

Strategy Games

Sugih Jamin and John Laird

University of Michigan

12/4/06

Based on articles by Tozour, P., Snook, G., Higgins, D.,
Woodcock, S., McLean A.W., Brockington, M.,
Shoemaker, S., Kent, T., Ramsey, M., and Pottinger, D.C.

*(Game Programming Gems vols. 2, 3, and AI Game
Programming Wisdom vols. 1, 2)*

and Dunnigan, J.F., *The Complete Wargames Handbook*
(revised ed.)

Topics

- Genre Definition and Taxonomy
- History of Wargaming
- Structure and Innovations
- Isometric Tilemaps and Billboarding
- Random Map Generation
- Influence Map: Engaging the Enemy
- Formation, Coordinated Movement, and Chasing
- Influence Map: Terrain Analysis
- RTS AI Decision Making

Genre Definition and Taxonomy

4X: eXplore, eXploit, eXpand, eXTERMINATE!

Resource gathering:

- explicit: Civ I/II/III, WarCraft -> StarCraft -> AoE and descendants
- automated: Koei's games, TotalWar series, Dune II -> C&C, Homeworld
- combat only (wargames): Gettysburg!, Combat Mission, Takeda
- city building: Caesar I/II/III/IV and descendants

Types of play:

- conquer the world: Civ, Warcraft, AoE, Total War
- scenario based: C&C, Cossacks
- play a battle, death match



Control of Time

Turn-based:

- time stops while player decides on action
- player explicitly ends turn
- combat automatically resolved
- player may set up formation and watch combat
- examples: Civ, SMAC, Master of Orion I/II/III, Galactic Civilizations I/II, Europa Universalis, Takeda, Legions

Real-time:

- “micro-turn”/game loop that continuously marches on
- player can pause and issue commands
- examples: Warcraft and family, C&C and family, AoE and family, Homeworld



Control of Time

Hybrid:

- turn-based conquest on world map
- realtime combat on realistic terrain
- example: Koei's games, TotalWar series, Rise of Nation/Legend, Star Wars: Empire at War, LotR: Battle for Middle Earth II



Notable Examples

- **SimCity** (1989, 2: 1995, 3: 1998, 4: 2003)
 - **Stronghold** (1992, Crusader: 2002, II: 2005)
 - **Caesar** (I: 1993, II: 1995, III: 1998, IV: 2006)
 - Pharaoh (1999), Zeus (1999), Middle Kingdom (2002), Children of the Nile (2004)
- **Turn-based Strategy**
 - Koei's Nobunaga's Ambition (1988), Romance of the Three Kingdoms (1988), Bandit Kings of Ancient China (1989), etc. with RPG-like characters and turned-based combat/wargames
 - **Civilization** (I: 1991, II: 1996, III: 2001, IV: 2005/6)
 - Alpha Centauri 1998
 - **Master of Orion** (I: 1993, II: 1996, III: 2003)
 - **Galactic Civilizations** (I: 2003, II: 2004)
- **Real-time**
 - Herzog Zwei – Sega Genesis - 1989
 - **Dune II** (1992, Emperor: 2001)
 - Command & Conquer (1995, Red Alert 1996, Tiberian Sun 1999, Yuri's Revenge 2001, Generals 2003, Tiberium Wars 2007)
 - **WarCraft** (I: 1994, II: 1995, III: 2003)
 - StarCraft (1998)
 - **Total Annihilation 1997, Supreme Commander 2007**
 - **Age of Empires** (1997, II: 1999, III: 2005, Mythology: 2002)
 - Empire Earth (2001, II: 2005)
 - **Homeworld** (1999, Cataclysm: 2000, II: 2003)
- **Hybrids**
 - **Total War** (Shogun: 2000, Medieval: 2002, Rome: 2004, MedievalIII: 2006)
 - **Rise of Nation** (2003, Legends: 2006)
 - **Star Wars: Empire at War** (2006)
 - **LotR: Battle for Middle Earth** (2004, II: 2006)

