Stavros Vassos, University of Athens, Greece stavrosv@di.uoa.gr May 2012

INTRODUCTION TO AI STRIPS PLANNING

.. and Applications to Video-games!

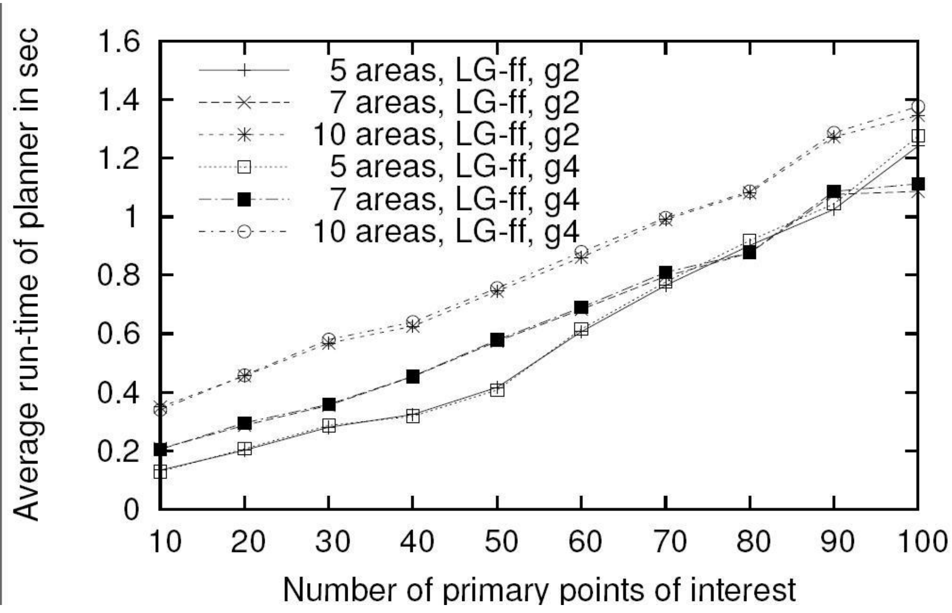
Project ideas

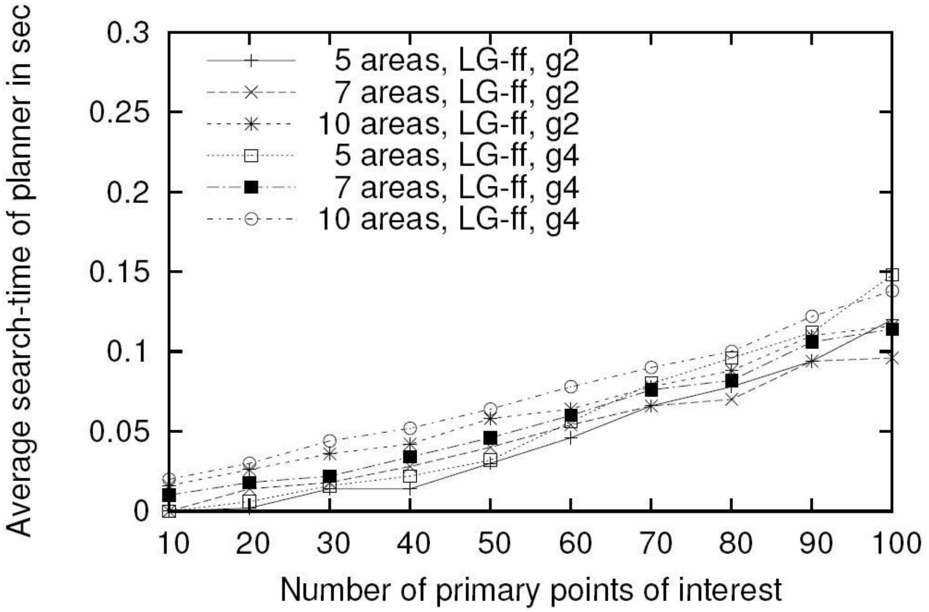
- Smart pre-processing
- PDDL–Golog synergy
- FSM–BTs synergy
- Replanning in Unity3D
- STRIPS extensions in Unity3D
- STRIPS in Valve’s GE

