Collective Behavior in Terasology

GSoC 2019 Proposal by Arthur Casals

What is this about?

- Improving the behavior mechanism to allow collective behaviors
- “Collective” means Actors need to be able to form (and act as) groups
- Behavior trees need to be adapted accordingly

Proposal.get();

- Project Details
  - Overview
  - Goals and Stretch Goals
  - Detailed Work Plan
  - Timeline
- Personal Details
  - Background
  - Why me? (Pick me!)

Project Details

Overview

“*Next in AI: Improvements to Behavior Tree architecture”

This proposal revolves around the existing card on Trello*. Behavior trees are used to describe and implement simple actor behaviors. Each behavior tree uses a combination of tree descriptors (hierarchical nodes - *.behavior) and behavior implementations. Each behavior tree is particular to an individual actor. Nodes can communicate among themselves using each actor’s own blackboard.

Reactive, individual behaviors are OK.

Using an internal hierarchical structure makes sense for individual behaviors. Since the behavior components exist only within the actor, however, it also means that they can only be influenced by (i) environmental factors (that can be perceived by the actor) or (ii) results from previous behaviors. As a result, actors are primarily **reactive** - their behaviors are a consequence of what is happening around them.

Overview

Sometimes, however, we need to play with others.

The existing behavior structure is not appropriate for *collective* behavior among actors. Collective behavior can be illustrated by a pack of jackals: depending of a certain criteria being satisfied (number of jackals in the pack, number of preys/enemies around), they can either decide to attack or to flee - *as a group*. But for that to happen, they need to be *aware* of each other, and to *share* a behavior.

So, how do we make it happen?

Collective behavior resides in two specific pillars: (i) actor groups (where they are aware of each other) and (ii) a behavior structure that can be shared among actors within a group. The simplest way to implement this is to create a *(n+1) coordinated group* - a group formed by n actors and 1 coordinator. In this case, the coordinator is invisible for the rest of the world, acting as a "collective mind" for the other actors.

Goals and Stretched Goals

We can:

- **Create a group structure** able to support grouping of multiple actors.
- **Create a collective mind mechanism** in order to reflect a collective behavior among multiple actors within a group.
- **Improve the existing behavior tree** to make it compatible with collective behaviors.
- **Create a collective behavior example** and add it to the repo
- (Refactor, test, document, make sure the whole thing works)

We wish (in order of importance):

- **Automatic group creation** as a result of existing behaviors (example provided in the detailed work plan section)
- **Multiple groups for each actor** and a prioritized reasoning mechanism for cumulative behavior
- **Create a supergroup structure** in order to make groups discoverable.

Detailed Work Plan (let us detail it here so we don’t pollute the timeline later)

We need to be able to group actors

The basic concept of collective behavior implies a group of entities following a common behavior. For that to happen, the entities (Actor) need to be grouped. While it’s not necessarily true that every entity must be aware of each other, the group should be managed at a certain level so the proper common behavior can be processed (in opposition of having decentralized decision making). That means extending the behavior core and creating a ActorGroup entity (and any necessary auxiliary structures). An ActorGroup should also be able to act on a behavior tree.

Tasks

- Design and implement the ActorGroup structure
- Create a simple mechanism to apply an existing behavior tree to multiple actors within a group

Detailed Work Plan (let us detail it here so we don't pollute the timeline later)

We need to create a collective reasoning mechanism

A collective reasoning mechanism implies reasoning with multiple actor inputs (how each actor is affected by its surrounding environment) and group parameters. Ideally, collective reasoning should support *norms* - rules that may be applied according to the group characteristics (penalties and rewards can be used to enforce norms). The entity structure should reflect the relationships between: one reasoning mechanism (which can be a reusable inference engine), a set of norms, a group of actors, and a shared behavior description (tree).

Tasks

- Design and implement a reusable reasoning engine that can be used as described
- Design and implement a simple norms structure (similarly to .behavior files)
- Design and implement an inference object that can be used with a shared behavior (described next)
- Extend ActorGroup so it can use the new collective reasoning mechanism

Detailed Work Plan (let us detail it here so we don't pollute the timeline later)

We need to change the behavior logic

In order to be collective, it will be necessary to change part of the structure of the behavior core. It will be challenging to make the necessary changes AND to maintain backward compatibility with existing behavior structures. This should be feasible since there would be no changes related to the state transitioning mechanism, and the tree runner/builder associated classes can be extended. While the tree runner mechanism is not ideal, it assumes a 1:1 relationship between actor and behavior. Not changing this implies on building a complementary structure (*CollectiveBehavior*).

Tasks

- Design the new tree structure (it can be adapted for hierarchical tree networks, or we can use a descriptive Kripke structure if necessary)
- Adapt the core mechanism to support collective behavior processing
- Modify the tree runner mechanism if necessary
- Adapt the ActorGroup and the collective reasoning mechanism if necessary
- Extend the BehaviorTreeBuilder class in order to accommodate any changes necessary on the .behavior files

Detailed Work Plan (let us detail it here so we don’t pollute the timeline later)

We need to see it working

Showcasing the new collective behavior structure is important - that is how we can get feedback from the community. The following is proposed after discussing it with @Cervator: the WildAnimalGenome module can be adapted/extended as a showcase. The idea is to apply different collective behaviors according to group size. As a related stretch goal, new herds (groups) can be automatically created once the current herd reaches a certain size.

