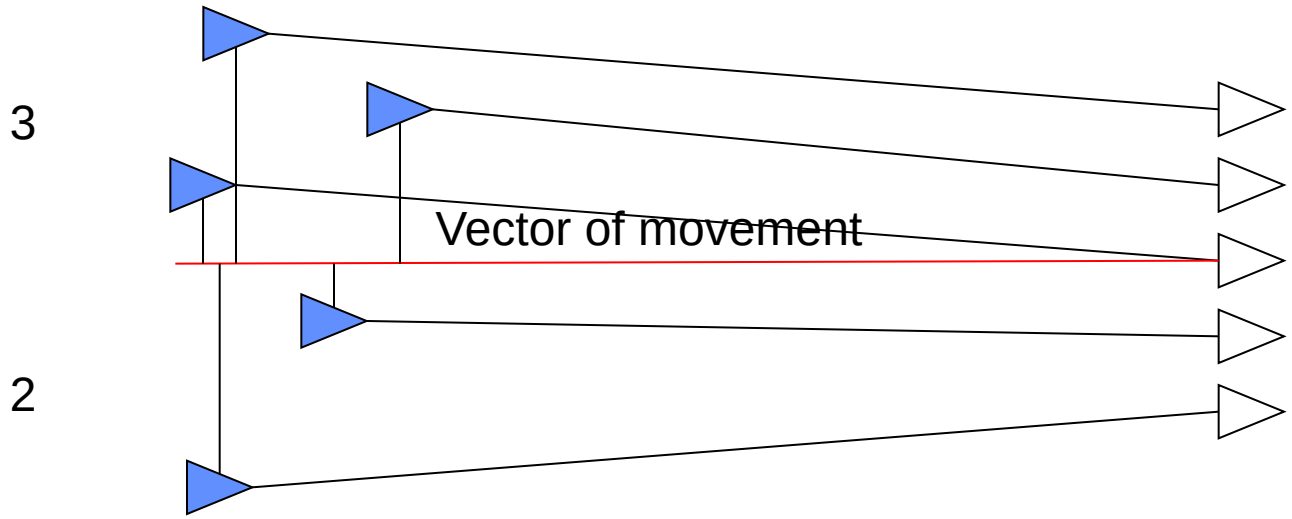


Quelle: Seminarvortrag und Ausarbeitung von Jan Lipski, SS 2007, Nr. 3

2. Real-time strategy games

c) Combat tactics and strategies

- Combining units into a formation
- Simple example: All units of the same type
- Terrain without obstacles
- Desired formation: 1 line