Project ideas: Smart pre-processing

Project ideas: Smart pre-processing

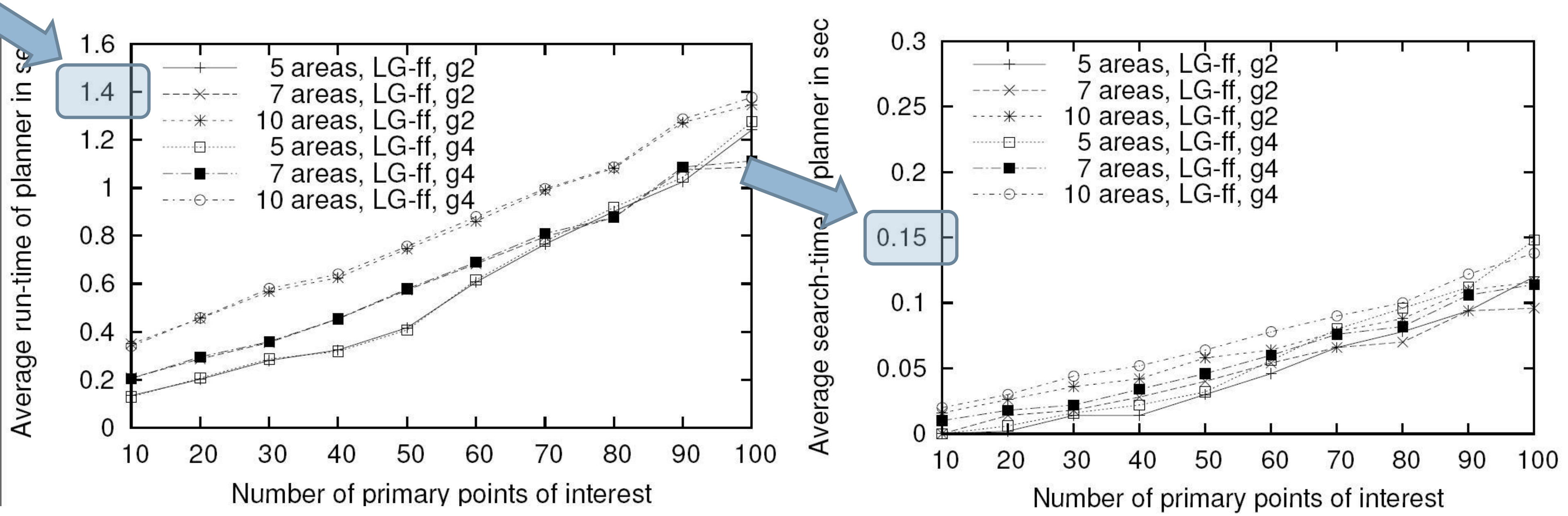
- Academic planners rely on serious pre-processing
  - FastDownward compiles the problem into a SAS format (*)





