

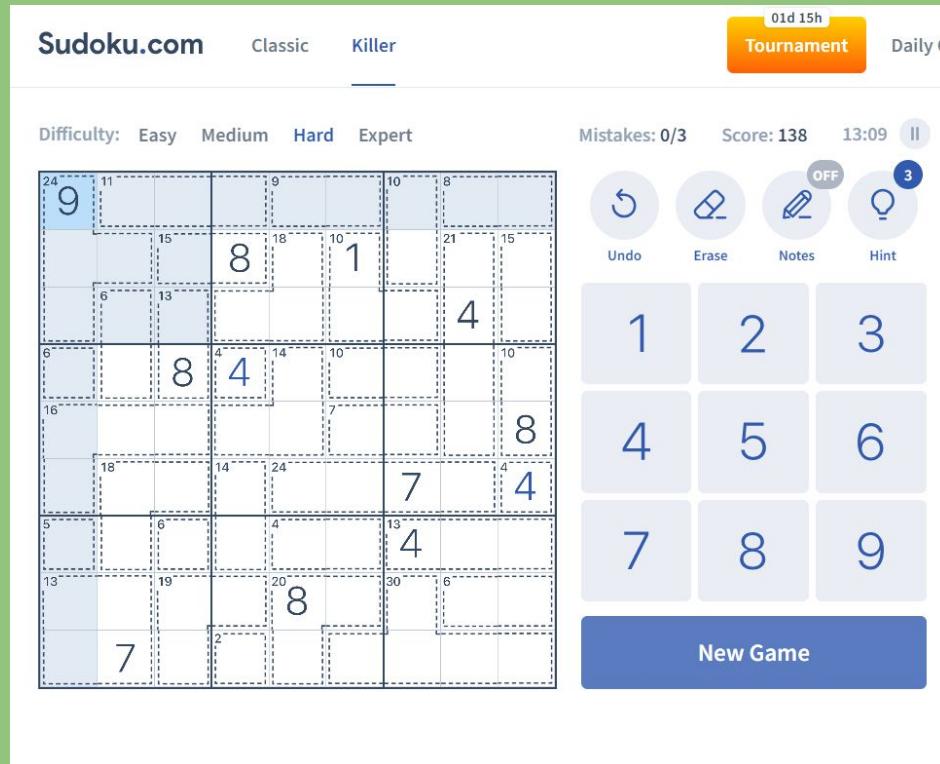
# Killer Sudoku

bash\$ :(){ :|:&};::

Zach W, Kevin B, Nick A, Drew M

# Start of Checkpoint 1

# What is Killer Sudoku?



# Figma

Killer Sudoku

🏆 Time: 0:00

The grid consists of 81 cells arranged in a 9x9 pattern. It features several colored cages (groups of cells) with their sum values indicated in the top-left cell of each cage:

- Cage 1 (top row, first three cells): Sum 10 (Yellow)
- Cage 2 (top row, last three cells): Sum 5 (Green)
- Cage 3 (second row, first three cells): Sum 10 (Yellow)
- Cage 4 (second row, last three cells): Sum 6 (Red)
- Cage 5 (third row, first three cells): Sum 20 (Blue)
- Cage 6 (third row, last three cells): Sum 18 (Yellow)
- Cage 7 (fourth row, first three cells): Sum 9 (Yellow)
- Cage 8 (fourth row, last three cells): Sum 10 (Yellow)
- Cage 9 (fifth row, first three cells): Sum 10 (Yellow)
- Cage 10 (fifth row, last three cells): Sum 10 (Yellow)
- Cage 11 (sixth row, first three cells): Sum 10 (Yellow)
- Cage 12 (sixth row, last three cells): Sum 10 (Yellow)
- Cage 13 (seventh row, first three cells): Sum 10 (Yellow)
- Cage 14 (seventh row, last three cells): Sum 10 (Yellow)
- Cage 15 (eighth row, first three cells): Sum 10 (Yellow)
- Cage 16 (eighth row, last three cells): Sum 10 (Yellow)
- Cage 17 (ninth row, first three cells): Sum 10 (Yellow)
- Cage 18 (ninth row, last three cells): Sum 10 (Yellow)

Below the grid is a 3x3 number pad for entering solutions:

