Augustana College Augustana Digital Commons

Celebration of Learning

May 3rd, 12:00 AM - 12:00 AM

Migrating Behavior Search's User Interface from Swing to JavaFX

An Nguyen Dang Augustana College, Rock Island Illinois

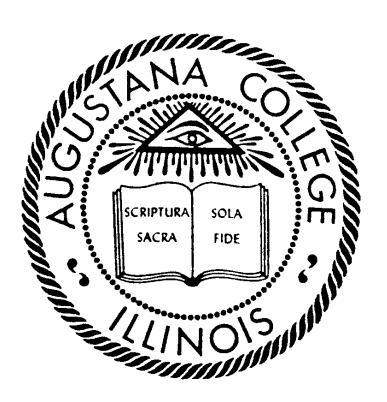
Follow this and additional works at: http://digitalcommons.augustana.edu/celebrationoflearning Part of the <u>Education Commons</u>

Augustana Digital Commons Citation

Nguyen Dang, An. "Migrating Behavior Search's User Interface from Swing to JavaFX" (2017). *Celebration of Learning*. http://digitalcommons.augustana.edu/celebrationoflearning/2017/posters/10

This Poster Presentation is brought to you for free and open access by Augustana Digital Commons. It has been accepted for inclusion in Celebration of Learning by an authorized administrator of Augustana Digital Commons. For more information, please contact digitalcommons@augustana.edu.

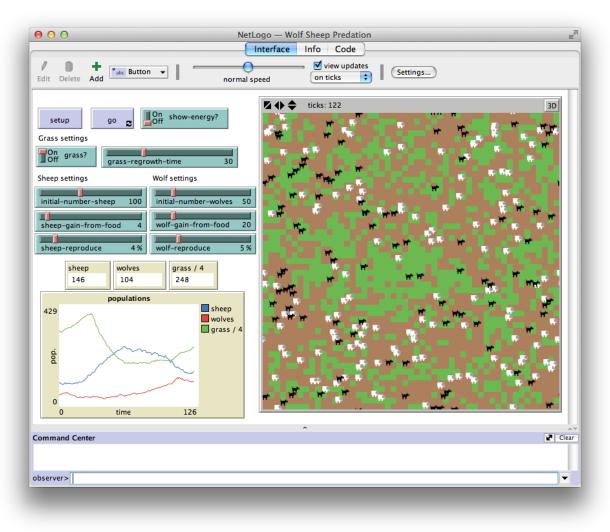




I. Introduction

Agent-Based Models (ABMs) and NetLogo

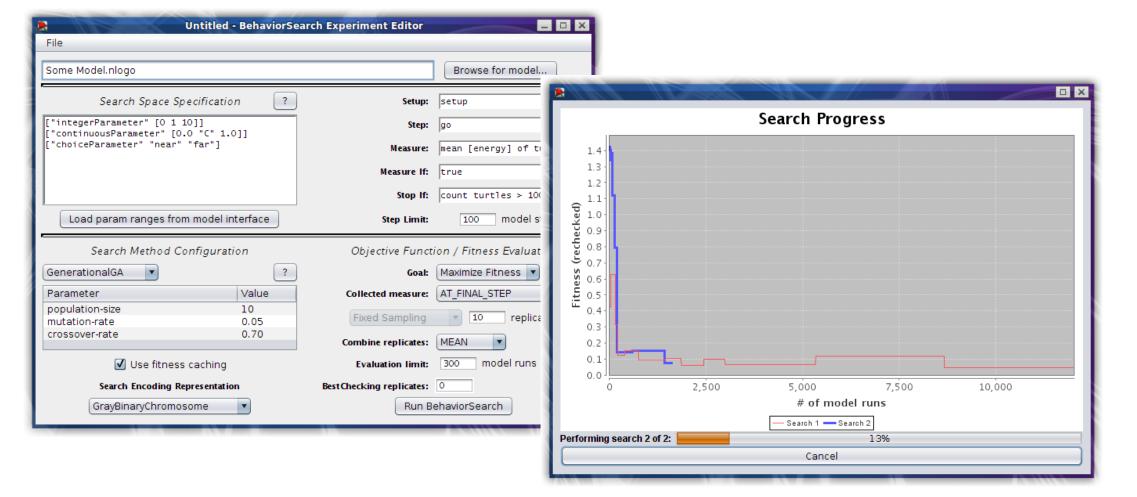
- Agent-based modeling is a computer modeling technique that focuses on modeling the rules of individuals ("agents") and simulating the interactions between these individuals.
- ABMs are widely used to simulate behavior in many fields including archaeology, biology, economics, and social science.
- NetLogo is an agent-based modeling language and integrated modeling environment. It is a popular platform for building and running ABMs. (See: <u>http://ccl.northwestern.edu/netlogo/</u>)



A sample predator-prey model in NetLogo

BehaviorSearch

- BehaviorSearch is a tool to help automate the exploration and analysis of ABMs. (See: <u>http://www.behaviorsearch.org/</u>)
- This software interfaces with the NetLogo platform and allows the client to search for combinations of model parameter settings that will result in a specified target behavior