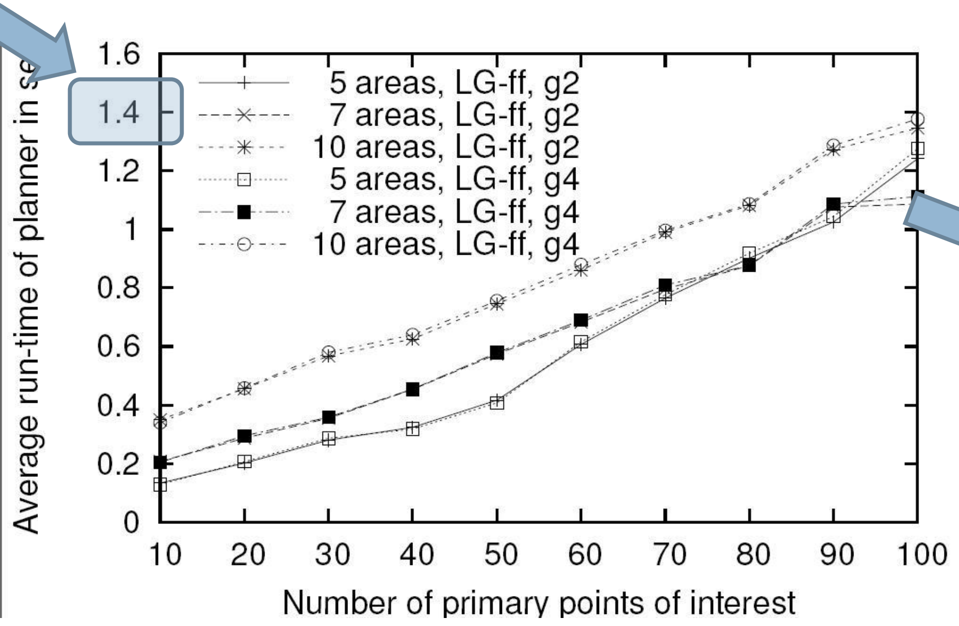
Project ideas: Smart pre-processing

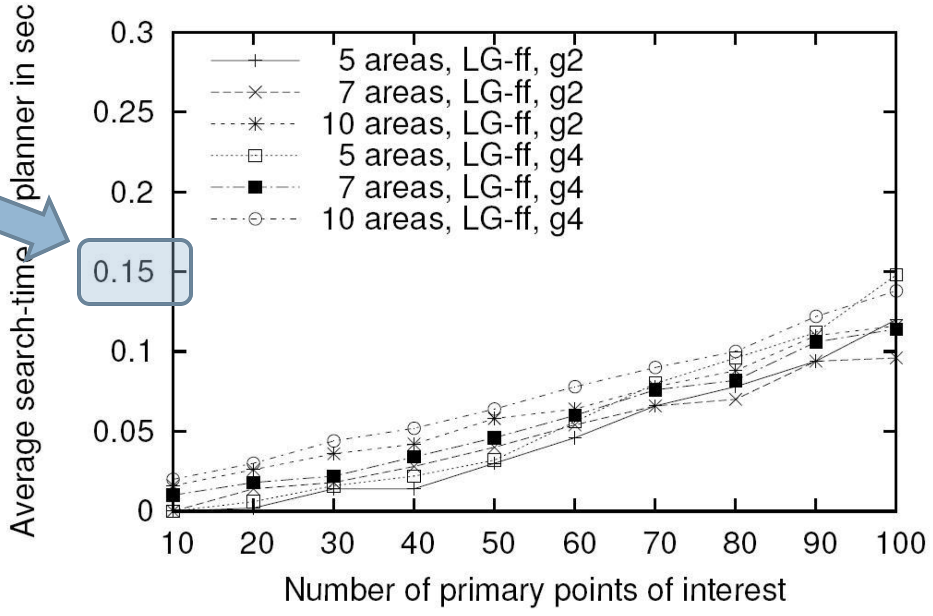
- Academic planners rely on serious pre-processing
  - FastDownward compiles the problem into a SAS format (*)



Project ideas: Smart pre-processing

- □ Academic planners rely on serious pre-processing
  - FastDownward compiles the problem into a SAS format (*)
- □ Maintain such a structure for many NPCs and update the information instead of re-compute every time