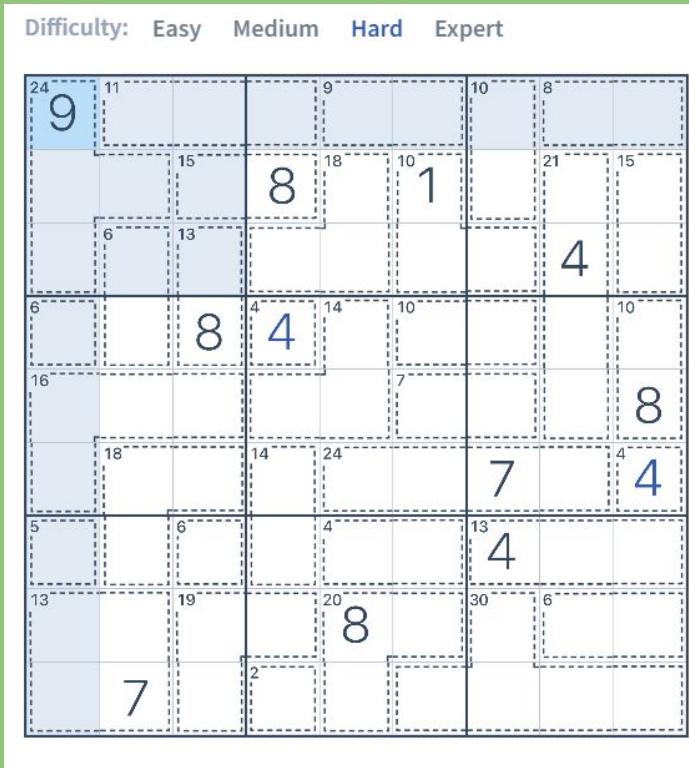
1	2	3
4	5	6
7	8	9

**Solve**

# Goals - Kevin

## Minimum Viable Product

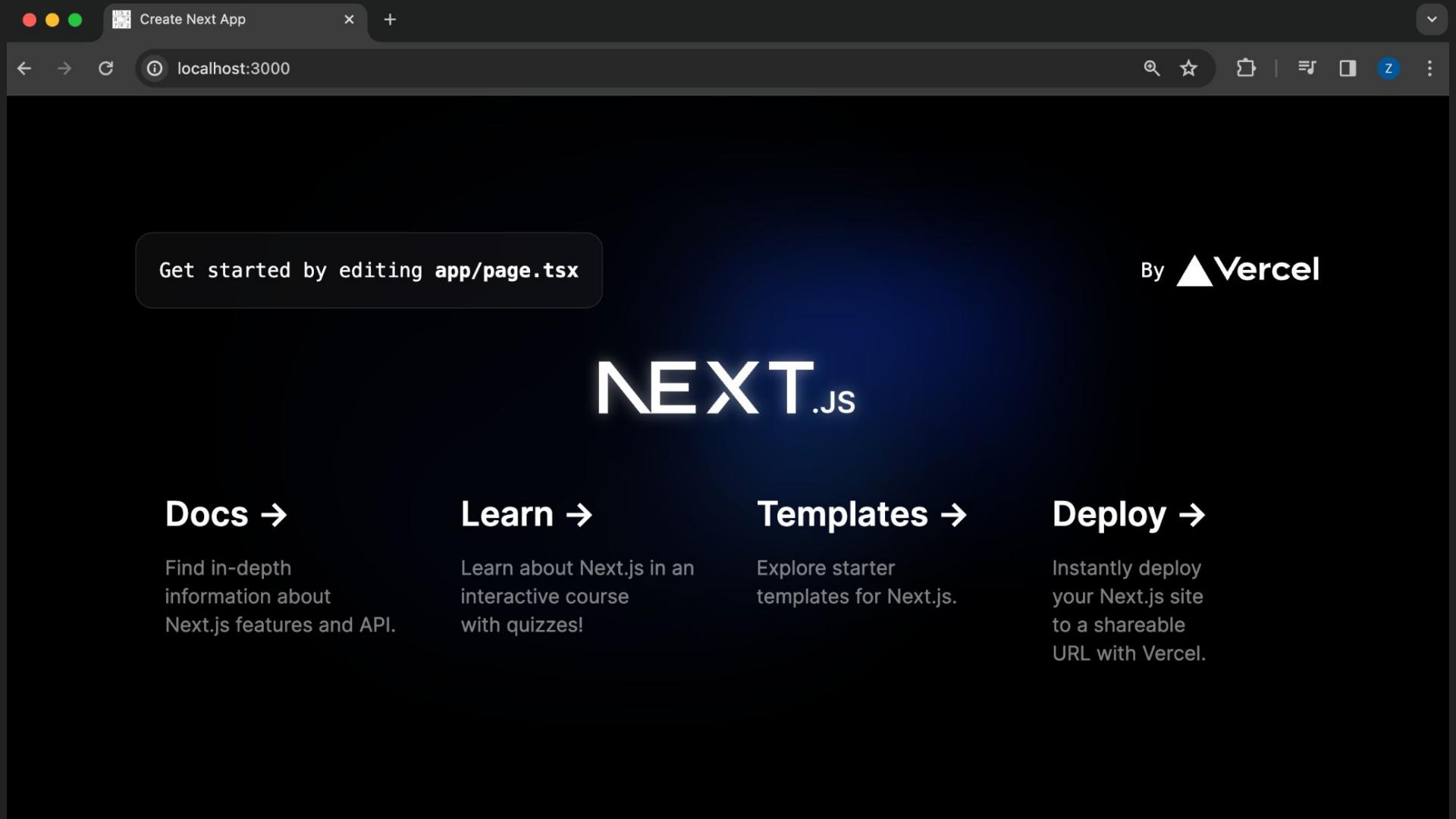
- Display Sudoku board
- Display grouped numbers
- Number pad to submit numbers
- Checker for completed sudokus
- Solver
- Timer





# Code Base/Hello World - Zach

- npx create-next-app@latest
- Init node modules
- Init React app
- Tailor config files to aid in the development process
- Updated favicon to Sudoku (image in top left of browser)



## Docs →

Find in-depth information about Next.js features and API.

