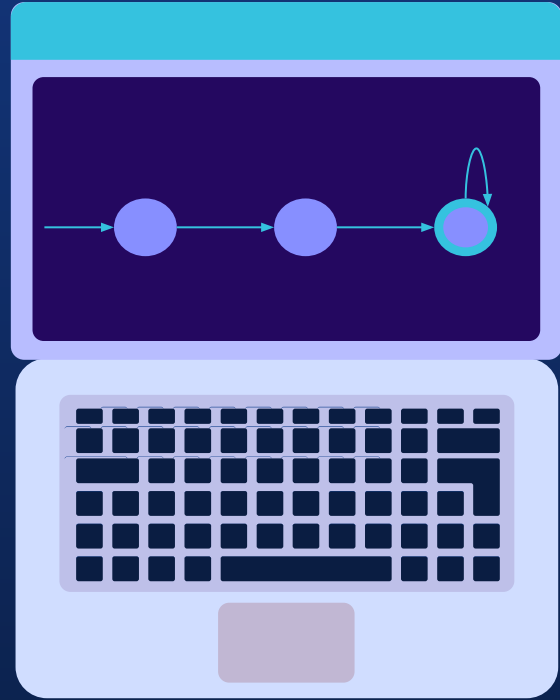


# Improved Visualization for Formal Language: Milestone 5

<https://kmcnear2022.github.io/>

**Group Members:** Chris Pinto-Font, Vincent Borrelli, Andrew Bastien, Keegan McNear





## Milestone 5 Goals

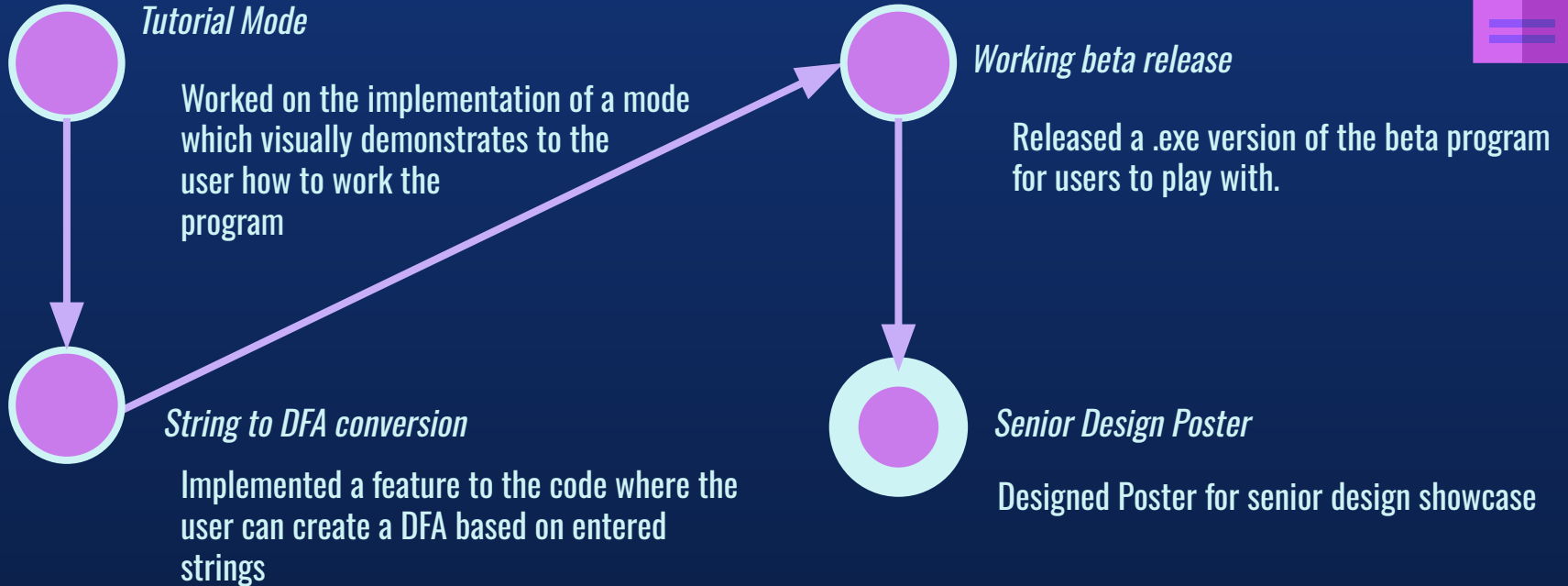
*We set out to improve the navigability of the program through reorganizing the GUI to accommodate new features. We too set out to implement requested features and foundational program components.*

