

Children of Galaxy AI

What is Children of Galaxy

Sci-fi 4X turn based strategy game (Explore, Expand, Exploit, Exterminate)

- Sid Mayer's Civilization, Galactic Civilizations, Master of Orion, Total War

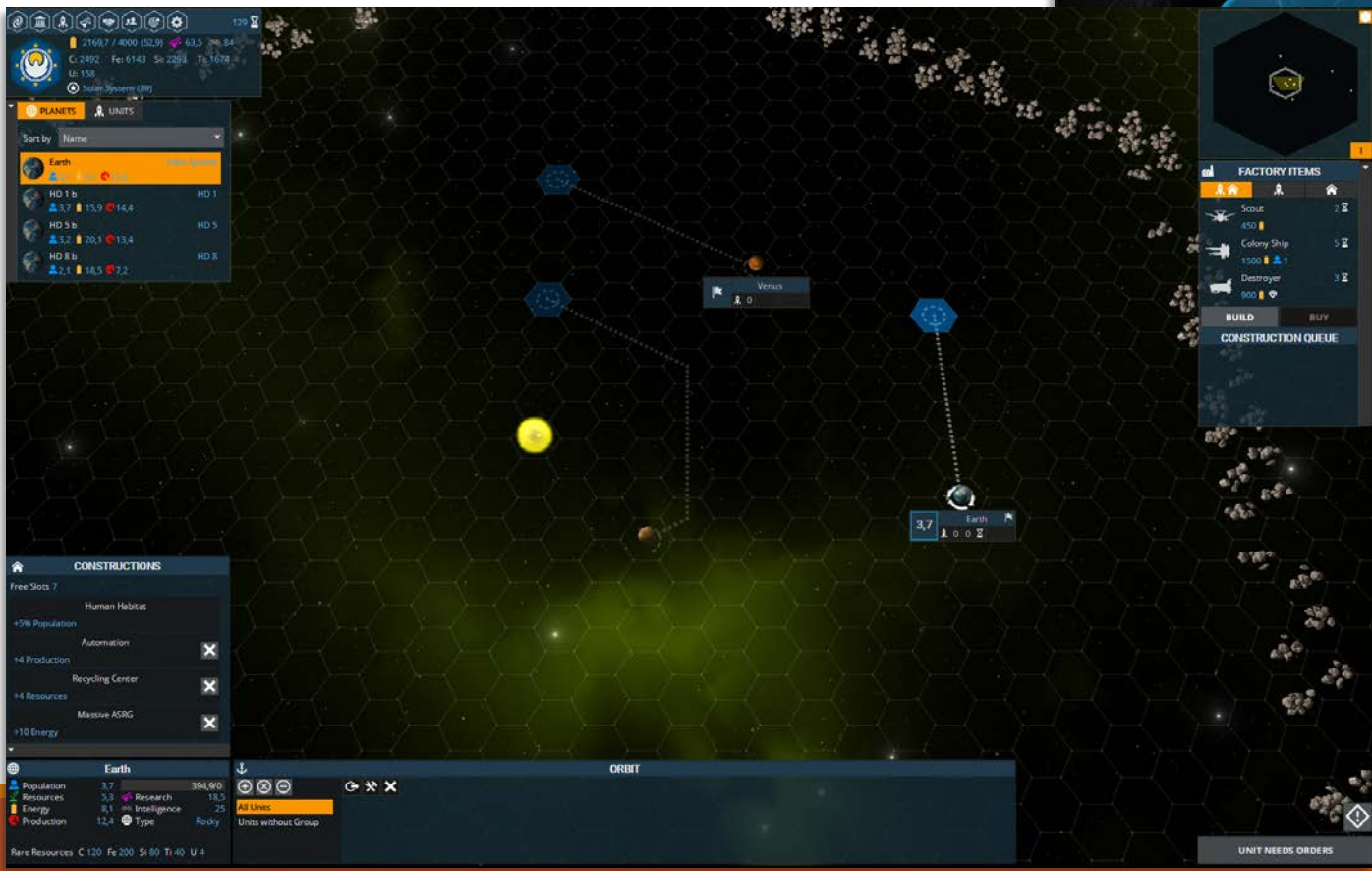
Developed by one man - Filip Dušek (EmptyKeys)