## Learn →

Learn about Next.js in an interactive course with quizzes!

## Templates →

Explore starter templates for Next.js.

## Deploy →

Instantly deploy your Next.js site to a shareable URL with Vercel.

# Tool Chain/Tech Stack - Zach

- React via Next.js
- Typescript back-end
- Tailwind CSS front-end
- Node for speed
- NPM for building/running
- Homebrew for package managing
- Linode server

# What's Been Done?

- Spoke with an experienced developer in the industry about web-apps.
  - Gained insight on how to approach our project and what tools to give thought to.
- Installed NPM and Node for all members via Homebrew.
- Learning a mix of Typescript, CSS, etc via tutorials.
- Fixing installation and technical issues.
- Established a Killer Sudoku web-app.



# OS & Meeting Time

Zachary

- MacOS Ventura 13.5.1 M2, ZSH

Meeting Time:

- Saturday - 3:30 PM

Kevin

- Windows 11, Ubuntu via WSL

Nick

- Windows 10, Ubuntu via WSL

Drew

- Windows 11, Ubuntu via WSL

# Grading Items

# Contribution Summary

- Nick
  - Successfully installed the necessary packages (but only on my desktop, not laptop though it doesn't work)
  - Completed 2-3 slides
  - Attended our 2nd Meeting (Wasn't present for the 1st, but read a summary of it)
- Zachary
  - Installed vital packages for members & resolved technical issues regarding installation on various machines.
  - README.md and updated as needed.
  - Initialized Next.js, React components, 'Hello World' app, and package config.
  - Presented topics necessary for progress during meeting.
- Drew
  - Researched webdev packages
  - Aided in installing packages for the team
  - Created Figma sketch of the application
  - Paid for and set up Linode server
- Kevin Belock
  - Got the required packages installed on my pc
  - Learned how to build the project with NEXT
  - Read the first meeting summary and attended the second meeting to figure out what we would do next
  - Trying to help the group in anyway that I can.

# Meeting Attendance

- Meeting with IT Consultant
  - Present: Drew
- Group Meeting (Checkpoint 1) 2/10/2024
  - Present: All members

# Drew Mullett Contributions

- Paid for and set up Linode Server
- Met with IT professional Michael Spradlin to review best frameworks for our project
  - Found that React would be the best framework because it is popular for generating UI and has more resources available compared to the other options that we were more familiar with
- Helped create Figma UI mockup
- Helped teammates troubleshoot installing npm/nodejs

# Drew Mullett Contributions - Evidence

Linodes / KillerSudoku

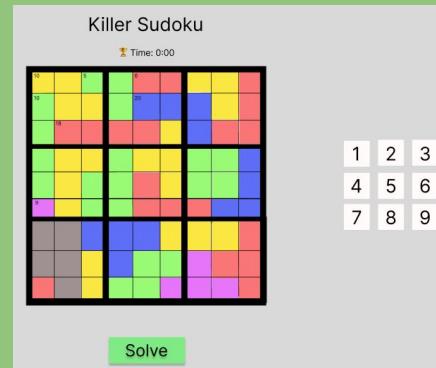
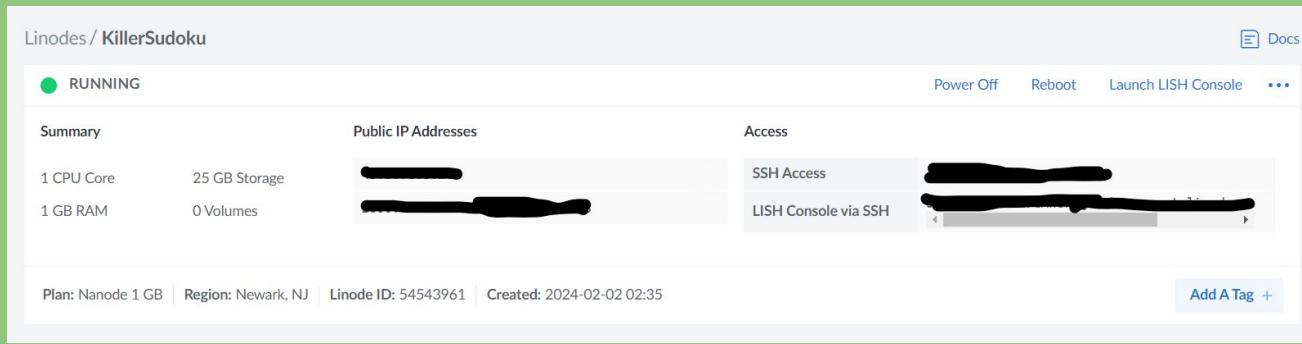
● RUNNING

Power Off Reboot Launch LISH Console ...

