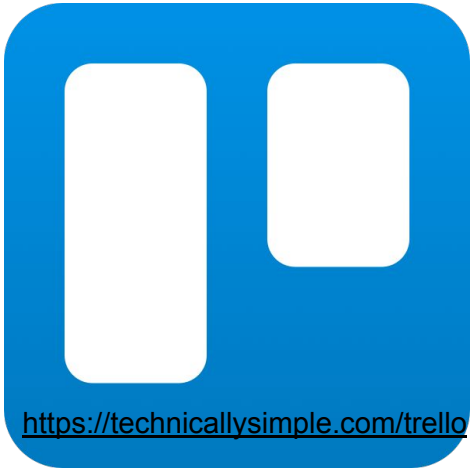
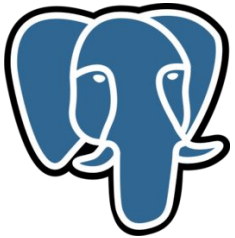
Maze Runner

Group 101-3

Mehdi Ahmadi, Naomi Brown, Ben Jacobs, Corey Lam, Daniel Levin, Jeff Rael

Tools

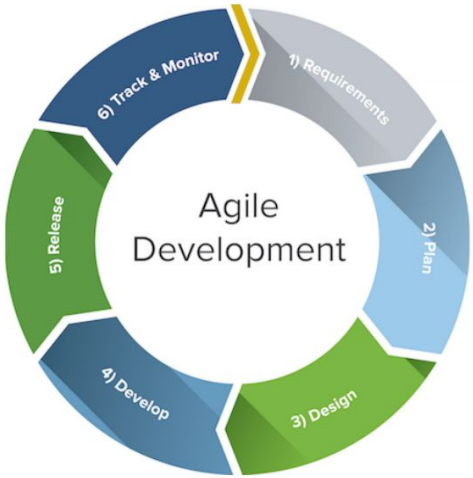


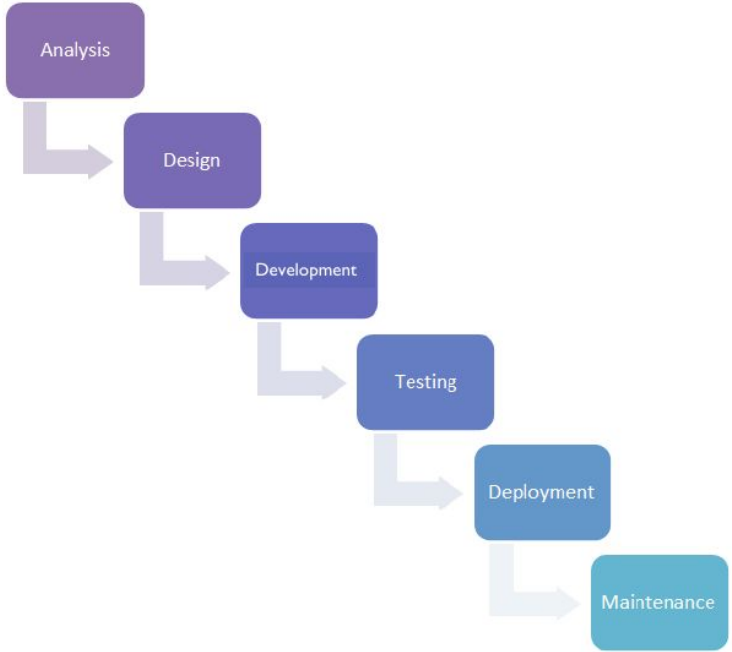


https://www.postgresql.org/

Local Host

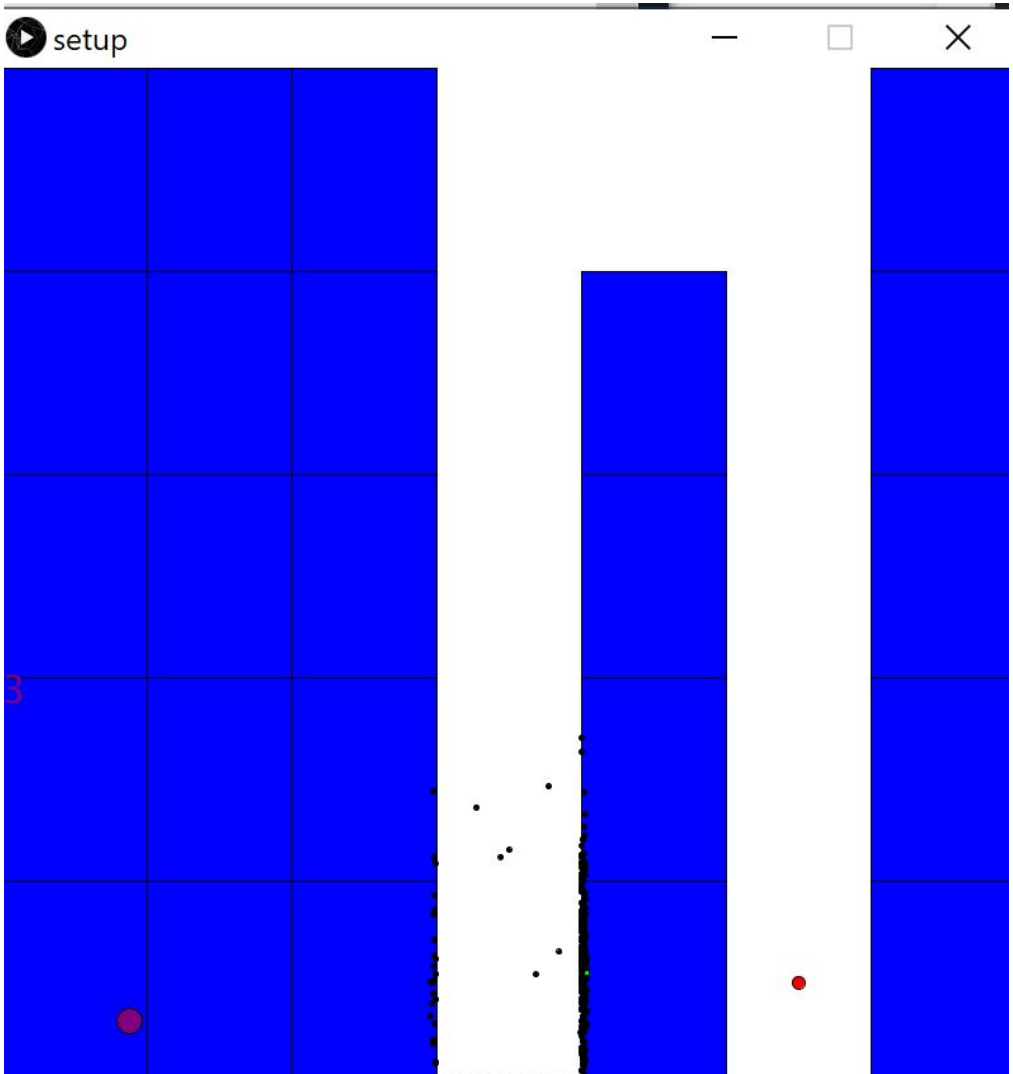