Greenlit on Steam – EA in 2017

Standard 4x/Grand strategy features

- 4 Unique Races
- Tactical Combat - 7 unique combat classes
- Procedurally generated Galaxy
- Planet Colonization and Customization
- Research Technology Tree
- 3 possible game victories - expansion, domination and research
- Diplomacy (currently basic)
- ...

2 LoDs



AI challenges

Many very different interconnected types of reasoning (similar to RTS)

- Micromanagement
- Spatial reasoning (tactics)
- Production
- Research
- Diplomacy
- Strategy (win conditions)
- Fog of war

It should all be believable

- Different races/opponents should have certain personality

It should have different levels of difficulty

Moving units - macro scale

Turn based = plenty of time to search... right?

Not really - 3 sec per civ with 7 civ = 21 sec

Raw movement search

- 7 Actions for every ship (6 adjacent hexes + stay)
- About 200 ships?
- 7^{200} branching factor (there are 8 players total)
- *(Actually more than 7 actions since you can move further in one turn)*

Search for solar systems

- About 50 systems on border
- 1 action - DEFEND my systems, ATTACK enemy systems
- 50^{200} branching factor

How to improve search

Reduce number of systems

- Not all systems are relevant
- Generate *tasks* for them if they become relevant
- E.g. if I see enemy approaching my system I want to defend it
- If I see poorly guarded system I want to attack etc.

Reduce number of ships

- Create squads and plan for whole squads
- Reduce precision or try every combination

How do others do it – Rome II: Total War

Loosely based on Belief-Desire-Intention framework

Data driven – to balance and define personality

1. Get current context

- How are we doing?
- Diplomacy

2. Movement

- Generate tasks
 - Many generators with their reasons and weights.
 - Pattern matching: If you see X propose Y.
- Allocate armies – Determine who will do what – MCTS
 - Get prioritized list of tasks – determine how to satisfy them best
- Coordinate armies – Determine who does what this turn – MCTS
 - Army A should attack city X. Army Y blocks the way. Should we go around or is there another army we can work with to destroy Y?
- Recruit new units – which units and where – MCTS

Rome II: Total War

3. Building

- Templates
 - Another way to get personality
- Building scoring
 - Multi-layer system, synergies...
- Importance of given city
- Each city tries to match its pattern (or a global one)