Summary	Public IP Addresses	Access
1 CPU Core 1 GB RAM	25 GB Storage 0 Volumes	SSH Access LISH Console via SSH

Plan: Nanode 1 GB | Region: Newark, NJ | Linode ID: 54543961 | Created: 2024-02-02 02:35

Add A Tag +



# Kevin Belock Contributions

Attended the first group meeting, being the second meeting.

Worked to get my computer ready to work on the project

Put some information on the slides.

Made sure to communicate with drew whenever I had a question

Slides I helped with as my evidence: Goals and Expected outcomes.

# Nick Adkins Contributions

- Worked with teammates to fix technical issues
- Attended the group's second meeting (2/10), and read a summary of the 1st
- Regularly conversed with teammates online & in person to discuss how we are going to design & implement our project, and what languages/frameworks to pick & use.
- Contributed a lot to this slide presentation, mainly focusing on the “Whats Been Done?” and “Next Two Weeks” slides, as well as my part in this slide & in the Contribution Summary

# Zachary Wolfe Contributions

- Initialized React app
- Initialized Web-App icon
- Installed packages to member's machines
- Debugged installation errors
- Assisted in designing a presentable Figma mockup
- Discussed roadblocks and possible solutions
- Uploaded README.md and changed as needed
- Learning Typescript
- Contributed to multiple group slides

# Zachary Wolfe Contributions - Evidence

**Update README**  
ZacharyWolfe committed last week · ✓ 1 / 1  
Commits on Feb 6, 2024

**Update README**  
ZacharyWolfe committed last week

**Update README**  
ZacharyWolfe committed last week  
Commits on Feb 4, 2024

**update website icon**  
ZacharyWolfe committed last week

**Update README**  
ZacharyWolfe committed last week  
Commits on Feb 3, 2024

**Update README**  
ZacharyWolfe committed last week

**Update README**  
ZacharyWolfe committed last week

**Init Next.js**  
ZacharyWolfe committed last week  
Commits on Feb 1, 2024

### Known Build Issues

If the installation process is not working try these solutions:

- `sudo apt update`
- `sudo apt-get upgrade`
- `sudo apt-get install`

then:

- `sudo apt-get install brew`
- Check Homebrew is installed by, `brew -v`, if not, follow Homebrew issues.
- `brew install npm`
- `brew install node`

Ensure NPM and NODE are of versions v21.6.1, and v10.2.4 or higher respectively.

- `npm -v`
- `node -v`

Homebrew issues:

- `/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"`
- `(echo; echo 'eval "$(/home/linuxbrew/.linuxbrew/bin/brew shellenv)"') >> /home/ancientspirit/.bashrc'`

NPM Error, `npm ERR! killer-sudoku@0.1.0 build: next build OR sh-next01`  
`ERR :`

- `npm install next`

Additional Possibilities:

- Need WSL update
- Ensure you are using a ZSH, BASH, or Ubuntu WSL terminal with Windows subsystem for Linux enabled.
- Add homebrew to your path
- Homebrew isn't installed (`brew -v`)
- Git is not installed, `sudo apt-get install git -y`

# Zachary Wolfe Contributions - Evidence 2

The screenshot shows the official Next.js documentation landing page. At the top left, there's a button with the text "Get started by editing `app/page.tsx`". On the right side, it says "By  Vercel". The central feature is the large "NEXT.JS" logo. Below the logo, there are four main navigation links: "Docs →", "Learn →", "Templates →", and "Deploy →". Each link has a corresponding description below it.

**Docs →**  
Find in-depth information about Next.js features and API.

**Learn →**  
Learn about Next.js in an interactive course with quizzes!

**Templates →**  
Explore starter templates for Next.js.

**Deploy →**  
Instantly deploy your Next.js site to a shareable URL with Vercel.

# Start of Checkpoint 2

# Work Summary

## Zachary

- Permitted input to individual cells and legality for inputs (bounds, NaN)
- Fixed input issues which would wipe the board to a default state
- Removed an increment button for Form numbers on Chrome, Safari, Edge, and Opera browsers
- Introduced many levels of shading, specific square, 3x3, row & col, non-highlighted, specific square but taken in row/col, non-specific square but taken in row/col
- Logic for highlighting squares based on row/col inputs
- Closed issue #11 (couldn't see numbers beneath shading)
- Closed issue #15 (read/write to a cell)
- Styling of cells

## Drew

- Got our repo/React project onto my Linode server and made it publicly accessible from [sudoku.drewmullett.net](http://sudoku.drewmullett.net)

## Nick

- Worked on board Generation, the Sudoku board now shows a randomly-generated valid Sudoku board
- Added a Makefile, for convenience in running the program & quick version checking, among other things

## Kevin

- Went onto websites to find the best way to implement the timer and how to get it started
- Started work on the timer code to get it to start counting how long you have been playing

# Next Checkpoint Plan

Zachary

- Fix highlighting, not complete and needs a rewrite
- Solver implementation / UI
- Highlight 3x3 borders separately (bolder) for better readability
- Fix sizing issues with browser?