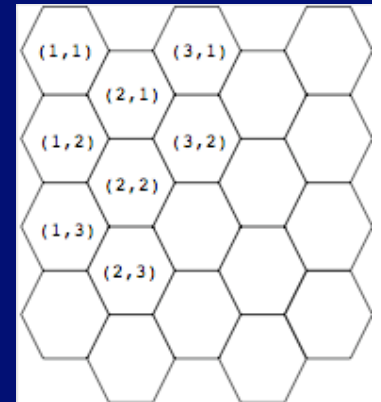
History of Wargaming

Wargaming: “realistic” modeling and simulation of combat

- Chess (antiquity to 18th century)
- Prussian Army's *Kriegspiel*, the first detailed and realistic wargame
- *Kriegspiel* met Operation Research (1940-1965)
- *Kriegspiel* goes digital (1980 on)

Types of Wargaming

- A bunch of people sitting in a room, each playing a different role, the “game” scenario evolves over time and the players react in-role
- A hex grid board with terrain drawing, some cardboard units with stats written on them, a dice, and a rulebook
- Table-top miniature terrain with realistic “toy soldiers”
- Computer based: less simulation, more fun



Why Play Wargames

- To experience history
- To play “What if?” scenarios:
 - to indulge in historical wishful thinking
 - and to help future planning:
 - consider various nuclear first-strike scenarios: conclusion MAD
 - war planning
 - disaster response
- To indulge in micro-management
- To play god (or just an emperor)

Dunnigan: “Wargaming is the hobby of the overeducated”

Game Structure

- Units (including buildings):
 - Produce/gather resources
 - Research technology
 - Create units
 - Fight and defend
- Map
 - Different terrain
 - Strategic chokepoints
 - Building locations
 - Random maps
- Tech tree and build tree: hierarchies of technology and units

Goals

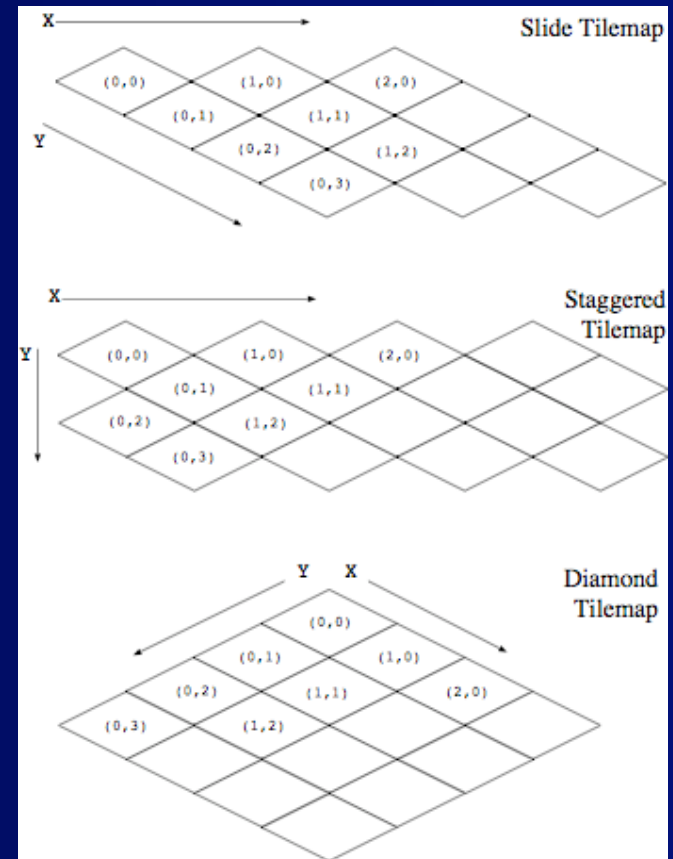
- Long term: dominate opponents
 - Militaristic: control strategic points, destroy opponents
 - Cultural: first to find a religion, create a wonder, or reach final civilization age
 - Economic: control world resources
 - Diplomatic: force diplomatic victory
- Medium term:
 - Move up tech tree
 - Occupy territory
 - Accumulate wealth
- Short Term Goals
 - Win battles
 - Collect resources
 - Create new units