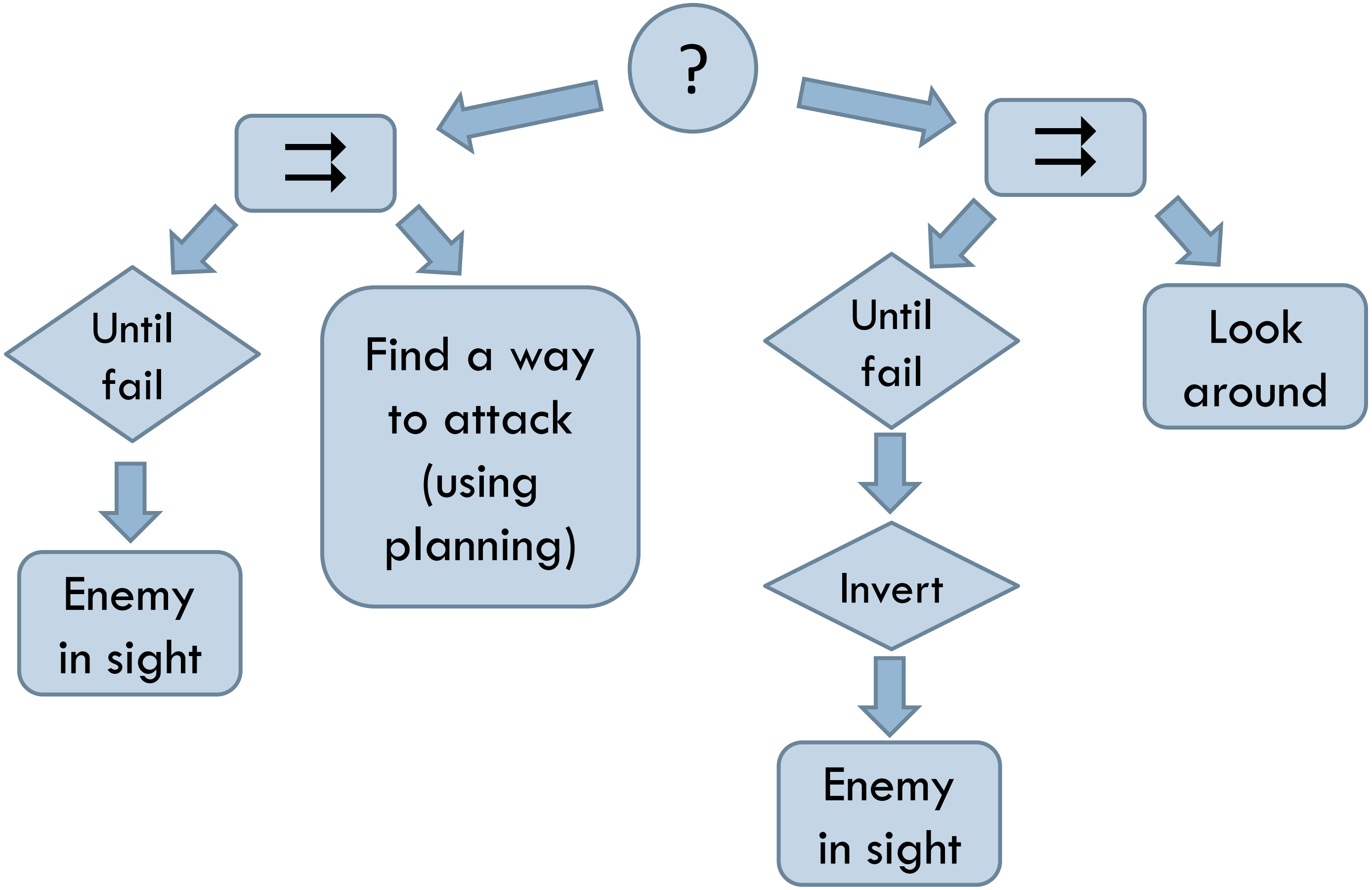
Project ideas: PDDL–Golog synergy

Project ideas: PDDL–Golog synergy

- Simple Golog implementation in Prolog
  - http://www.cs.toronto.edu/cogrobo/main/systems/index.html
- Simple PDDL parsing and planning in Prolog
  - Provided in the files of Lecture 3
- Adapt the Golog implementation to take as input a PDDL domain/problem instance and solve wrt a Golog program instead of the goal