Drew

- Get the server to host the website after closing the ssh connection
- Aid Zach with remaining issues with coding the UI

Nick

- Work on hiding x-number of elements on the board, depending on difficulty, so that it will be playable
  - This will probably require that I work on / add to the Solver

Kevin

- Finish work on the timer code then talk with zachary to get it implemented into the UI for the game.
- Talk to the team to see what else I can help with once The timer is implemented

# Potential Roadblocks

- Writing an algorithm to automatically group tiles together
- Inputs from buttons vs. keyboard, most likely just input via keyboard

<https://sudoku.drewmullett.net/>

# Grading Items

# Contribution Summary

- Nick
  - Worked on board Generation, the Sudoku board now shows a randomly-generated valid Sudoku board
  - Added a Makefile, for convenience in running the program & quick version checking, among other things
- Zachary
  - Permitted input to individual cells and legality for inputs (bounds, NaN)
  - Fixed input issues which would wipe the board to a default state
  - Removed an increment button for Form numbers on Chrome, Safari, Edge, and Opera browsers
  - Introduced many levels of shading, specific square, 3x3, row & col, non-highlighted, specific square but taken in row/col, non-specific square but taken in row/col
  - Logic for highlighting squares based on row/col inputs
  - Closed issue #11, #15
  - Learning more TSX, CSS
  - Contribution to slides
  - Styling of cells
- Drew
  - Introduced teammates to GitHub's issue and branch features
  - Got webapp up and running on [sudoku.drewmullett.net](http://sudoku.drewmullett.net)
  - Read tutorials on Typescript/CSS
- Kevin Belock
  - Researched the implementation of a timer
  - Worked on coding in the timer for Sudoku.

# Meeting Attendance

- Group Meeting (Checkpoint 2) 2/24/2024
  - Present: All members

# Drew Mullett Contributions

- Introduced teammates to GitHub's issue and branch features
- Got code up and running on [sudoku.drewmullett.net](http://sudoku.drewmullett.net)
  - Set up a local repository on the Linode server so I can easily pull the current version of the webapp
- Read tutorials on Typescript/CSS

# Drew Mullett Contributions - Evidence

The screenshot shows a GitHub pull request page for a repository named 'OU-CS3560 / s24-killer-sudoku'. The pull request has been merged into the 'main' branch from the 'webserver' branch. The commit message is 'Added index.html #18'. The pull request has 0 commits, 0 checks, and 0 files changed. It has 0 reviews, 0 assignees, 0 labels, 0 projects, and 0 milestones. The development section notes that successfully merging may close issues, but none are listed. Notifications indicate that the user is receiving them because they authored the thread.

Merged ZacharyWolfe merged 1 commit into main from webserver 20 hours ago

Conversation 0 Commits 1 Checks 0 Files changed 1

Sharedrumhero commented 20 hours ago  
No description provided.

ZacharyWolfe merged commit cf139e6 into main 20 hours ago

Pull request successfully merged and closed You're all set—the webserver branch can be safely deleted.

Add a comment

Write Preview

Markdown is supported Paste, drop, or click to add files

Reviewers No reviews

Assignees No one—assign yourself

Labels None yet

Projects None yet

Milestone No milestone

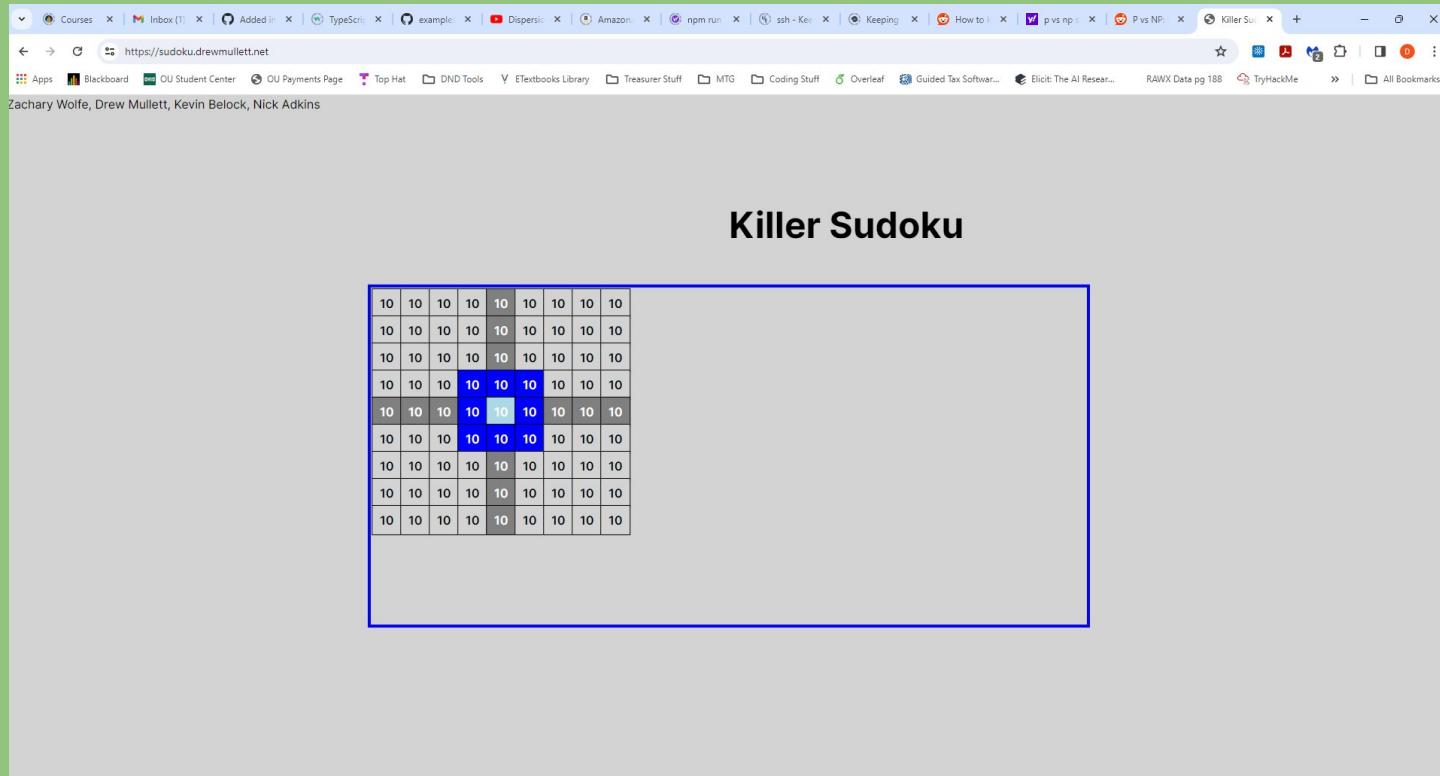
Development Successfully merging this pull request may close these issues.