A screenshot of the BehaviorSearch software before this project

IV. Results (Improved UI with JavaFX)

E Untitled - BehaviorSearch 1.10 — 🗆 🗙				Model Search Objective Search Algorithm		🖹 Run Options Dialog — 🗌		
File Help Some Model.nlogo Browse for model Model Search Objective Search Objective Search Algorithm		Objective / Fitness Function ? Goal: Minimize Fitness			arch Method Configuration	Output file stem: mySearchOutput	Browse	
Parameter Specification	?			Parameter	Value			
["integerParameter" [0 1 10]] ["continuousParameter" [0.0 "C" 1.0]] ["choiceParameter" "near" "far"]		Collected measure:	MEAN_ACROSS_STEPS -	mutation-rate	0.05	Number of searches: 1		
	Setup: setup		▼ 5	population-size	30	Starting at search ID: 1		
	Step: go Measure: mean [energy] of turtles	Fixed Sampling 🔹		crossover-rate	0.7			
	Measure if: true		MEAN	population-model	generational	Initial random seed: 1455164852		
	Stop if: count turtles > 1000	Combine replicates:		tournament-size	3	Number of threads: 4		
	Step limit: 100 model steps	Take derivatives?	Use ABS value?			Brief Output?		
Load parameter ranges from model interface	Run BehaviorSearch		$\Delta = 0.0$	Use fitness caching Search Encoding Representation GrayBinaryChromosomer ?	Evaluation limit: 12000 Best Cheking replicates: 10 model replicates:	runs Start Search	Cancel	

Migrating BehaviorSearch's User Interface from Swing to JavaFX

An Nguyen Dang, and Forrest Stonedahl*

Mathematics and Computer Science Department, Augustana College *Faculty Advisor

II. Motivation

Java Swing Graphical User Interface (GUI)

- Earlier versions of BehaviorSearch used the Swing GUI library
- With Swing, all of the graphical components and controlling methods get embedded in the same code, which makes the code long and hard to debug
- The GUI style looked and felt dated (e.g. like Windows 2000)
- Application didn't scale well to modern high-resolution screens Did not support displaying charts inherently
- (BehaviorSearch used the third-party JFreeChart library for this)
- All of the input forms were packed into one cluttered window

JavaFX Graphical User Interface (GUI)

- JavaFX fixes many of the problems with Swing
- It separates the graphical component details into a separate .fxml file that is easy to read and change
- It also provides SceneBuilder, which allows developers to easily modify .fxml with real-time preview using drag and drop
- Programs can be styled with CSS3, allowing easy changes of style
- Supports chart components (no third party library required)
- Organizes input forms into tabs for a less cluttered display

鷵 Un	ntitled -	BehaviorSearch 1.1	0						\times
File	Help								
S	ome Mo	odel.nlogo					Browse	e for mode	
1	Model Search Objective Search Algor			📡 Untitled - BehaviorSearch 1.10			_		×
		Param	eter Specifi	File Help					
6	· ·			Some Model.nlogo	Browse for model				
	["integerParameter" [0 1 10]] ["continuousParameter" [0.0 "C" 1.0]]		Parameter Specification	?	Setup:	setup			
	["choiceParameter" "near" "far"]			["integerParameter" [0 1 10]] ["continuousParameter" [0.0 "C" 1.0]] ["choiceParameter" "near" "far"]		Step:	-		
							Measure: mean [energy] of turt1 Measure If: true		
						Stop If:	count turtles	> 1000	
				Load param ranges from model in Step Lim			t: 100 model steps		
			Search Method Configuration Objective			e / Fitness Function ?			
			StandardGA ~	?		Maximize Fitn			
	Load parameter ranges from			Parameter Value mutation-rate 0.03	Collected measure: AT_FINAL_STEP ~				
				population-size 50 crossover-rate 0.7	Combine replicates: MEAN				
				population-model generation tournament-size 3	al	Take derivative?	Use	ABS value?	
						w.r.t		Δ= 0.0	
			Use fitness caching	Evaluation limit: 300 model runs					
			Search Encoding Representation		BestChecking replicates:				
				GrayBinaryChromosome ~	?		Run	BehaviorSea	arch

Comparison of before and after new GUI (Java FX is bigger)*

*screenshot taken on 2160 x 1440 HD monitor

This software development project was completed over the course of 12 months. The final GUI was composed of 440 lines of Java code, and 163 lines of Java comments, spread across 9 source files.

III. Challenges

Multithreading in JavaFX

- When dealing with time-consuming computational tasks, like what BehaviorSearch does to analyze models, it is important to do those tasks in a parallel worker thread, so that the GUI stays responsive.
- Java Swing had a specialized solution for this, a SwingWorker class does all the computation in the background
- JavaFX doesn't have similar class to SwingWorker

Solution: Create the new thread manually, create a task worker that implements the Runnable interface, and then use Platform.runLater() to update the GUI.

Utilizing Multiple Scenes

• Scenes in JavaFX are not easy to get access to, since they are created automatically by the FXMLLoader. This makes it hard to pass data from one scene to another scene.

Solution: Get the Controller object from the FXMLLoader, then pass parameters into the controller using an ini method.

Example:

RunOptionDialogController runController; runController = loader.getController(); runController.ini(runOptions, this);

Table Functionality in JavaFX

• The default table in JavaFX required users to hit ENTER to commit each value they entered. This was not a natural behavior for the table in this software and could confuse users.

Solution: Design a custom subclass of the TableCell class that is named AcceptOnExitTableCell (class outline shown at right) which overrides the default table behavior such that user's entries are stored regardless of whether they hit ENTER.