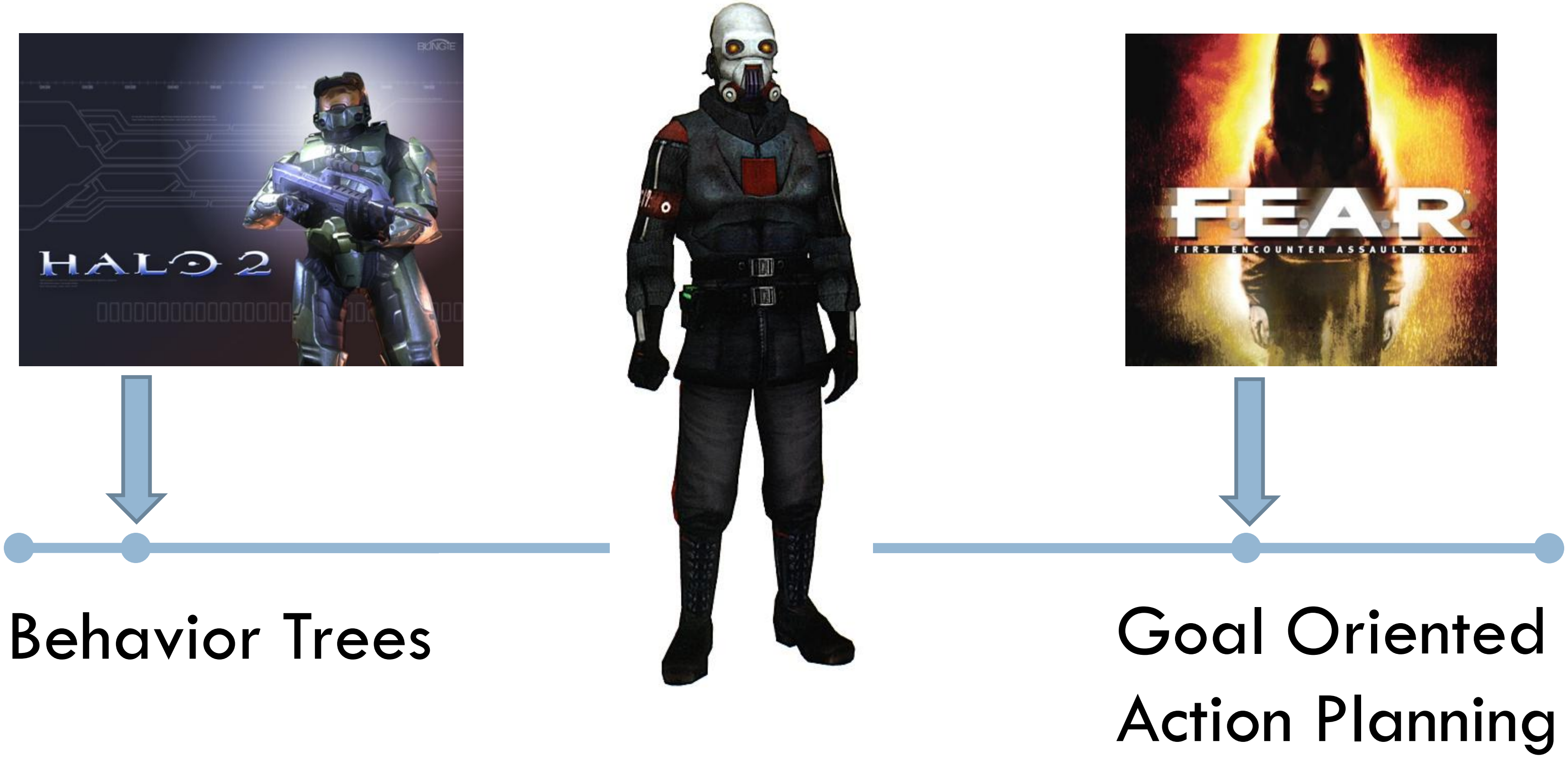
Project ideas: PDDL–Golog synergy

- Develop a visual language (and GUI) for specifying Golog programs based on BTs!