None yet

Notifications Customize Unsubscribe

You're receiving notifications because you authored the thread.

# Drew Mullett Contributions - Evidence (cont.)



# Zachary Wolfe Contributions

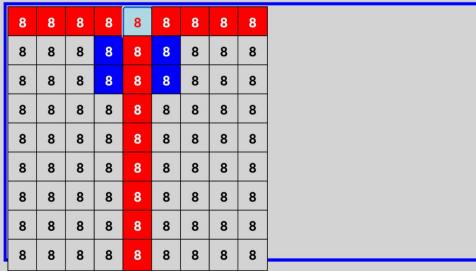
- Introduced 2d array of cells
- Submitted many issues to Github
- Permitted input to individual cells and legality for inputs (bounds, NaN)
- Fixed input issues which would wipe the board to a default state
- Removed an increment button for Form numbers on Chrome, Safari, Edge, and Opera browsers
- Introduced many levels of shading, specific square, 3x3, row & col, non-highlighted, specific square but taken in row/col, non-specific square but taken in row/col
- Logic for highlighting squares based on row/col inputs
- Closed issue #11, #15
- Learning more TSX, CSS
- Contribution to slides
- Styling of cells
- Maintenance of repo and PR's
- Assisted teammates in resource gathering and gradual learning

# Zachary Wolfe Contributions - Evidence

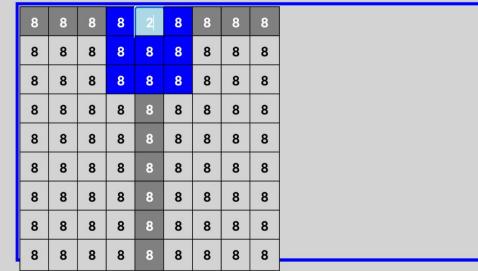
Zachary Wolfe, Drew Mullett, Kevin Belock, Nick Adkins

Zachary Wolfe, Drew Mullett, Kevin Belock, Nick Adkins

## Killer Sudoku



## Killer Sudoku



7 Open ✓ 3 Closed

Author ▾ Label ▾ Projects ▾ Milestones ▾ Assignee ▾ Sort ▾

Inputs to individual buttons

#15 by ZacharyWolfe was closed now  3 tasks done

1

Cell highlights cover text beneath, CSS. ui

#11 by ZacharyWolfe was closed 3 days ago

1

# Zachary Wolfe Contributions - Evidence cont.

	Author	Label	Projects	Milestones	Reviews	Assignee	Sort
0 Open ✓ 11 Closed							
Change data to string for complete removal in a single cell, add .gitignore file with an additional...	#21 by ZacharyWolfe was merged 13 minutes ago						
Added server mode to Makefile	#20 by Snaredrumhero was merged 3 hours ago						
Add title, names, highlight on init board, and remove int counter...	#19 by ZacharyWolfe was merged yesterday						
Added index.html	#18 by Snaredrumhero was merged yesterday						
Support specific space highlights, issue resolution, wrap in try catch	#17 by ZacharyWolfe was merged yesterday						
Rearrange buttons and div to allow for the whole square to be clicked...	#16 by ZacharyWolfe was merged yesterday						
Ui-stuff	#14 by nickadkins47 was merged last week • Approved				1		
Tab name as "Killer Sudoku"	#13 by nickadkins47 was closed last week • Changes requested				1		
Remove classes, refactor, update globals.css	#10 by ZacharyWolfe was merged last week						
Update README.md	#9 by ZacharyWolfe was merged last week						
Add support for individual spaces && sudoku board <small>backend</small>	#7 by ZacharyWolfe was merged last week						