Some Notable Innovations

- Isometric tilemap
- Fog of war (Warcraft II)
- Issuing commands when paused
- Faction-specific tech tree
- Faction-specific units
- Better AI (path planning, strategy, ...)
- Waypoint
- Build queue
- Auto-tasking and city management
- 3D Terrain (Total Annihilation)
- Commander unit (can gain experience and evolve, borrowing from RPG--TA)
- Formation
- Diplomacy
- Religion

Isometric Tilemaps

- Isometric (2 1/2D, god view): no arches, no two points with the same (x, z) coordinate with different y 's
- Largely superceded by 3D terrain, except on hand-held devices, e.g., mobile phones, PDAs, Gameboy
- Diamond tilemap used in AoE, C&C, CivIII



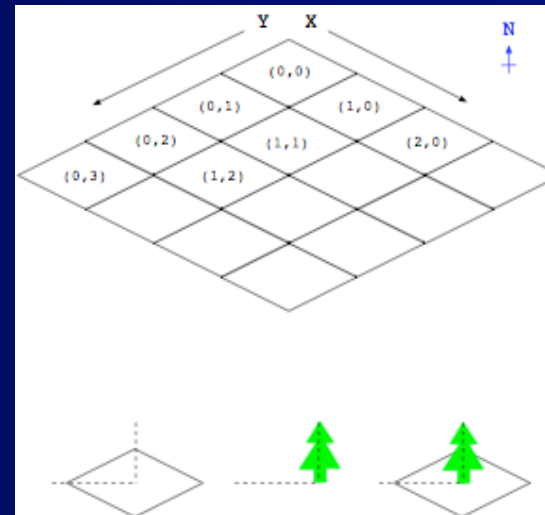
Diamond Tilemap

- Tile size: width (w_t) usually double the height (h_t), e.g., 64x32
- Mapping tilemap coordinate (x^m_i, y^m_i) to world coordinate (x^w_i, y^w_i) :
 - $x^w_i = x^w_0 + (x^m_i - y^m_i) * w_t / 2$
 - $y^w_i = y^w_0 + (x^m_i + y^m_i) * h_t / 2$

Diamond Tilemap

- Tile walking:

Direction	Dx	Dy
N	-1	-1
NE	0	-1
E	1	-1
SE	1	0
S	1	1
SW	0	1
W	-1	1
NW	-1	0



- Placing sprite: put tile anchor in the center of the tile, put sprite anchor at the “base” of the sprite

Billboarding

Even when 3D terrain is used, can use billboarding to fake 3D objects using sprites:

- Put sprite on a polygon (billboard)
- Billboard polygon always faces the current viewing angle:
 - polygon rotates along y-axis to always face viewing angle
 - matrix multiply billboard polygon with the transpose of the viewing matrix

To achieve more 3D look:

- use multiple depth slices of a 3D model
- put each slice on a separate billboard
- stack them together to give illusion of depth

Random Map Generation

Method used in *Empire Earth*:

- assume a world of water
- first place players roughly equi-distant from each other
- then grow a cluster of land around each player, in a round-robin fashion, until a given cluster size is reached
- for each land cluster, grow a cluster of flat land
- add elevation to the rest (e.g., using fractal algorithm)
- place resources and initial units

	Ⓟ	P1					
P1	P1	P1					
					P2	P2	
					P2	Ⓟ	
					P2		

F	F	P1	P1	P1			
F	Ⓟ	F	P1	P1			
P1	F	F	P1	P1			
P1	P1	P1					
	P1				P2	P2	P2
				F	F	F	P2
			P2	F	F	Ⓟ	F
			P2	P2	F	F	F

Engaging the Enemy

From Sun Tzu's *Art of War*:

Attack where the enemy's defence is weak. Defend from an impregnable position. In attacking, do not let the enemy find a strong position of defense. In defending, leave the enemy at a loss as to where to attack.