*The completion of our code base refactoring assisted in this.*



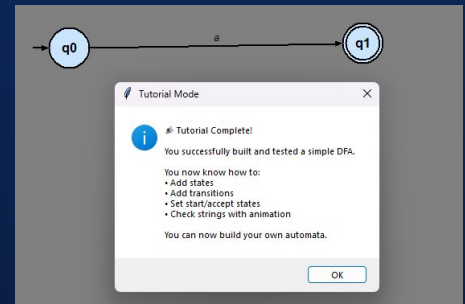
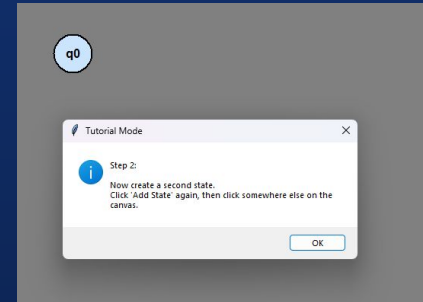
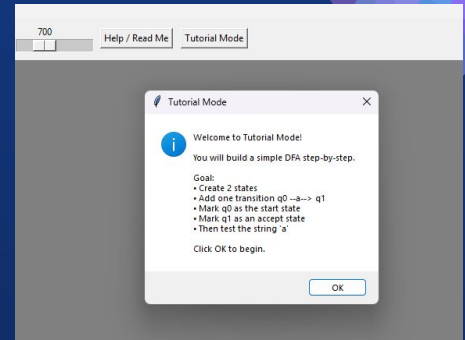


# Milestone Four Deliverables



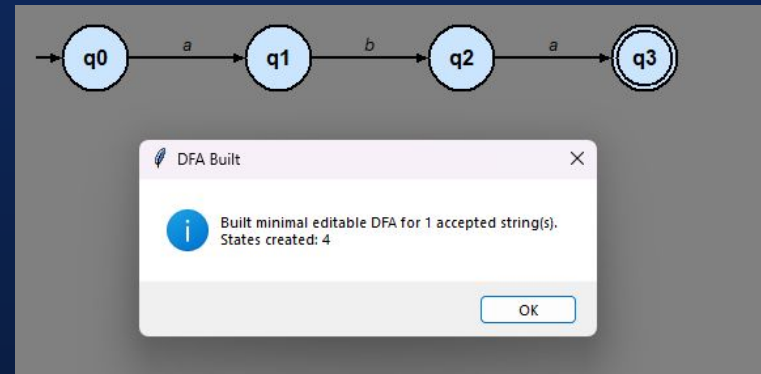
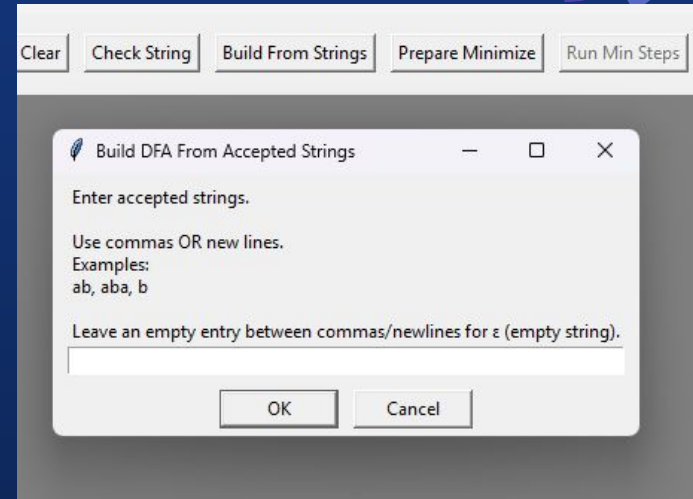
# Tutorial Mode

Serves to showcase the most basic DFA construction elements of the program as to take the guesswork out of using the program for more inexperienced users, may be expanded to showcase more features however the scope of the read-me file and anticipated user background knowledge for other related DFA features makes the necessity of each expansion dubious.



# Building DFAs from strings




Feature allows the user to build a functional DFA based on entered strings, needs to be stress tested for complex strings and many items.





## Public Release Version for user input

Released a separate independently executable beta release version of the program to gauge wider user feedback based on requests for a working version of the grapher. We hope to use this to outsource the bug testing and feature requests process via this, with a possible questionnaire being attached.