Project ideas: FSM–BTs synergy

Project ideas: FSM–BTs synergy



Behavior Trees

Goal Oriented Action Planning

- A combination of these techniques?
  - BTs for reactive decision making
  - GOAP for tactical decision making

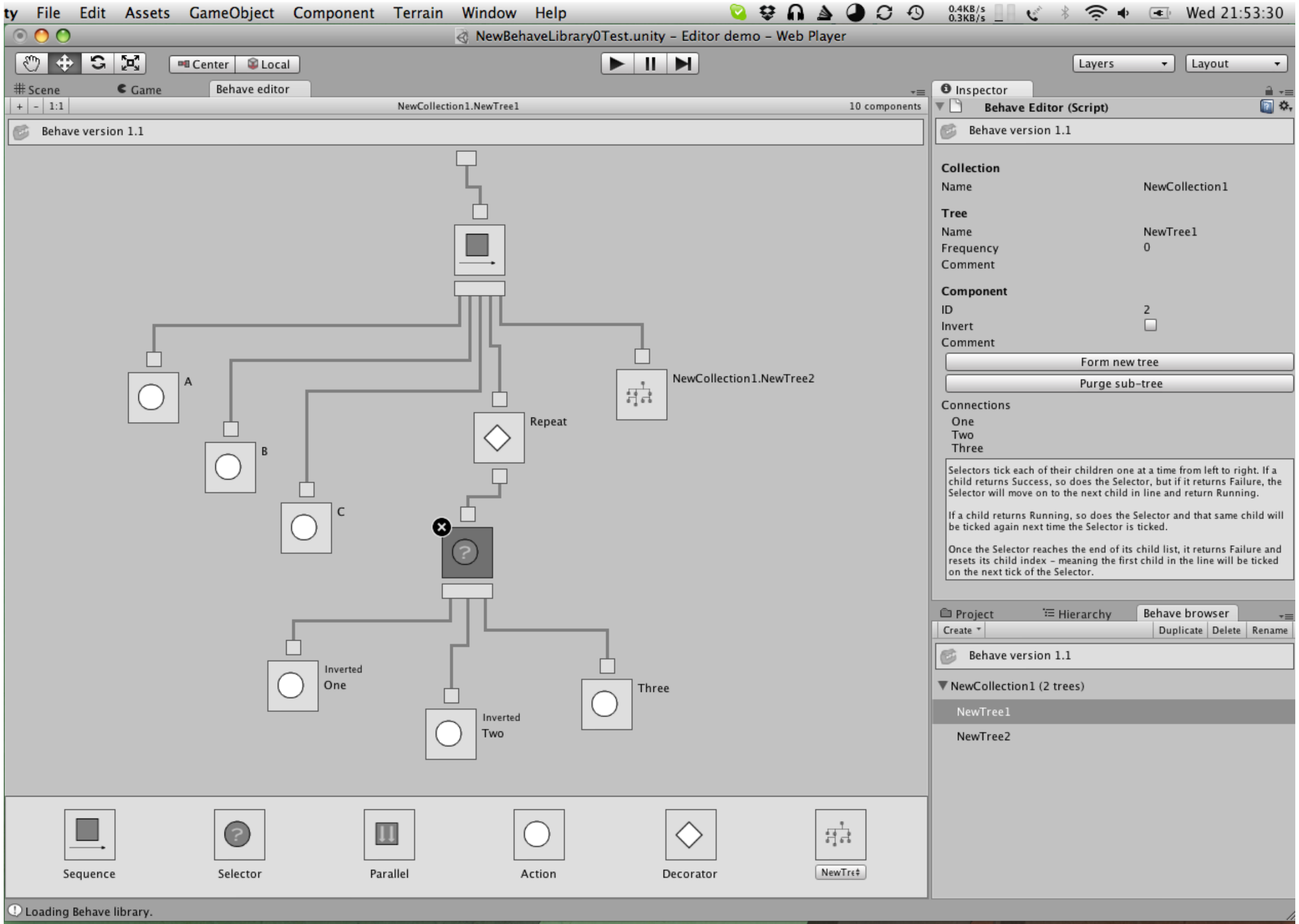
Project ideas: FSM–BTs synergy

- "Behave" tool for Unity3D by Angry Ant
  - http://eej.dk/angryant/behave/
  - Visual interface for building BTs
  - Generates code to be used in Unity3D

- "iThink" tool for Unity3D by [Anastassiou, Diamantopoulos, Vassos, Koubarakis 2012]
  - https://code.google.com/p/ithink-unity3d/

Project ideas: FSM–BTs synergy

- “Behave” tool for Unity3D by Angry Ant



Behave version 1.1

Collection

Name	NewCollection1

Tree

Name	NewTree1
Frequency	0
Comment	

Component

ID	2
Invert	☐
Comment	

Connections
 One
 Two
 Three

Selectors tick each of their children one at a time from left to right. If a child returns Success, so does the Selector, but if it returns Failure, the Selector will move on to the next child in line and return Running.

If a child returns Running, so does the Selector and that same child will be ticked again next time the Selector is ticked.

Once the Selector reaches the end of its child list, it returns Failure and resets its child index – meaning the first child in the line will be ticked on the next tick of the Selector.