If you outnumber the enemy ten to one, surround them; five to one, attack them; two to one, divide them. If you are equally matched, take the offensive. If the enemy forces are slightly stronger, prepare for defence; if vastly superior, retreat.

The strength of an army does not lie in mere numbers. The general who is able to use all available forces to best advantage and to anticipate enemy moves correctly will surely be successful.

An attack should have the forceful impact of a grindstone crushing eggs. To do this the strength and weaknesses of the enemy must first be ascertained.

Strategic Dispositions

How can the RTS AI ascertain the strength and weaknesses of the enemy?

- Where is the enemy's defence the weakest?
- Where would be a strong defensive position for the enemy?
- What is the number of enemy forces and where are they?
- How to anticipate enemy moves?

Solution: use an *influcence map*

Influence Map

- An *influence map* is a spatial representation of an AI agent's knowledge about the world
- It indicates the part of the game world under each (human and computer) player's control:
 - where a player's units are deployed
 - where the enemies' units are deployed
 - the area under the “influence” of each player
 - where the frontiers between the forces are
 - where the areas of “no man's land” are
- It is also useful for characterizing different locations:
 - areas of high strategic control
 - weak spots in one's or opponent's defenses
 - defensible “camping” location
 - strategically vulnerable areas and choke points for laying ambush

Influence Mapping

A simple way to compute influence map:

- assign a positive “influence” value (e.g., combat strength) to each player unit, and a negative value to each enemy unit
- Influence propagation: halve the value for each tile away from the unit
- for each tile, add up the influence of both players
- the frontier between players is where the influence values of two adjacent tiles switch sign
- a zero value indicates no man's land

0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0
2	2	2	1	-1	-1	-1	-1
5	5	2	0	-1	-2	-2	-2
10	5	1	0	-2	-3	-5	-2
5	5	1	-1	-2	-4	-3	-2
2	2	1	-1	-5	-3	-1	-1
1	1	0	-1	-2	-2	-1	0