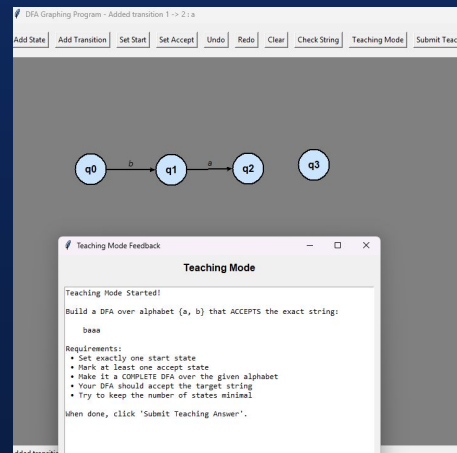
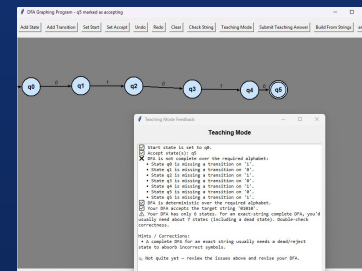
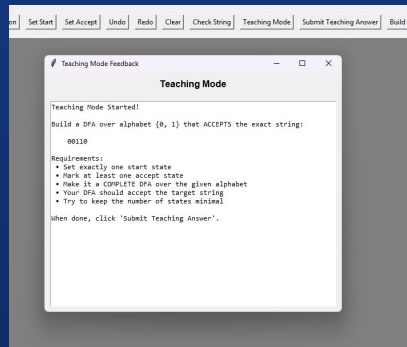
|  |  |
|--|--|
|  UpdateVersGDFA.exe | Add files via upload                   |
|  legacy.py          | Fixing transition popup on tkinter gui |
|  legacyGDFA.exe    | Add files via upload                   |





# Teaching Mode Beta

Worked on implementing a basic version of the much requested “teaching mode” to instruct the user on pragmatic DFA construction via prompts. This will be expanded upon and finalized for milestone 6, with a more interactive take on the construction process (real time comments on user decisions as opposed to a construct and check method)





# Live Demos

The screenshot shows a software window titled "DFA Graphing Program". The menu bar includes the following items: Add State, Add Transition, Set Start, Set Accept, Undo, Redo, Clear, Check String, Teaching Mode, Submit Teaching Answer, Build From Strings, Prepare Minimize, Run Min Steps, Play, Pause, Step, -- Step, Speed(ms): 700, Help / Read Me, and Tutorial Mode. The main area of the window is a large, empty gray rectangle. At the bottom left of the window, the text "Ready" is visible.

A Windows notification from GIGAT Internet Security. The title is "Microphone Monitor". The notification contains the following text: "Shipping Tool (1124) is accessing your microphone." Below this, there is a link "Do not notify for this application." and the text "End process accessing the microphone." At the bottom, there is a link "Learn more about this message" and a red bar with the text "Renew subscription" and "Your computer is not protected".





# Live Demos

DFA Graphing Program

Add State Add Transition Set Start Set Accept Undo Redo Clear Check String Teaching Mode Submit Teaching Answer Build From Strings Prepare Minimize Run Mini Steps Play Pause Step -- Step + Speed(ms): 700 Help / Read Me

Ready



## DFA Visualizer

Chris Pinto-Font, Vincent Borrelli, Andrew Bastien, Keegan McNear
   
 Faculty Advisor(s): Dr. David Luginbuhl, Dept. of Computer Science, Florida Institute of Technology

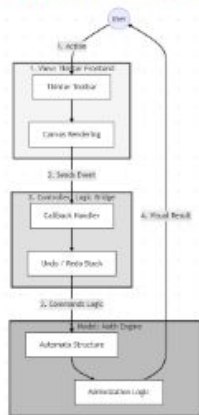
### Motivation/Goal

- Our project is important for students and professors because DFA Visualizer helps to understand Deterministic Finite Automata (DFA). This system helps visualize and teach how a DFA is created and expressed.
- The canvas space is used more like a dynamic space using interactive manipulation and animation.
- Student comprehension of complex DFA concepts is the primary goal.