Research works in a very similar fashion

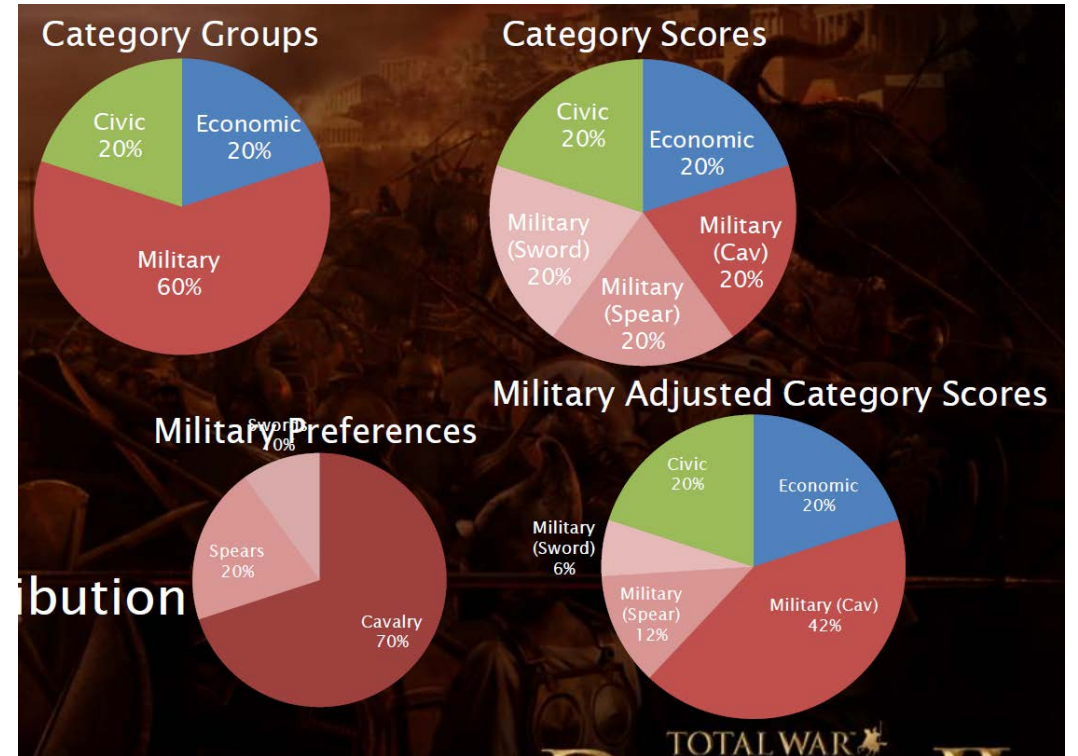


Image from Divide and Conquer, The Campaign AI of Total War presentation at Game AI Conference

