

DM842

Computer Game Programming: AI

Lecture 9

Tactical Analysis Board Games

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

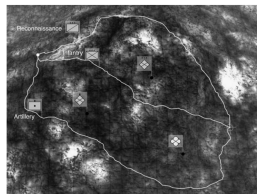
1. Tactical and Strategic AI
 - Tactical Pathfinding
 - Coordinated Action
2. Board game AI

1. Tactical and Strategic AI
 - Tactical Pathfinding
 - Coordinated Action
2. Board game AI

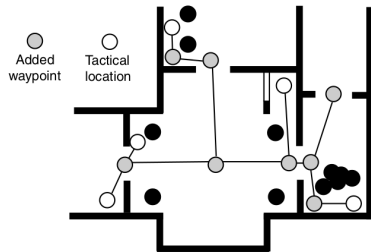
1. Tactical and Strategic AI
 - Tactical Pathfinding
 - Coordinated Action
2. Board game AI

Tactic Weights and Concern Blending

- If a tactic has a high positive weight, then locations with that tactical property will be avoided by the character. Conversely if large negative weight, they will be favoured.
- but! negative costs are not supported by normal pathfinding algorithms such as A* ;
- Performing the cost calculations when they are needed slows down pathfinding significantly.
Is the advantage of better tactical routes for the characters outweighed by the extra time they need to plan the route in the first place?
- Weights have an impact on character personalization also they are not static



- Issue: heuristic may become invalid (not admissible)
Eg: Euclidean distance not anymore valid; maybe multiplying it by a factor?
- graph: nodes derived from influence maps for outdoor and tactical waypoints for indoor
connections: generate them by running movement checks or line-of-sight checks between map locations or waypoints
if grid based: adjacent locations are connected + gradient threshold considerations
- necessary to add additional waypoints at locations that do not have peculiar tactical properties. Superimpose the tactical waypoints onto a regular pathfinding graph.



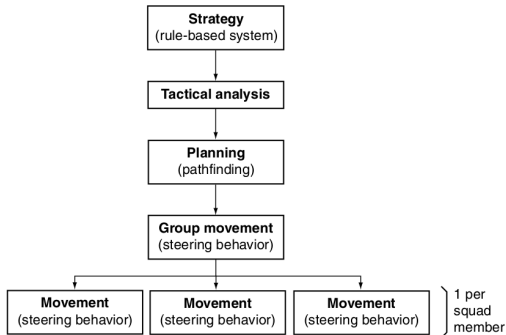
1. Tactical and Strategic AI
 - Tactical Pathfinding
 - Coordinated Action
2. Board game AI

- a whole side in a real-time strategy game
- squads or pairs of individuals in a shooter game.
- AI characters acting in a squad led by the player: cooperation occurs without any explicit orders being given.
Characters need to detect the player's intent and act to support it.

AI characters need to tell each other what they are planning through some kind of messaging system,

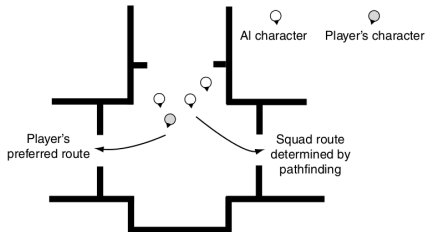
top-down vs bottom-up (emergency) approach

Hierarchy of behaviors: at some levels decisions shared among multiple characters



- Group Decisions
standard expert systems or state machines, tactical analysis or waypoint
tactic algorithms
difference: actions are not scheduled for execution by the character, but
are orders passed down
- Group Pathfinding
different blend of tactical concerns for pathfinding that any individual
would have alone
heuristic of weakest character
decision tree, expert system, or other decision making technology to
detect constraints and separate the group
- Group Movement
steering behaviors (feasibility issues)

- Including the Player
high-level decision making may make a decision that the player subverts



The game design may involve giving the player orders, but ultimately it is the player who is responsible for determining how to carry them out.

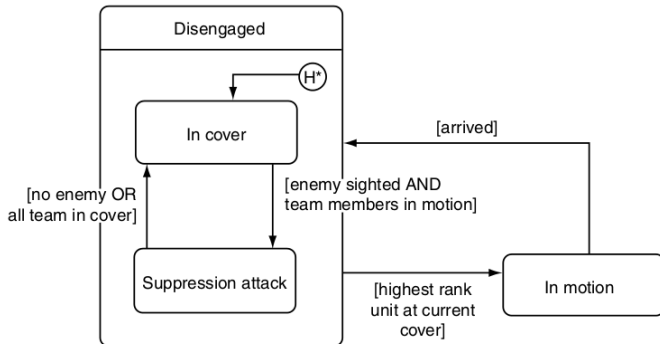
Solution: put the Player at the top level of the hierarchy and conflicts avoided.

- Explicit Player Orders, eg. in RTS games
intent-identification problem

Structuring Multi-Tier AI

- communication mechanism that can transfer orders from higher layers in the hierarchy downward. This needs to include information about the overall strategy, targets for individual characters, and typically other information (such as which areas to avoid because other characters will be there, or even complete routes to take).
- A hierarchical scheduling system that can execute the correct behaviors at the right time, in the right order, and only when they are required.

- Less centralized techniques to make a number of characters appear to be working together.
- Each character has its own decision making
- ↪ the decision making takes into account what other characters are doing
- If any member of the team is removed, the rest of the team will still behave relatively efficiently
- most squad based games



- Scalability: bad if characters have different behaviors they can display.
- Predictability: often hard to predict how the system will evolve. May end up in unintelligent behaviors.