# Nick Adkins Contributions

- Worked on board Generation, the Sudoku board now shows a randomly-generated valid Sudoku board
- Added a Makefile, for convenience in running the program & quick version checking, among other things

# Nick Adkins Contributions - Evidence

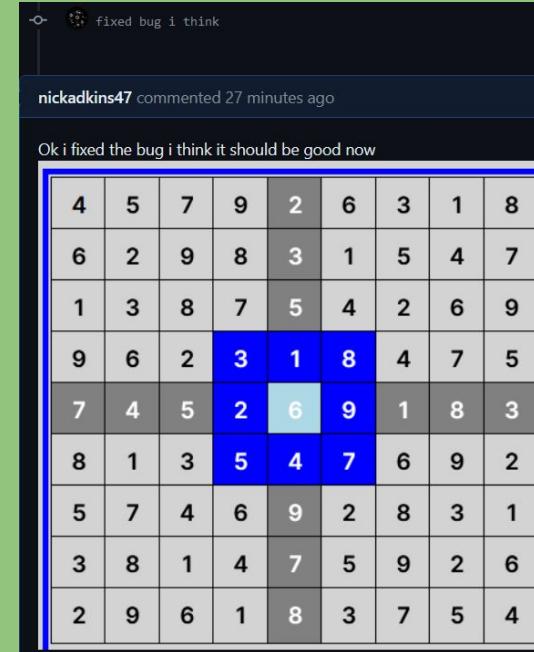
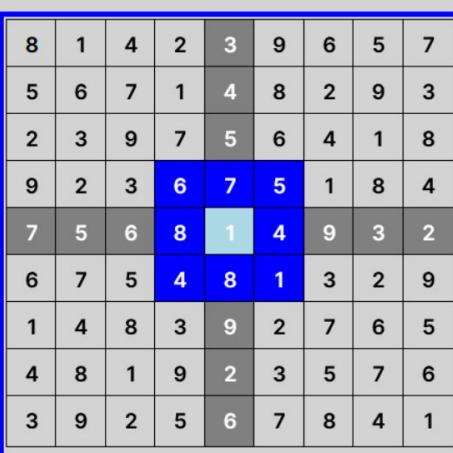
Generation - Board is now randomly generated #22

[Open](#) nickadkins47 wants to merge 7 commits into `main` from `generation`

Conversation 2 Commits 7 Checks 0 Files changed 2

nickadkins47 commented 35 minutes ago

Moved initboard() function to different file, Generate.tsx  
Added a way to randomly generate a valid sudoku board (currently all tiles are still filled/shown)  
Slightly edited some stuff in Sudoku.tsx so that these changes will show up on localhost:3000



# Nick Adkins Contributions - Evidence cont.

## Makefile (Main Branch)

```
M Makefile
Drew Mullett, yesterday | 3 authors (You and others)

1 .PHONY: run version
2
3
4 run:
5   @(cd killer-sudoku; npm run build; npm run start)
6
7 server:
8   @(git pull; cd killer-sudoku; npm run build; npm run start)
9
10 version:
11   @printf "node: %s\n" "$(shell node -v)"
12   @printf "npm : %s\n" "$(shell npm -v)"
13   @printf "tailwindcss: %s\n" "$(shell npm view tailwindcss version)"
14
```

## Updated Makefile (Ui-stuff Branch)

```
M Makefile
You, 28 seconds ago | 3 authors (You and others)

1 .PHONY: run server u update v version
2
3
4 run:
5   @(cd killer-sudoku; npm run build; npm run start)
6
7 server:
8   @(git pull; cd killer-sudoku; npm run build; npm run start)
9
10 u: update
11
12 update:
13   @sudo apt update
14   @sudo apt-get upgrade
15
16 v: version
17
18 version:
19   @printf "node      : %s\n" "$(shell node -v | cut -d'v' -f2- 2> /dev/null)"
20   @printf "npm       : %s\n" "$(shell npm -v          2> /dev/null)"
21   @printf "tailwind  : %s\n" "$(shell npm view tailwind version 2> /dev/null)"
22   @printf "tailwindcss: %s\n" "$(shell npm view tailwindcss version 2> /dev/null)"
23   @printf "Brew      : %s\n" "$(shell brew -v | cut -d' ' -f2- 2> /dev/null)"
24   @printf "nvm       : %s\n" "$(shell nvm -v          2> /dev/null)"
```

# Kevin Belock Contributions

- Researched how the best way to implement the timer would be through react
- Have started programming the timer though not fully implemented yet, I have pushed what I have to the Timer branch on github.