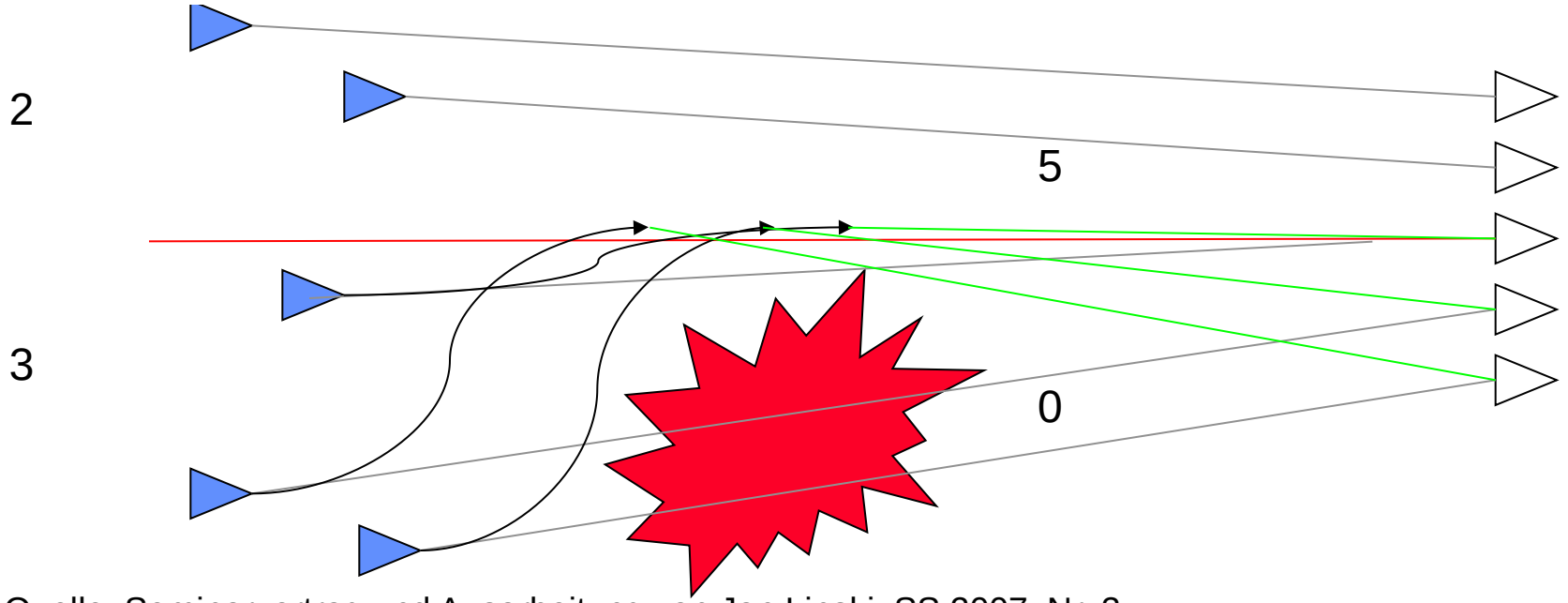


Quelle: Seminarvortrag und Ausarbeitung von Jan Lipski, SS 2007, Nr. 3

2. Real-time strategy games

c) Combat tactics and strategies

- Combining units into a formation
- Simple example: All units of the same type
- Terrain with obstacles
- Desired formation: 1 line



Quelle: Seminarvortrag und Ausarbeitung von Jan Lipski, SS 2007, Nr. 3

3. Multi-user strategy games

Example: World of Warcraft

Role game with different characters being played in the WWW

3. Multi-user strategy games

Example: World of Warcraft

AI features

- offers basic algorithms, e.g. for path finding
- Waypoints für patroling against enemy forces
- Elements of social behaviour
- No real AI features as in a modern RTS

Quelle: Seminarvortrag und Ausarbeitung von Julian Huppertz, SS 2007, Nr. 1

3. Multi-user strategy games

Example: World of Warcraft

SW technique

- World events are scripted.
- Possible extension of WoW:
  - Autonomous behaviour of artificial enemies (planning of attacks, etc.) → agent technology