1. Movement
2. Pathfinding
3. Decision making
4. Tactical and strategic AI
5. Board game AI

1. Tactical and Strategic AI
 - Tactical Pathfinding
 - Coordinated Action
2. Board game AI

Board game AI

- different techniques from the ones seen so far
[tree-search algorithms](#) defined on a special tree representation of the game.
- limited applicability for real-time games
- a strategic layer only occasionally used. Eg. making long-term decisions in war games.
- but needed for AI in board games.

- Game theory is a mathematical discipline concerned with the study of **abstracted, idealized games**.
(Only a few definitions from this theory are needed in games.)
- classification of games according to:
 - number of players
 - kinds of goal
 - information each player has about the game.

Number of players

- most of the board games have two players.
- **ply** one player's turn (aka half-move with 2 players)
- **move** One round of all the players' turns (aka **turn**)

Goal

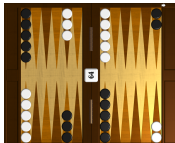
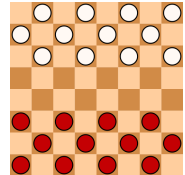
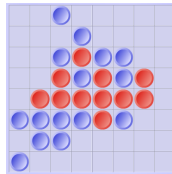
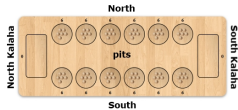
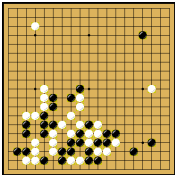
- **zero-sum game**: your win is the opponent's loss ($1; -1$)
trying to win \equiv trying to make your opponent loose.
- **non-zero-sum game**: you could all win or all lose
focus on your own winning, rather than your opponent losing
- with more than two players and zero-sum games, best strategy may not be making every opponent loose.

Information

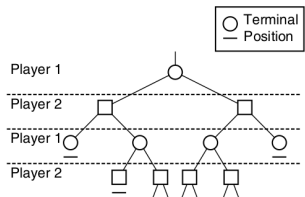
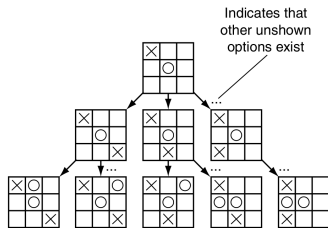
- **perfect information** fully observable environment
complete knowledge of every move your opponent could possibly make
- **imperfect information** partially observable environment
eg, random element that makes unforeseeable which move you and the opponent will take.

Types of Games

	deterministic	chance
perfect information	chess, checkers, kalah go, othello	backgammon, monopoly
imperfect information	battleships, blind tictactoe	bridge, poker, scrabble



For turn-based games: each node in the tree represents a board position, and each branch represents one possible move.



terminal positions: no possible move, represent end of the game. Score given to players

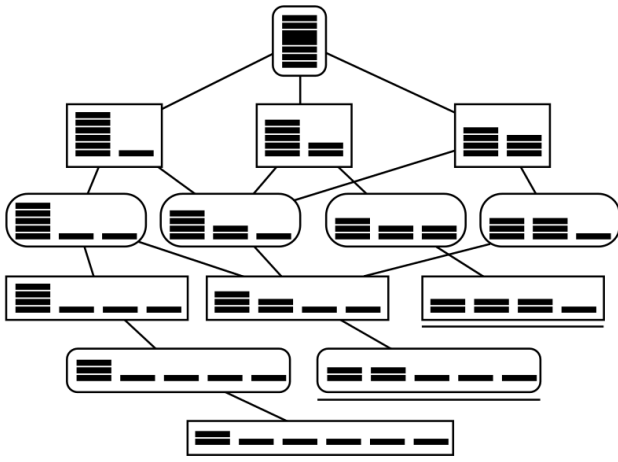
branching factor: number of branches at each branching point in the tree

tree depth: finite or infinite

transposition same board position from different sequences of moves \rightsquigarrow cycles

Example

7-Split Nim: split one pile of coins into two non-equal piles. The last player to be able to make a move wins



Measures of Game Complexity

- **state-space complexity**: number of legal game positions reachable from the initial position of the game.

an upper bound can often be computed by including illegal positions

Eg, TicTacToe:

$$3^9 = 19.683$$

5.478 after removal of illegal

765 essentially different positions after eliminating symmetries

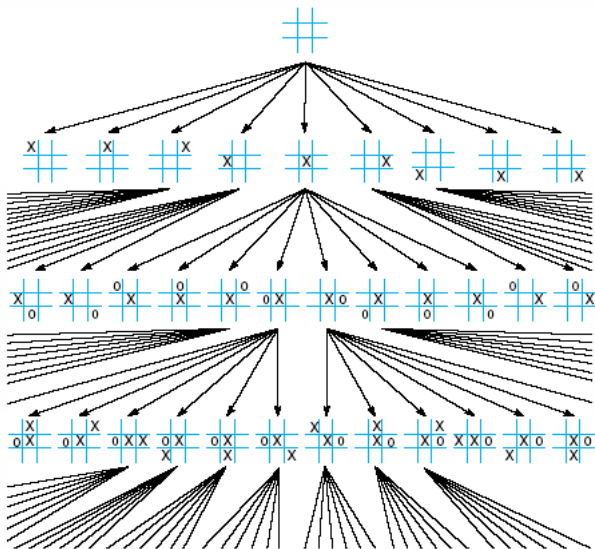
- **game tree size**: total number of possible games that can be played: number of leaf nodes in the game tree rooted at the game's initial position.

Eg: TicTacToe:

9! = 362.880 possible games

255.168 possible games halting when one side wins

26.830 after removal of rotations and reflections



First three levels of the tic-tac-toe state space reduced by symmetry: $12 \times 7!$