### Approach/Features

- Interactive workspace with drag and drop state manipulation
  - Point and click transition drawing
  - Real time string traversal animation
- Teaching mode: reverse engineer DFAs from string input
  - Minimization of DFA
- Traverse states given a string input and accepts or rejects based on DFA and string input
  - Minimization of DFA
- If NFA is made instead, can be converted into DFA
  - String input to create matching DFA

### System Architecture



### Evaluation Results

- Accuracy: Successfully converted NFAs to DFAs. Ability to fully minimize. Correct management of  $\lambda$  transitions.
- Performance: GUI responsiveness. Memory efficient.  $O(1)$  time complexity for UIID.
- Stability: Crash free. Architecture prevents GUI errors. Maintains machines without memory collision.

### Future Works

#### Limitations:

- Desne graphs become visually cluttered
- Limited to finite automata
- Static canvas size

#### Possible Improvements:

- Implement a layout to have better visuals
- Export feature
- Upgrade to support Turing machines

### Visualizer Examples

The examples show several DFA diagrams with states and transitions. One diagram shows a DFA with states q0, q1, q2, q3, q4, q5 and transitions labeled 'a' and 'b'. Another diagram shows a DFA with states q0, q1, q2 and transitions labeled 'a' and 'b'. A screenshot of the software interface shows a DFA diagram with a string input field containing '100' and a 'Process String' button. A code editor window shows the following code:

```

    # DFA Visualizer - User Guide
    1. Viewing States:
    - Click 'Add State' this click appears on the canvas to place a new state.
    2. Adding Transitions:
    - Click 'Use Transitions' when a state exists, then an edge exists.
    - You'll be prompted to enter transition details. Use canvas to see available (e.g. 'a').
    - Add 2 states and separate.
    3. Mark & Accept States:
    - Click 'Set Start' and then a state to designate it as the start state.
    - Click 'Set Accept' and then a state to designate it as acceptable.
    4. Minimize Automata:
    - Click a string and 'DFA Minimize' to prepare an optimized DFA.
    - Use 'Eg' to minimize the transitions. Based on string, 'Step 1' to adjust the transitions, and 'Step 2' to stop back.
    - Add 'Optimize' to change the previous DFA (Lower - better).
    5. Visualization:
    - Click 'DFA Minimize' to prepare and run the minimized DFA.
    - The string will be added to the state and then transitioned to the next state.
    6. Editing & Settings:
    - 'Add' and 'Delete' allow users to control the DFA's structure.
    - 'Clear' removes everything and starts over.
  
```

### Implementation/Tools



# Milestone 5 Progress



| Task   | Completion % | Chris | Vincent | Andrew | Keegan | To Do  |
|--|--------------|-------|---------|--------|--------|--|
| NFA dead state and bug fixing                    | 80%          | 30%   | 30%     | 20%    | 20%    | Add dead state creation  |
| Canvas GUI upgrade with quality of life features | 100%         | 30%   | 15%     | 40%    | 15%    | Make further improvements as they arise and continue to rearrange buttons and GUI features |
| Tutorial mode                                    | 80%          | 25%   | 25%     | 15%    | 15%    | Continue to bug test and stress test current system  |
| String input converted into a DFA                | 80%          | 30%   | 30%     | 10%    | 10%    | Continue to bug test and stress test current system  |
| User feedback                                    | 80%          | 40%   | 20%     | 20%    | 20%    | Continue to collect user input/judge input based on .exe release package                   |
| Update Read Me File                              | 100%         | 30%   | 30%     | 20%    | 20%    | Continue making updates as they arise  |
|  |              |       |         |        |        |  |

# Milestone 6 Plan

| <b>Task</b>                                      | <b>Chris</b>                              | <b>Vincent</b>                            | <b>Andrew</b>                       | <b>Keegan</b>                       |
|--|---|---|-------------------------------------|-------------------------------------|
| Finalized Touches on user experience/bug testing | Bug Fixer and Project stress tester       | Bug Fixer and Project stress tester       | Bug Fixer and Project stress tester | Bug Fixer and Project stress tester |
| Zoom out/item selection and moving feature       | Bug Fixer/Code Contributor and researcher | Bug Fixer/Code Contributor and researcher | Co-Lead coder and development head  | Co-Lead coder and development head  |
| Teaching mode                                    | Bug Fixer/Code Contributor and researcher | Bug Fixer/Code Contributor and researcher | Co-Lead coder and development head  | Co-Lead coder and development head  |
| Update Poster as new features are added          | Co-lead Writer                            | Co-lead Writer                            | Advisor                             | Advisor                             |
| Update “read me” file                            | Co-lead Writer                            | Co-lead Writer                            | Advisor                             | Advisor                             |



# *Questions?*

*Visit Our Site*

<https://kmcnear2022.github.io/>

