CprE 492 Bi-Weekly Report 01

2/28/2021 - 3/14/2021

Group #: 50

Project Title: Cy-Sec Game

Advisor: Manimaran Govindarasu

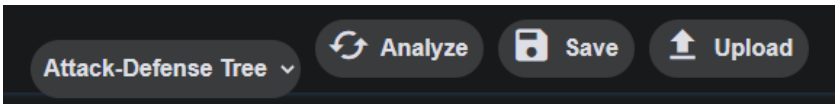
Front End Members: Jon Greazel, Hayden Sellars, Joseph Strobel

Back End Members: Harrison Majerus, Nicholas Battani, Stefan Peng

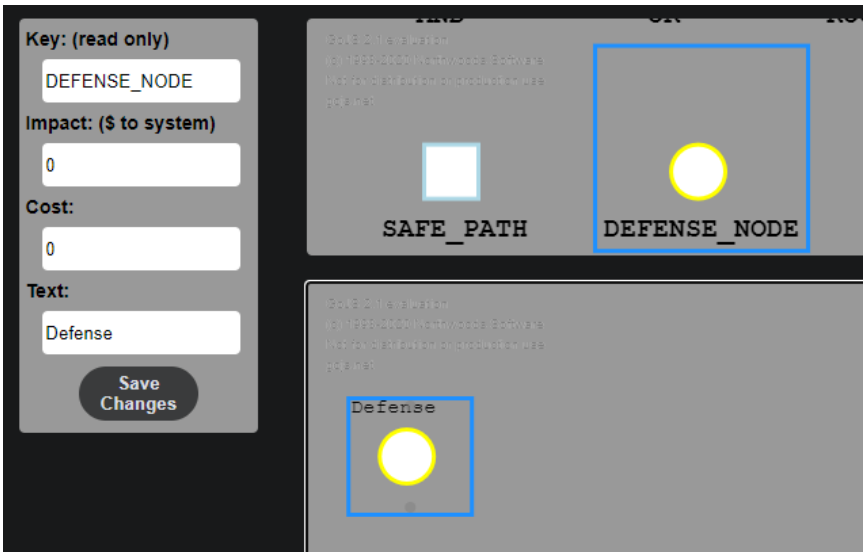
---

Bi-Weekly Summary: For the past weeks our team has been focusing on nailing down details for each computation engine. We've spent a lot of time with our advisor deciding which inputs and attributes best suit each version of the application. Our back end team is moving forward with all three analytical algorithms while the front end is working to implement the correct inputs per engine.

Past 2 Weeks Accomplishments:

- Jon:
    - Worked more on the inspector: Some of our inputs were incorrect so I changed the values on the inspector. Our root node also needed its own value, so I created a node on the palette for the root.
    - Worked on engine selector: The user will be able to select which engine they want to use, so I made a select input that will eventually set a global variable and modify the palette and inspectors accordingly.
        - 

- Hayden:
    - Added animation triggers onto the nodes for front end visualization that highlights vulnerable defense nodes (right now random)

- Joe:
    - Added the defense node with associated inputs for the backend. Added the inspector for clicking on the defense node. Discussed best practice for the engine selector.
        - 

- Harry:
    - Changes to attack-tree algorithm based on new specifications. Changed algorithm output for easier frontend processing. Started attack-defense algorithm implementation and researched ways to go about processing all possible paths for maximum investment efficiency.
- Nick:
    - Added functionality for engines to be access by front-end via the API. Helped integrate the engine functionality to the API.
- Stefan:
    - Started work on the game theory engine. Using the computed attack and defense scenarios, the payoff for the attacker for each combination of scenarios is calculated as $U_A = C_D + I - C_A$, where C_D is the cost of the defense scenario, I is the impact of the breach, and C_A is the cost of the attack scenario. Since this is a zero-sum game, the payoff for the defender is negative of the attacker's payoff, or $-U_A$.
    - Sample output: first section shows the attack and defense scenarios, first array shown is the payoff matrix, second matrix is the Nash equilibrium

for the system.

```
prob: 0.0101    cost: 320.0     weighted cost: 3.2327   : LEAF LEAF3
prob: 0.0048    cost: 109.0     weighted cost: 0.52     : LEAF LEAF4
prob: 0.0762    cost: 242.0     weighted cost: 18.4463  : LEAF2 LEAF3
prob: 0.036     cost: 31.0      weighted cost: 1.1158   : LEAF2 LEAF4
prob: 1.0       cost: 93.0      weighted cost: 93.0     : DEFENSE1 DEFENSE2
prob: 1.0       cost: 77.0      weighted cost: 77.0     : DEFENSE3 DEFENSE4
[[99.7673469387755, 83.7673469387755], [102.48002551020409, 86.48002551020409], [84.5
5367346938775, 68.55367346938775], [101.88415816326531, 85.884158163265311]]
[(array([0., 1., 0., 0.]), array([0., 1.]))]
```

Pending Issues: After our advisor meetings we realized there is a lot of fine tuning that needs to be done before our application is ready to be user tested. We did a good job of finalizing the requirements of our application with our client this past week, but we are going to make a presentation for our client running through all specific inputs and outputs on the frontend and the backend to make sure we are not wasting time on confused requirements.

Individual Contributions:

Team Member	Contribution	Weekly Hours	Total Hours
Jon Greazel	Worked on the inspector and engine selector	6/6.5	31.5
Hayden Sellars	Continue fixing the animation triggers	6/6	30
Joe Strobel	Added defense node and defense node inspector	6/6	31
Harry Majerus	Making changes to functionality and output of the attack tree algorithm. Research into and starting implementation of attack-defense tree processing.	6/7	32
Nicholas Battani	More API work, ensuring frontend/backend communication is sending the correct data across.	6/6	30
Stefan Peng	Worked on the game theory engine	6/6	30

Plans for the Upcoming 2 Weeks:

- Jon:

    - Continue to work on the analytical engine selector. I don't have a lot of experience with Angular so figuring out the best way to conditionally render that many components will be my main challenge.
- Hayden:
    - Finalize animations for the front end graph and work on fine tuning the inputs.
- Joe:
    - Check node input requirements now that we have them documented from the client. Begin working on output visualization for the json received from the backend.
- Harry:
    - Write attack-defense tree algorithm to client specifications and maybe then to maximum investment efficiency if time allows.
- Nick:
    - Add ability to switch between engines from Frontend via the API. Ensure that all engines are returning correct and readable values to the frontend.
- Stefan:
    - Continue refining game theory engine and integrating with the front end
    - Help work on attack-defense tree engine

Summary of Weekly Advisor Meetings: The focus of our advisor meetings was mainly on the correct implementation of back end engines. There are a few different approaches with varying complexities so we discussed the best approach moving forward. We also spent some time hammering out which input should belong to each engine; that information will determine how the inspector looks per node per engine and which parameters the back engine requires.