Quelle: Seminarvortrag und Ausarbeitung von Julian Huppertz, SS 2007, Nr. 1

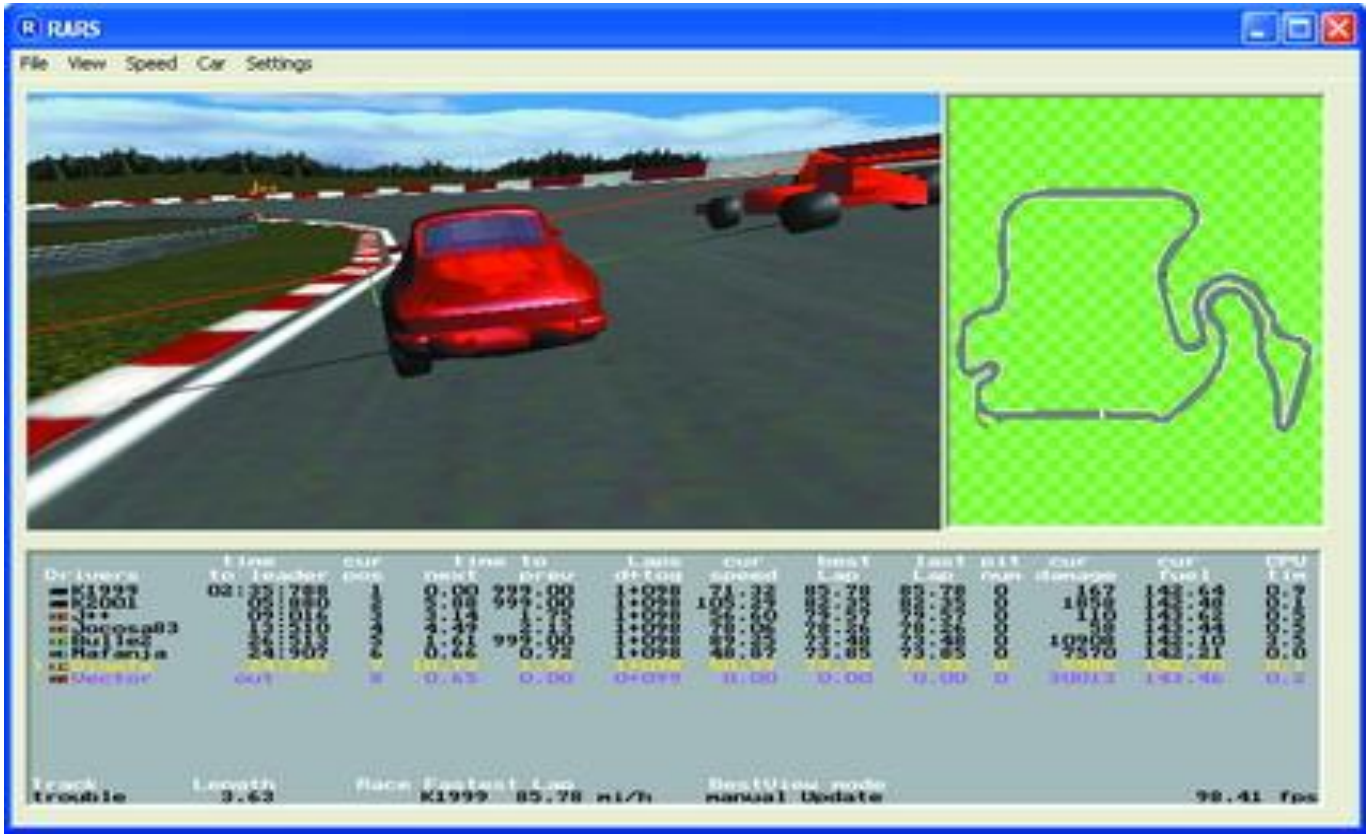
4. Sport simulations

Car racing

Quelle und Details: Seminarvortrag und Ausarbeitung von Yannick Block, SS 2007, Nr. 8

4. Sport simulations

Car racing

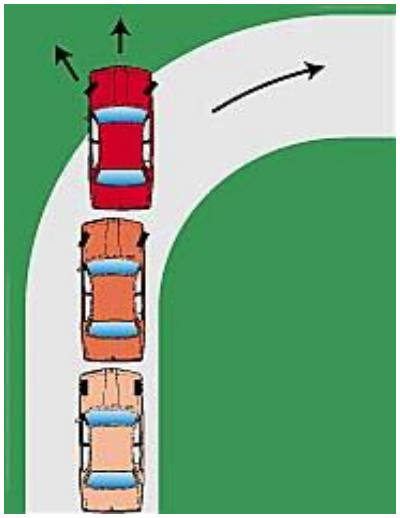


Quelle und Details: Seminarvortrag und Ausarbeitung von Yannick Block, SS 2007, Nr. 8

4. Sport simulations

Car racing

Intelligent vehicle control

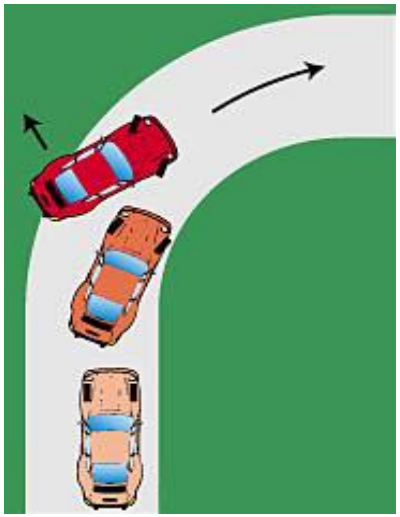


Quelle und Details: Seminarvortrag und Ausarbeitung von Yannick Block, SS 2007, Nr. 8

4. Sport simulations

Car racing

Intelligent vehicle control



Quelle und Details: Seminarvortrag und Ausarbeitung von Yannick Block, SS 2007, Nr. 8

4. Sport simulations

FIFA PC

A human player controls exactly 1 virtual player.
The other 21 players are controlled by the computer.