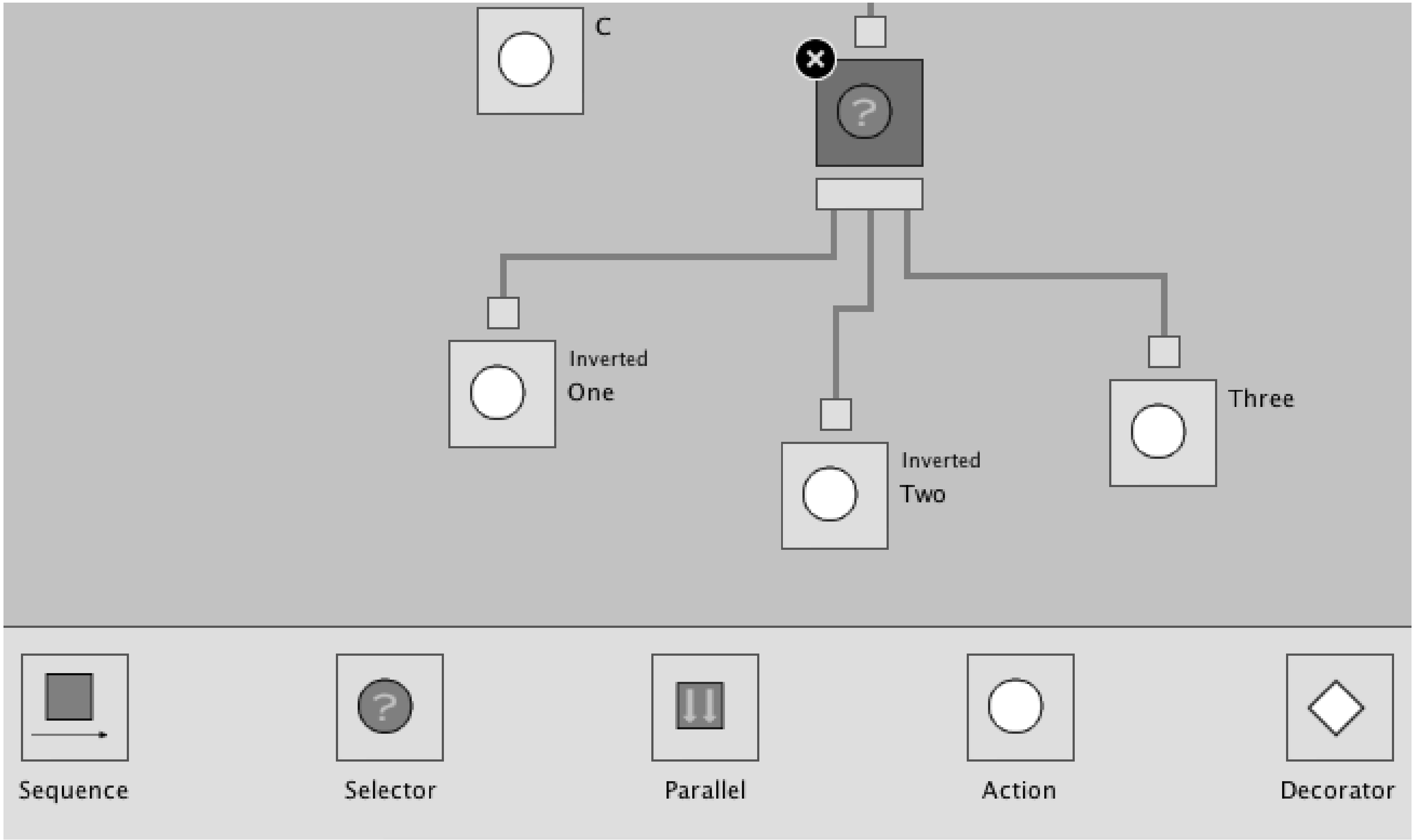
Behave version 1.1

▼NewCollection1 (2 trees)
 NewTree1
 NewTree2

Loading Behave library.

Project ideas: FSM–BTs synergy

- “Behave” tool for Unity3D by Angry Ant



Project ideas: Replanning in Unity3D

Project ideas: Replanning in Unity3D

- Use iThink-Unity3d as the basis
- Extend it with a sensory system that updates the current state according to changes in the game-world

Project ideas: Replanning in Unity3D

- Use iThink-Unity3d as the basis
- Extend it with a sensory system that updates the current state according to changes in the game-world
- Implement some simple policies for execution monitoring and replanning
  - Check the next n actions in the plan and verify that they can be executed
  - Extend the planning system to include conditions at each step that need to be true for continuing the execution

Project ideas: STRIPS ext. in Unity3D

Project ideas: STRIPS ext. in Unity3D

- Adopt an open world assumption
- Adopt functionality similar to PKS
- Introduce features that are important for a particular game genre, e.g., durative actions
- …

Project ideas: STRIPS in Valve's GE

Project ideas: STRIPS in Valve's GE

- □ Similar to iThink-Unity3d but now in a real game!
- □ C++
- □ Adventurous as there is only some basic documentation (but also an active forum)
- □ It's the ~~best~~ only way to convince people in the game industry that your technique works

Project ideas: STRIPS in Valve’s GE

- Valve’s game engine already uses the terminology used in academic planning
  - Condition → literal
  - Task → action
  - Schedule → plan
- The thinking function of NPCs uses a nice execution monitoring mechanism that decides which plan to follow and when to stop executing a plan
- But instead for searching for plans, a small set of pre-defined schedules are used

Project ideas: STRIPS in Valve's GE

- https://developer.valvesoftware.com/wiki/AI_Programming
    - https://developer.valvesoftware.com/wiki/Conditions
    - https://developer.valvesoftware.com/wiki/Task
    - https://developer.valvesoftware.com/wiki/Schedule
    - https://developer.valvesoftware.com/wiki/State
    - https://developer.valvesoftware.com/wiki/NPCThink()
    - https://developer.valvesoftware.com/wiki/Shared_tasks
    - https://developer.valvesoftware.com/wiki/Shared_conditions

Project ideas: AI competition