

Milestone 2 - Java O-O Program Visualizer/Animator

Team Members:

Prottasha Deb pdeb2024@my.fit.edu
Angelina Martis amartis2023@my.fit.edu
Arya Patel arya2023@my.fit.edu
Aiden Stensrud astensrud2024@my.fit.edu

Faculty Advisor:

David Luginbuhl dluginbuhl@fit.edu

Client:

David Luginbuhl dluginbuhl@fit.edu

Progress of current Milestone:

Task	Completion %	Arya	Aiden	Prottasha	Angelina	To do:
Implement, test, and demo the Java code parser	100%	-	100%	-	-	None
Implement, test, and demo simple memory diagram generator	100%	20%	-	-	80%	None
Conceptualize, draft, and demo memory diagram animations	100%	-	-	100%	-	None
Resolve technical challenge	50%	80%	-	-	20%	None

Discussion of each accomplished task for the current Milestone:

- Implement, test, and demo the Java code parser: A parser for the language subset we are using has been developed. It performs syntax and lexical analysis, discovering errors such as invalid program statements. It also converts the given program into an Abstract Syntax Tree, on which type checking is performed to discover semantic errors such as type mismatches. The AST can be printed in a text display for demonstration purposes.
- Implement, test, and demo simple memory diagram generator: A memory-diagram generator demo has been developed using Python and Matplotlib to print out a current state based on what the Java parser would hypothetically output. Taking that output, the generator reads each line of “instruction” and outputs the state for each line that gets executed.
- Conceptualize, draft, and demo memory-diagram animations: A prototype memory-diagram animator has been developed using Python and Tkinter. It visually represents Java objects, variables, and references in memory. The animator supports step-by-step execution of Java statements, displaying how objects are created, variables are assigned, and object fields are updated. Animated arrows are used to show references between variables and objects. The prototype also includes code highlighting, automatic playback, and reset functionality to demonstrate memory changes during program execution.
- Resolve technical challenge: The generator demo currently only reads primitive/reference variables and objects along with the fields associated with each object. Methods and string objects have yet to be implemented, so this technical challenge still needs to be resolved.

Discussion of contribution of each team member to the current Milestone:

- Prottasha Deb: Prottasha is responsible for the memory diagram animation portion of the project. She developed the current prototype using Python and Tkinter, implementing visual representations of Java objects, variables, and references. She added step-by-step execution, animated reference arrows, code highlighting, and playback controls to demonstrate how Java statements affect memory.
- Angelina Martis: Angelina is responsible for developing the majority of the memory-diagram generator, mainly focusing on getting the states to print out accurately for different types of parser output.
- Arya Patel: Arya is responsible for assisting Angelina in developing the memory-diagram generator demo(syntax and formatting) and making different parser output text files to use for testing purposes.
- Aiden Stensrud: Aiden is responsible for the Java parsing part of the project. Thus, he solely developed the current features of the Java parser, implementing the syntax and lexical analyzers, the AST converter, and the type checker. He also wrote the presentation slides for his completed milestone task.

Plan for the next Milestone:

Task	Arya	Aiden	Prottasha	Angelina
Implement, test, and demo the converter from parsed Java code to memory state representations	-	100%	-	-
Implement, test, and demo memory diagram animator	-	-	20%	80%
Resolve technical challenge	-	-	Accurate animations for complex Java programs	
Implement, test, and demo GUI	70%	10%	-	20%

Discussion of each planned task for the next Milestone:

- Implement, test, and demo the converter from parsed Java code to memory-state representations: Now that the Java code is parsed into an AST, its runtime can be simulated. This simulation needs to be implemented, converting the parsed code into a list of states. These states will be generated in the same structure that the memory diagram generator takes as input, so that Java code can be converted into memory diagrams.
- Implement, test, and demo memory diagram animator: Now that a demo for the animation functionality has been approved, it must be implemented fully. The animator will take an ordered list of memory diagrams and “animate” the transitions between them.
- Resolve technical challenge: Producing accurate animations, even for complex Java programs, was a foreseen issue that must be addressed as we develop the memory diagram animator. A clean solution must be found in order to implement the memory diagram animator.
- Implement, test, and demo the GUI: When completed, the above tasks will produce a coherent application that meets the project's requirements. Thus, encapsulating said application in a GUI will produce a full, demonstrable application that can be improved upon in the second semester's milestones.

Date of meeting with Client during the current milestone:

Prattasha and Aiden met with the client on Wednesday, October 21st, to demo the Java code parser and memory diagram animator. Arya and Angelina met with the client on Friday, October 23rd, to demo the memory diagram generator.

Client feedback on the current milestone:

Faculty Advisor feedback below.

Dates of meeting with Faculty Advisor during the current milestone:

Prattasha and Aiden met with the advisor on Wednesday, October 21st, to discuss the Java code parser and memory diagram animator. Arya and Angelina met with the advisor on Friday, October 23rd, to discuss the memory diagram generator, overall milestone progress, and plans for the next milestone.

Faculty Advisor feedback on each task for the current Milestone:

- Implement, test, and demo the Java code parser:
- Implement, test, and demo simple memory diagram generator:
- Conceptualize, draft, and demo memory diagram animations:
- Resolve technical challenges:

Faculty Advisor Signature: _____ Date: _____

Evaluation by Faculty Advisor

Faculty Advisor: detach and return this page to Dr. Chan (HC 209) or email the scores to pkc@cs.fit.edu

Score (0-10) for each member: circle a score (or circle two adjacent scores for .25 or write down a real number between 0 and 10)

Prottasha Deb	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Angelina Martis	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Arya Patel	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Aiden Stensrud	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10

Facutly Advisor Signature: _____ Date: _____