- Path finding algorithms
- Specifications for tactics
- Implementation of standard moves:
  - Passes
  - Corners
  - Free kicks / penalties

Quelle: Seminarvortrag und Ausarbeitung von Julian Huppertz, SS 2007, Nr. 1

4. Sport simulations

FIFA PC

Physical requirements:

- Other players must not collide.
- Game movements must result in goal shots.
- Goal shots must not be too perfect.
- All situations according to the rule (corners, free kicks, etc.) must be implemented.

Quelle: Seminarvortrag und Ausarbeitung von Julian Huppertz, SS 2007, Nr. 1

4. Sport simulations

FIFA PC

Implementation with state automata for the players:

- Each state stands for the current task of a player.
- States are like „I have got the ball“, „I am next to the ball“, „I am injured“

Quelle: Seminarvortrag und Ausarbeitung von Julian Huppertz, SS 2007, Nr. 1

5. Evolving simulations

Example: Sims

- Player may modify the priority list of a Sim character.
- When the player is passive:

  Actions evolve automatically according to certain scripts

  (various needs are satisfied automatically)

Quelle: Seminarvortrag und Ausarbeitung von Julian Huppertz, SS 2007, Nr. 1

5. Evolving simulations

Example: Sims

Smart terrain / Message passing:

Example: Sim character is thirsty

- Kitchen board signalises: I have got a drink.
- Sim character drinks an needs a toilet afterwards.
- Toilet signalises: Here I am.

Quelle: Seminarvortrag und Ausarbeitung von Julian Huppertz, SS 2007, Nr. 1

5. Evolving simulations

Example: Sims

- uses genetic / evolutionary algorithms
- passing on phenotypic and behavioural features to descendents

Quelle: Seminarvortrag und Ausarbeitung von Julian Huppertz, SS 2007, Nr. 1

Game AI in Practice

Critique 2008

- Real-time strategy games involve much more complex rules than chess. But the AI inside is very primitive.
- AI does not apply real intelligence but rather cheats to be equal to human players.
- Most rules are simple static scripts.
- The AI should not only be logical: We need emotions!
- AI does not receive the same resources as graphics: Menpower, hardware, development time

Quelle:
http://www.tecchannel.de/webtechnik/entwicklung/1744817/warum_kuenstliche_intelligenz_ki_in_spielen_stagniert/index.html

Game AI in Practice

Critique 2008

- Subsequent changes of AI in some code segment may have huge effects to the global program.
- AI elements are directly integrated in the programming code: There is seldom an engine with interfaces that can be applied by game designers.
- Many experts for AI technology refuse to work on games.
- The best AI is in the most old fashioned games.

Quelle:
http://www.tecchannel.de/webtechnik/entwicklung/1744817/warum_kuenstliche_intelligenz_ki_in_spielen_stagniert/index.html

Game AI in Practice

Dynamic Scripting

- developed at University of Utrecht.
- applied at the Fantasy-CRPG Neverwinter Nights

Basic Principles:

- At development a set of rules is defined.
- At execution only some rules are active with different weights
- Weights of rules change dynamically according to outer circumstances
- Change has got a nondeterministic component.

Quelle und nähere Infos: Seminarvortrag und Ausarbeitung von Jan Riemer, SS 2015, Nr. 3
http://intern.fh-wedel.de/mitarbeiter/iw/lv/ss-2015/seminarspieleki.html

Game AI in Practice

Further remarks on path finding in games

- Path finding must be involved in any strategy game, and nowadays this is frequently done by A* or Dijkstra.
  There are a lot of publications on the application of these algorithms referring to special adaptations to real time behaviour, software technology and storage allocation.
- For efficient path finding, not only the algorithm itself but also the underlying navigation mesh matters. There are intelligent algorithms for generating such meshes.

Further examples on this topic may be found in seminar talks at FH Wedel, e.g. from SS 2007, SS 2015 or WS 2019 at iw website.