Methodologies





Challenges

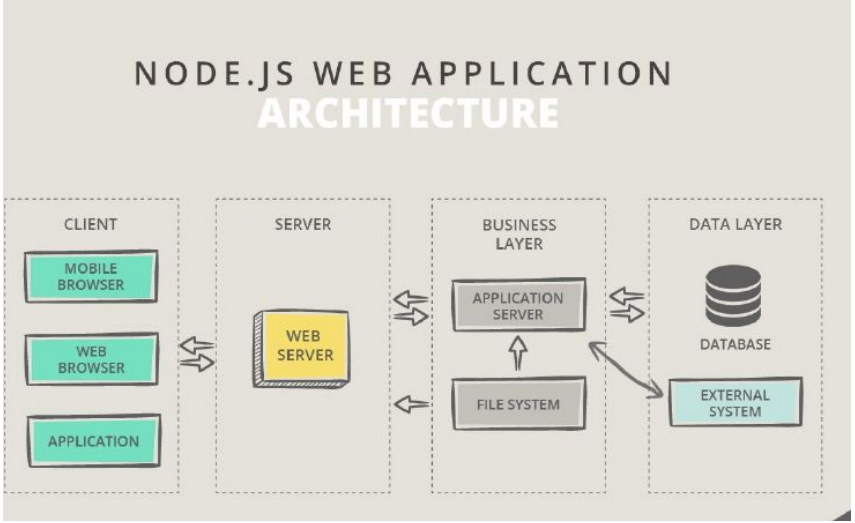
- Processing pde doesn’t allow full functionality
- Original AI algorithm based off distance (ran into walls)
- For player control, had a wonky collision area
- A* Algorithm wasn’t working for bigger mazes as it gives too many infinite directions and will run out from the array.