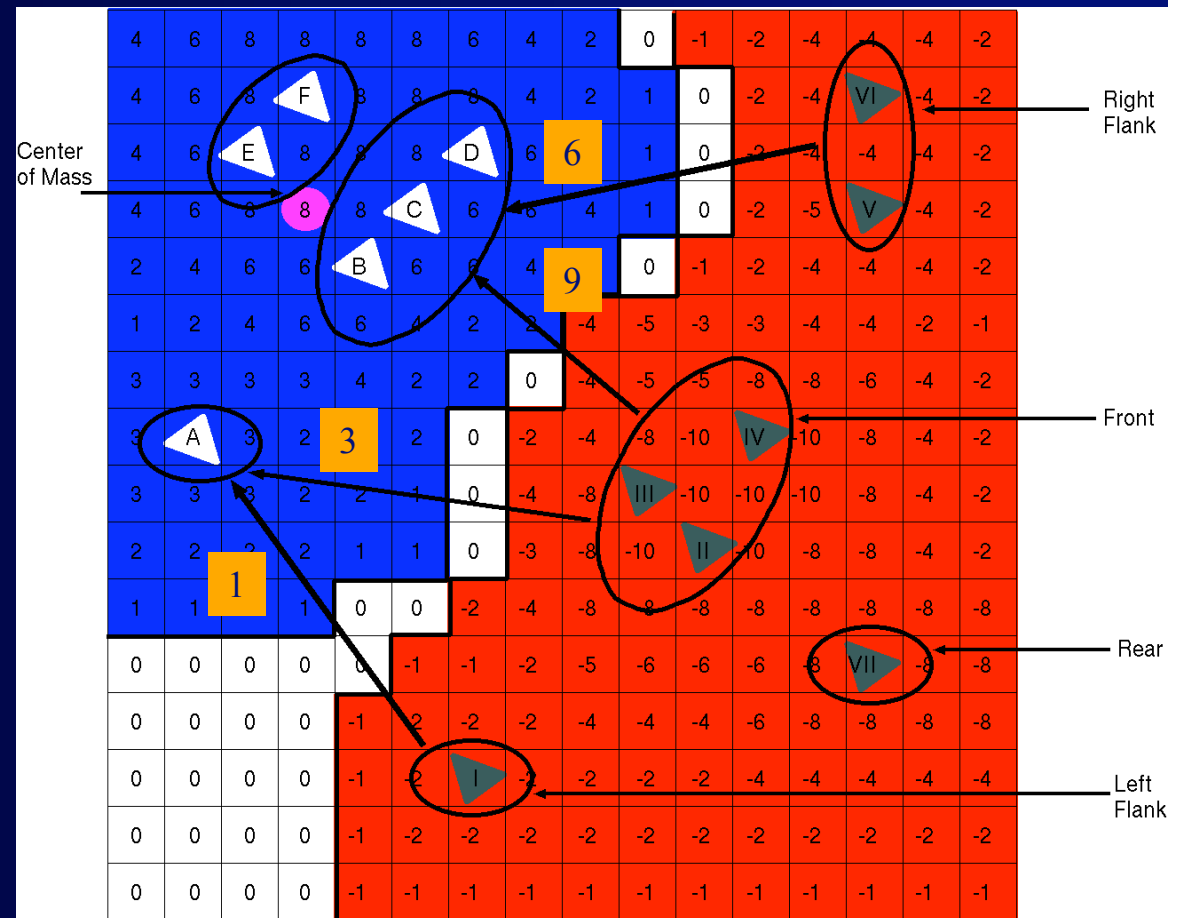
Enemy's Strategic Dispositions

- **Gradient:** difference in influence values of two tiles
- **Center of mass:** the center of tiles with highest influence value
- **Front:** enemy unit whose tile has largest difference in gradient with our center of mass. Group all units within two tiles from **front** as part of *front*
- **Flank:** units not part of front, group units within two tiles of each other as part of *flank*. There can be more than one flank.
- **Rear:** if shortest path between a flank and our center of mass passes through the front, then the flank is actually a *rear*.

Red Attack

Red is on the offensive:

- maximize contact points
- move away from our center of mass
- move “uphill” towards enemy’s center of mass



Influence Value

In addition to, or aside from, combat strength, the influence value can take into account:

- terrain type
- visibility (fog of war)

Influence propagation can take into account:

- unit range: range vs. melee units
- unit facing: can get “flanking” attack for free?
- unit type: airborne, seafaring

The size of a “tile” may be larger than game world tile, to reduce computation load

Desirability Value

Best offensive position:

- high passability (plain, plateau)
- high cover and low visibility (boulders, forest)
- good line-of fire (high ground)

Worst defensive position:

- low passability (mountain pass)
- high visibility and low cover (plain, valley, passes)

Terrain Analysis

Terrain analysis: gathering information about the environment

What types of “area” to recognize?

Static areas: pre-processed

- Continents: both land and water, determines accessibility
- Hills: height advantage
- Shore tiles: shore waves, transport drop off/pick up points

Dynamic areas: makes game world more interesting, but re-processing takes time:

- Forests: harvesting, wall building, hide units for flanking and ambush
- Towns: help AI knows where to attack, how to attack (coastal?)
- Herds, armies: group movement and behaviors

Convex Hull

Basic tool for terrain analysis

Roughly: a geometric shape (e.g., a polygon) where a straight line between any two points inside doesn't go outside

Represent areas as convex hulls: more accurate than bounding boxes, less expensive than keeping track of the actual shape

Choke Points

Choke points:

- narrow routes connecting two areas
- good guard locations
- good places to set ambush
- fortress/wall building locations
- but also where *not* to build to prevent blocking oneself in



FIGURE 3.4.4 Ambush in a choke point.

