InSecurity: A Strategy Game on Infrastructure Security

David Feldheim, Krisha Paula Olanday, Michael Wu, Vinayaditya Shivakumar, and Kunal Patel
{daf163,kmo88, vs338,kp451 }@scarletmail.rutgers.edu
Advisors: Prof. Predrag Spasojevic, Rick Anderson, and Prof. Melike Baykal-Gursoy

Goal

- ❑ Develop a standard two-player game for Prof. Baykal-Gursoy's research on Infrastructure Security and Game Theory.
- ❑ Required data collection of each players decisions (both movement and actions) for analysis.
- ❑ Improve the previous game prototype by making it more realistic but still be able to capture the essential requirements for the research. Afterwards build new game scenarios using the standard game.

Motivations and Objectives

- ❑ Motivations
  - Develop a fun, fair, and realistic game that people would want to play.
  - Create various game scenarios to simulate different attacker and defender strategies.
- ❑ Objectives
  - Move away from the prototype's shapes as objective points for players to attack/defend. Replace the shapes with a tiled map and redefine a node as an actual room or area.
  - Remove the feel of randomness from the prototype.
  - Obtain turn data that enables us to replay and analyze a game turn-by-turn.

Research Challenges

- ❑ Defining a basic game that complies with the research requirements
- ❑ Working with new and unknown frameworks for game development
- ❑ Database Connection

References

[1] http://www.piskelapp.com/
[2] http://coursescript.com/notes/interactivecomputing/objects/index.html
[3] http://buildnewgames.com/real-time-multiplayer/

Acknowledgement

We would like to thank our advisors for their time and contributions for our project.

Methodology

Frameworks and Methologies

- ❑ Node.js
- ❑ P5.js
- ❑ Express.js
- ❑ Bootstrap
- ❑ Socket.io
- ❑ MongoDB

Game Mechanics

- ❑ Two player search and capture game. The player cannot see the other.
- ❑ Each player maneuvers the map.
- ❑ Attacker (player) objective: collect all keys located in the map without being caught
- ❑ Defender (player) objective: protect the keys and apprehend the Attacker.

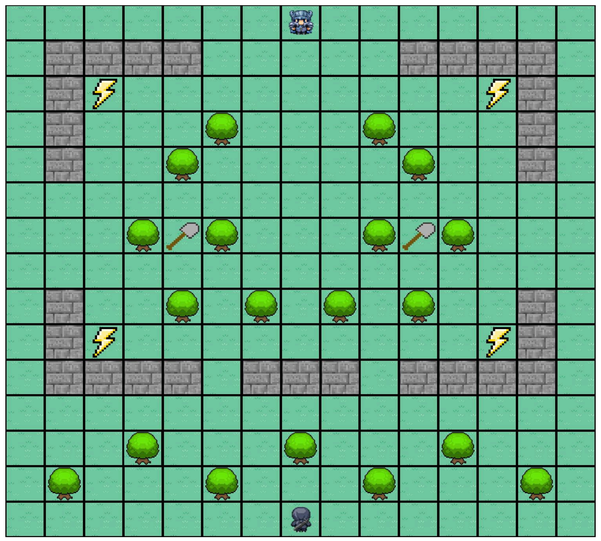
Turn Logic

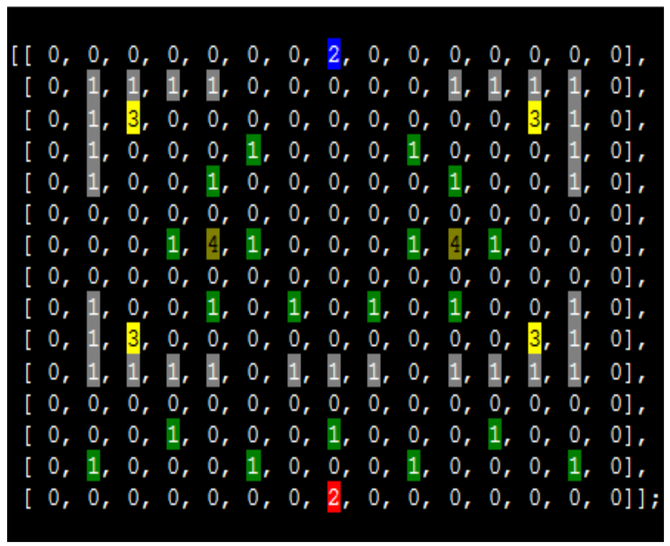
- ❑ Players make their move: move, attack (for attacker), or do nothing. This is done simultaneously.
- ❑ NPC (non-player characters) and game objects change, update, or move (made optional in game settings)
- ❑ All of these changes are captured and sent to the database

Data Collection using MongoDB

- ❑Database stores: map arrays, personalized game settings, and turn data.
- ❑ A variety of game maps are stored in the database.
  - ❑ Each game instance only stores its own map not all maps.
- ❑ Data is sent and retrieved using http/ajax calls

Game Visuals





Handling Large Maps