Tasks

- Analyze the existing WildAnimalGenome model
- Create an appropriate behavior model to showcase the new functionalities
- Extend the first showcase version according to the declared stretch goals (if there’s enough time)

Timeline - Community Bonding

Week (~40h)	**Work plan (predicted)**
May 6 - May 10	● Setting up the Terasology dev environment ● Connecting with mentors ● Playing Terasology ● Reviewing the core behavior mechanism ● Reviewing the code and issues related to Behavior Trees
May 13 - May 17	
May 20 - May 24	

Timeline - Phase 1

Week (~40h)	**Work plan (predicted)**
May 27 - May 31	● Create a model (written design) to accommodate the proposed changes ● Model a progressive scenario to be used in the following implementation, describing expected results (so they can be compared with the obtained results in the final report). (From now on, it is implicit that every activity block will be tested with the appropriate scenario structure)
June 3 - June 7	● Create the actor group structure (ActorGroup) ● Modify/extend any other classes as necessary
June 10 - June 14	
June 17 - June 21	● Create the collective reasoning mechanism ● Extend the ActorGroup entity
June 24 - June 28	

Timeline - Phase 2

Week (~40h)	**Work plan (predicted)**
July 1 - July 5	Design the new behavior tree structure
July 8 - July 12	● Adapt the core mechanism to support collective behavior processing ● Extend the behavior tree runner mechanism
July 15 - July 19	● Adapt the ActorGroup and the collective reasoning mechanism if necessary ● Extend the BehaviorTreeBuilder class in order to accommodate any changes necessary on the .behavior files ● Propagate all modifications across the event processing mechanism
July 22 - July 26	

Timeline - Phase 3

Week (~40h)	**Work plan (predicted)**
July 29 - Aug 2	Working on the GUI to reflect all modifications up to this point
Aug 5 - Aug 9	
Aug 12 - Aug 16	● Creating and adding a collective behavior example to the repo based on the WildAnimalGenome model. ● Additional documenting and finalizing the project ● Next steps (stretched goals)
Aug 19 - Aug 23	
Aug 26 - Aug 30	Final project evaluation

Personal Details

Background and motivation

Hi, my name is Arthur.

I am a third year PhD student at USP (Brazil) and a second year PhD at Sorbonne Université (France) (cool story). Before that, I received a BsC degree in Computer Engineering (2004 - ITA) and a MBA in 2016 (Insper). I've been developing software for a long time now, in different languages, for different purposes. Previous startup founder, tech manager, software engineer, oil drilling engineer.

OK, what else?

I’ve been developing software for a *very* long time now, and I’m absolutely passionate about it. I particularly enjoy learning and applying science to the real world. I’ve been looking for a more practical application of my research in the last couple of months, and for my surprise I found - you! - and a GSoC idea that was perfect for it. As a passionate developer and researcher (and gamer), I will commit to this project with all I got.

Background and motivation

(I also wrote some stuff related to this proposal)

- CASALS, A.; BELBACHIR, A.; SEGHROUCHNI, A.E.F. ; BRANDAO, A. A. F. Gestion d'intentions multiples pour agents ambiants coopératifs. In: Vingt-sixièmes journées francophones sur les systèmes multi-agents, JFSMA 2018, Métabief, France, 10-12 Octobre 2018. JFSMA 2018. Distribution et décentralisation, Cépaduès, Toulouse, pp 127–136, 2018.
- CASALS, A.; BELBACHIR, A.; SEGHROUCHNI, A.E.F. ; BRANDAO, A. A. F. Fostering Agent Cooperation in AmI: A Context-Aware Mechanism for Dealing with Multiple Intentions. In: Del Ser J., Osaba E., Bilbao M., Sanchez-Medina J., Vecchio M., Yang XS. (eds) Intelligent Distributed Computing XII. IDC 2018. Studies in Computational Intelligence, vol 798. Springer, Cham, 2018.
- NEGRONI, O.; OTHMANY, A.; CASALS, A.; SEGHROUCHNI, A.E.F.. Exposing agents as web services in JADE. In: Engineering Multi-Agent Systems. 6th International Workshop, EMAS 2018, Stockholm, Sweden, 2018 (pending publication: Springer, LNAU series).
- MASCARDI, V. et al. Engineering Multi-Agent Systems: State of Affairs and the Road Ahead. In: Engineering Multi-Agent Systems. 6th International Workshop, EMAS 2018, Stockholm, Sweden, 2018 (pending publication: Springer, LNAI series).
- CASALS, A.; FERMÉ, E. ; BRANDÃO, A. A. F. . D*omain-specific Trust for Context-aware BDI Agents - Preliminary Work*. In: 10th International Conference on Agents and Artificial Intelligence, 2018, Funchal. Proceedings of the 10th International Conference on Agents and Artificial Intelligence, 2018.
- CASALS, A.; SEGHROUCHNI, A.E.F. ; BRANDAO, A. A. F. *Augmented Agents: Contextual Perception and Planning for BDI architectures*. In: Engineering Multi-Agent Systems. 5th International Workshop, EMAS 2017, São Paulo, Brazil, May 8-9, 2017, Revised Selected Papers. Springer International Publishing, LNAI 10738, pp. 1–18, 2018.
- CASALS, A.; BRANDAO, A. A. F. . *Modeling a mobile learning context data ontology*. In: 2017 IEEE World Engineering Education Conference (EDUNINE), 2017, Santos. 2017 IEEE World Engineering Education Conference (EDUNINE), 2017. p. 71.