Computing choke point:

- use convex hull to represent areas (grouped tiles)
- compute influence map emanating from each area
- choke point is bad defensive position adjacent to good offensive positions in frontier/no man's land area

Terrain Analysis using Influence Map

Fortress/wall: placed near vulnerable assets or choke points

Town center: placed in least vulnerable area closest to resource

Barracks: placed in least vulnerable area closest to enemy

Vulnerable asset: placed in least vulnerable areas farthest from enemies and least accessible areas (map edges and corners)

Exploration: have an “unseen interval” attribute added to tile, explore those with longest unseen interval (if FoW used)

Formation and Coordinated Unit Movement

To avoid stacking units in group or formation:

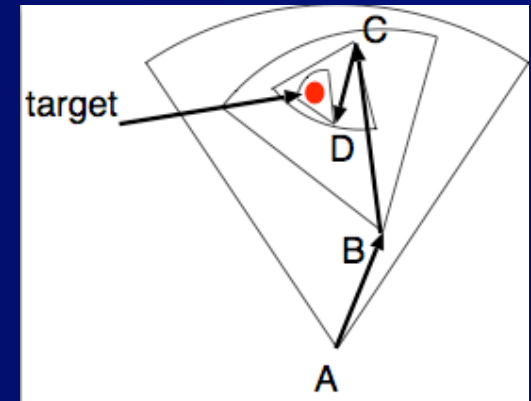
- use flocking rules: separation, cohesion, alignment, avoidance
- prioritize units
- use explicit unit-to-unit messaging
- use environment embedded information (reserved path)

It may be necessary to break up a group or formation to take multiple paths, but beware of ambush!

Hunting Down the Player

When chasing hidden player:

- don't head for player units directly
- make it look like the AI is “searching” for the target by *exploring* the world
- go to last known position or head in the general direction of the player but to the side or overshoot
- re-evaluate more frequently if player is moving
- allow player to “lure” the AI into regions or behind obstacles and then double back to escape
- when the player becomes visible, attack



RTS AI Strategic Decision Making

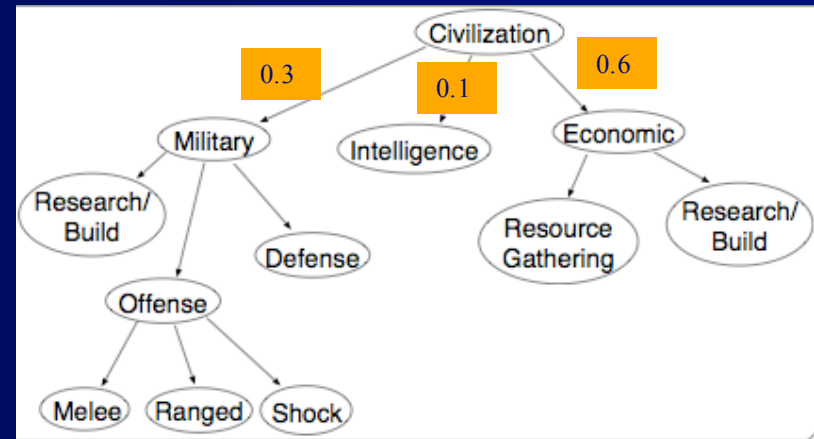
How does an RTS AI decide what to do next?

- use resource allocation tree
- use tech/building dependency graph

Resource Allocation Tree

Resource allocation tree:

- record current allocation and targeted allocation
- balance expenditure, maintain balance after unit loss (at parent node)
- spend based on faction stance and characteristics
- record which node has been effective or has been attacked in the past to learn enemy (player)'s behavior
- mimic player's recent past decisions for auto-governor (CivIII)
- difficulty: how to measure the “value” of a unit?
- both value and allocation can be tweaked to give an AI a distinct “personality”