Integration Challenges

Front-End to Middle-Layer to Back-End

- Code was changed to fit our deployment method
- Sending data



Changes Along the Way

As with any project, we realize some features were too challenging or couldn't be fit in the final result

Some code not used

-Different states, lives/ penalties

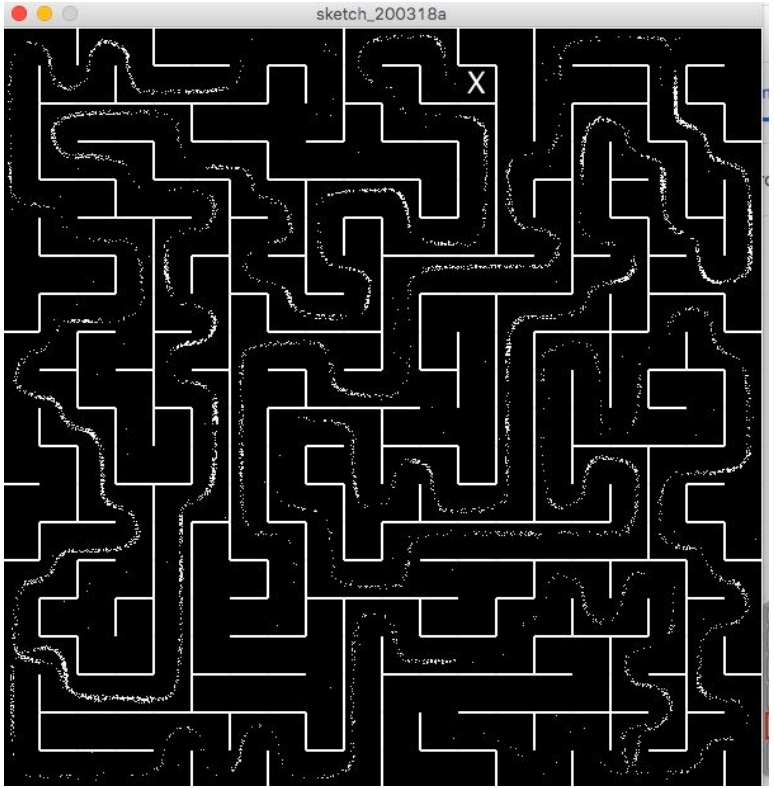
Multiple AI algorithm changes

```
void draw(){
   background(255);

   switch (runState) {
       case title:
         titleScreen();
         runState = RunStates.title;
         init();
         break;
       case running:
         drawingThings();
         break;
       case win:
         winScreen();
         break;
       case lose:
         loseScreen();
         break;
       default:
         print("Default");
   }
  print(runState + "\n");
}
```

A* Algorithm Challenges

- A* Algorithm wasn't working for bigger mazes as it gives too many infinite directions and will run out from the array.
- Realized that an A* algorithm can't be successfully function with the genetic algorithm of the maze (through multiple attempts)
    - Instead we tried to find another visualization on how the A* algorithm works via flowchart



Database - PostgreSQL

```
mazedb2=# select * from players;
    username      |  password
------------------+------------
 Bill T Raul      | tents234
 Ted L Rigs       | oblong890
 Ned R Stevenson  | coolguy545
 Brian O Flats    | pepper323
 Paul Z Waters    | whale454
(5 rows)
```

- 2 Databases
- Players: Username & Passwords
- Maze_Score: Rank, Username, Scores, Times

```
mazedb2=# select * from maze_score;
 rank |    username     | total_score | total_time | score_hard | score_medium | score_easy | time_hard | time_medium | time_easy
------+-----------------+-------------+------------+------------+--------------+------------+-----------+-------------+-----------
    1 | Bill T Raul     |       14420 |        300 |       4890 |         4800 |       4730 |       110 |         100 |        90
    2 | Paul Z Waters   |       14372 |        324 |       4882 |         4784 |       4706 |       118 |         108 |        98
    3 | Ted L Rigs      |       14240 |        390 |       4860 |         4740 |       4640 |       140 |         130 |       120
    4 | Brian O Flats   |       13880 |        570 |       4800 |         4620 |       4460 |       200 |         190 |       180
    5 | Ned R Stevenson |       13760 |        630 |       4780 |         4580 |       4400 |       220 |         210 |       200
(5 rows)
```

Database - Example Queries

```
mazedb2=# select avg (total_time) as avg_time from maze_score;
       avg_time
----------------------
 442.8000000000000000
(1 row)

mazedb2=# select avg (total_score) as avg_score from maze_score;
     avg_score
--------------------
 14134.400000000000
(1 row)
```

rank	username	total_score	total_time	score_hard	score_medium	score_easy	time_hard	time_medium	time_easy
1	Bill T Raul	14420	300	4890	4800	4730	110	100	90
2	Paul Z Waters	14372	324	4882	4784	4706	118	108	98
3	Ted L Rigs	14240	390	4860	4740	4640	140	130	120

(3 rows)

```
mazedb2=# select time_hard, (5000 - time_hard*1) as Score_Hard from maze_score;
 time_hard | score_hard
-----------+------------
       110 |       4890
       118 |       4882
       140 |       4860
       200 |       4800
       220 |       4780
(5 rows)
```