Current AI

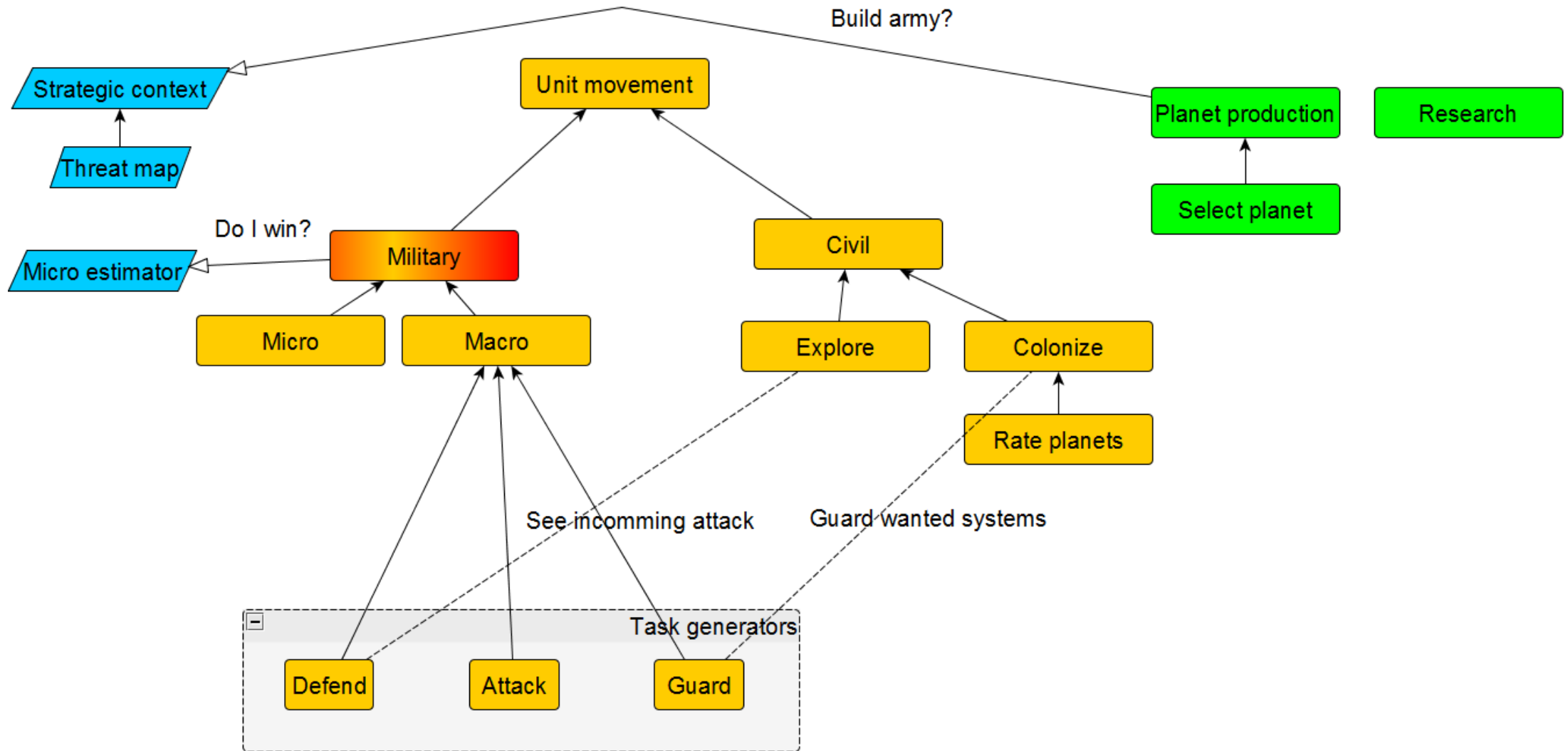
Behavior trees

- Everything is scripted with some random element
- Huge room for improvement

The screenshot displays a game engine interface with several panels:

- Code Editor (Left):** Shows XML code for a behavior tree. The code includes elements like `<Behavior Name="Brain">`, `<Sequence>`, `<Behaviors>`, `<Option>`, `<Considerations>`, `<DecayCurve Max="1" K="3">`, `<ExplorationRatioValue Ratio="8" />`, `<FindPlanetForBuild />`, `<PlayerSetPlanetBehavior BehaviorName="BuildScout" />`, `<LogCurve Middle="5" Denominator="10">`, `<ColonizationRatioValue Ratio="2" />`, `<FindPlanetForBuildColonyShip DistanceSystemUtilityCoefficient="100"/>`, `<PlayerSetPlanetBehavior BehaviorName="BuildColonyShip" />`, `<DecayCurve Max="20" K="3">`, and `<PlayerValue Property="EnergyIncome" />`.
- Overview (Top Left):** A small thumbnail of the behavior tree.
- Neighborhood (Middle Left):** A panel showing a small map or neighborhood view.
- Structure View (Bottom Left):** A search bar and a list of nodes, including `UnitsOnOrbes`, `UnitsTargetRai`, `UnitsTargetSy`, and `Inventer`.
- Main Canvas (Center):** A large, detailed behavior tree diagram. The root node is a purple box labeled "Behavior Name = UnitBehavior". It branches into several nodes, including "Sequence", "Option", and "Consideration". The tree is complex, with many nodes and edges, representing a highly scripted AI system.
- Palettes (Right):** A vertical toolbar with various shapes and icons, including "Shape Nodes", "Modern Nodes", "Edge Types", "Group Nodes", "Swimlane Nodes", "People", "Computer Network", and "UML".

How do we want to do it - architecture



Task

3 types – Attack, Defend, Guard

Attack

- How many units are needed to attack
- Success probability with given composition (allocation)

Defend

- Do I defend my system or do I defend against a threat (enemy fleet)?
- Influence map saying where threats are

Guard

- Identify desirable planets/systems
- Based on enemy influence send units to guard

How to implement it

Extend current behavior tree architecture

“Update context” or “Move units” may be nodes in the tree

It somehow works now - we can improve one node at a time

Game written in C# - call C++ MCTS from C# (very few calls, small overhead)

Thank you for you attention :)