Tech/Build Dependency Graph

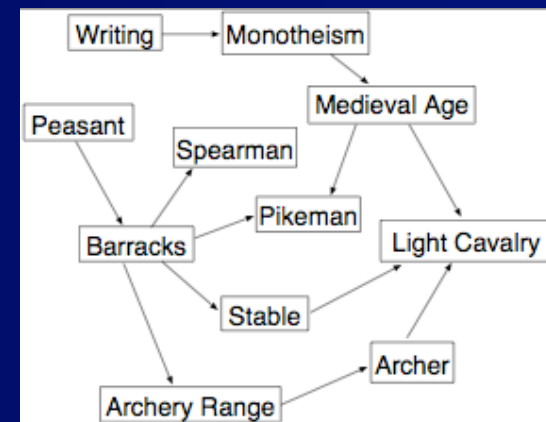
Tech tree and build tree form *dependency* graphs

Choose a building or tech to build deep in the dependency graph, the AI then follows the steps of the dependency graph

For example, starting from Peasants, telling the AI to build Light Cavalry causes almost everything else in the techtree to be built/researched

Resource dependencies can be embedded into the dependency graphs

AI can be initialized with target technologies to create personality bias



Build Dependency Graph

Build tree can be used to choose target:

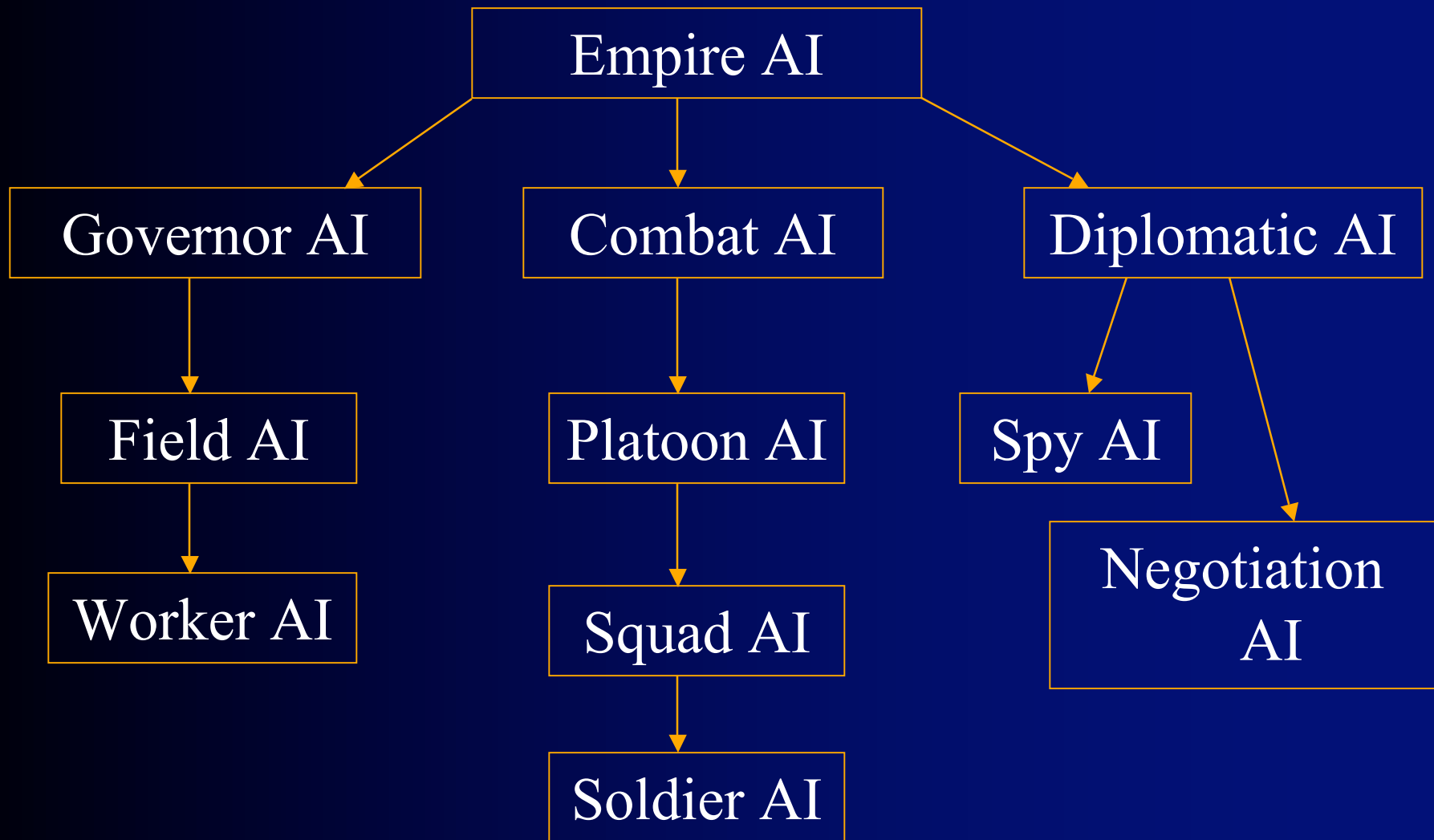
- **Intrinsic value:** buildings deeper in the graph usually have higher intrinsic value
- **Child dependencies:** taking out unit creation buildings and support buildings (power plant) can cripple the enemy
- **Parent dependencies:** prune a whole branch of the tree to disable creation of certain units; for defense, create multiple instances of the parent, and protect them
- Vulnerability value of various nodes can be tweaked to give an AI a distinct personality

Putting It All Together

- Influence map tells you where the enemy is
- Resource allocation tree tells you the forces you need
- Dependency graph tells you how to build the forces you need

How much would/should you cheat by looking at the player's influence map, resource allocation tree, and dependency graph?

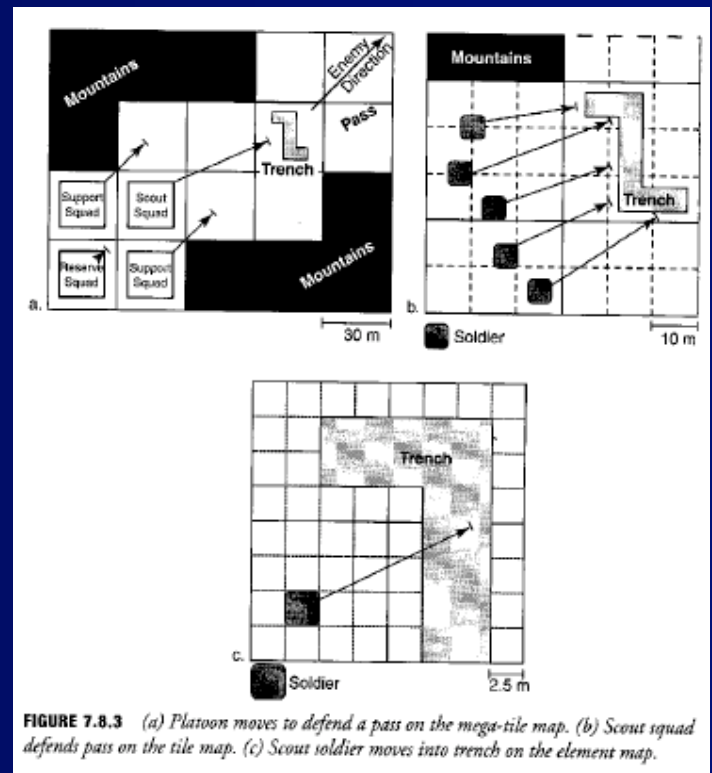
Use Hierarchical AI



Use Hierarchical AI

Each AI may have its own terrain analysis and influence map, at different levels of detail

- Empire AI decides which territory to conquer next
- Squad AI order units on battlefield map
- Unit AI performs local path finding
- Negotiation AI can also use influence map to decide when to negotiate and what to negotiate for



Final Thoughts on RTS AI

There may not be enough CPU power for your AI:

- separate tasks into periodic and constant tasks
- distribute computation across frames if possible
- use level-of-detail AI:
 - units interacting with players need most accurate AI
 - visible but non-interacting units must be believable
 - punt on the others