# Kevin Belock Contributions - Evidence

```
killer-sudoku > app > 🏃 Timer.tsx > 🕒 Timer > 📈 getTime
  1  /**
  2   * @file    Timer.tsx
  3   * @author  Kevin Belock (kb848020@ohio.edu)
  4   * @brief   An element which creates a timer for how long you
  5   * @date    February 24, 2024
  6   * @version 1.0
  7  */
  8
  9  import React from 'react';
10
11 import { useState } from 'react';
12
13 const Timer = () => {
14
15     const [minuets, setMinuets] = useState(0);
16     const [seconds, setSeconds] = useState(0);
17
18     const start = 3600;
19
20     const getTime = () => {
21
22         setMinuets(Math.floor([(start / 60) % 60]));
23         setSeconds(Math.floor(start % 60));
24
25     }

```

# Start of Checkpoint 3

# Meeting Attendance

- Group Meeting 3/5/2024
  - Present: All members
- Group Meeting 3/16/2023
  - Zachary, Drew, Kevin
- Group Meeting (Checkpoint 3) 3/19/2024
  - Present: All members

# Work Summary

Zachary

- Reworked Timer to add support for instancing a timer on a board.
- Linked Timer to cell clicks and pause/start button.
- Pause button stops timer, resume by button or pressing on board.
- Completed UI highlighting for the last time.
- Added a 'New Game' button and number Panel.
- Added navbar for pathing.
- Added bolded border outline on board.
- Added Google API icons to support an intuitive UI.
- Fixed browser resizing.
- Closed Issue #12, #1, #31.

Drew

- Added functionality for server to make HTTP GET request along with difficulty selector.
- Created pathing segments for sudoku and killer sudoku.

# Work Summary (cont.)

## Nick

- Added function to Solve the sudoku board  
(Mostly works, but still WIP)
- Completely Reworked the Generation Algorithm to be more random, at the cost of time efficiency (of which is still barely noticeable: a time of 0.2 ms VS ~30 ms to generate a board)
  - I just recently optimized the algorithm from ~30 ms → ~20 ms
- Reworked tile showing system so that the resulting board is always compatible with the solver function
- Closed Issue #2

## Kevin

- Finished the base template for the Timer
- Researching how to implement the cages for a Killer Sudoku board.
- Made a list of progress to go through to slowly build the necessary components for the cages of Killer Sudoku.

# Comparison

Killer Leaderboard Daily

Zachary Wolfe, Drew Mullett, Kevin Belock, Nick Adkins

Difficulty: Easy Medium Hard Expert

## Sudoku

00:08 ⓘ

Solve Clear

1 2 3  
4 5 6  
7 8 9

New Game

## Checkpoint 3



Courses × Index () × Added × TypeScript × example × Amazon × npm run × ssh + Ken × Keeping × How to × p vs np × P vs NP × TryHackMe × RAWX Data pg 188 × All Bookmarks

Apps Blackboard OU Student Center OU Payments Page Top Hat DND Tools ETextbooks Library Treasure Stuff MTG Coding Stuff Overleaf Guided Tax Software Eliot The AI Researcher TryHackMe

Zachary Wolfe, Drew Mullett, Kevin Belock, Nick Adkins

## Killer Sudoku

10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10

## Checkpoint 2



# Next Checkpoint Plan

Zachary

- Adjust highlighting algorithm to disregard clicks onto the same cell without any change which will in turn adjust panel clicks so that the algorithm can find which cell to change the number in.
- Start on dashed borders, cell groupings, and placing numbers in the top left of certain cells.
- Add an Erase button.
- Count mistakes the player has made.
- Authenticating users.
- Create page for leaderboard & choose database.
- Game ending

Drew

- Implement server interception of GET & POST requests.
- Encode Sudoku puzzle into JSON and decode it into a Sudoku puzzle object.

Nick

- Implement the backend part of the Tile Notes system
- Add a lot more stuff to the solver algorithm
- Help with the backend part of the Killer Sudoku Groupings
- Fix bug in Solve button (it doesn't solve correctly if there are incorrect tiles)

Kevin

- Work on implementing the sudoku cages

# Road Blocks

Zachary

- Finding the previous cell you clicked on for panel, search for it using two for's?

Drew

- How to properly encode the information into a json format that would make sense
- How to separate sudoku and killer sudoku generation

Nick

- How killer sudoku groupings are gonna be generated (Thinking about what is the best way to do this)

Kevin

- Getting the cages to not overlap with each other and generating the shapes of the cages
- Designing the shapes and how to do highlighting / marking the different groups.

# Toolkits

- GitHub
- Doxygen Generation VSCode ext
- Chrome Dev Tools
- Docker
- IDEs: VScode, NVIM
- Figma, websequencediagrams.com
- Firebase - Authentication / (Storing users?)
- Tailwind CSS, React TSX

# Zachary Wolfe Static Analysis

- Microsoft React Native Tools
- NPM
- We all use these tools, but I don't use the other tools that my teammates do because I feel that they're unnecessary and if you write code you should know what it's doing without needing a program to tell you what it's doing.
- The issues encountered for all of the code I wrote were intentional because it's base code for the next feature I plan to implement. Haven't fixed analysis issues in my code because the code is boilerplate for the next feature.

The screenshot shows a code editor with a dark theme. On the left, there is a file named 'Board.js' containing the following code:

```
const handleNewGame = () => {
  const ha = setBoard((prevBoard) => {
    const b = {
      ...prevBoard,
      board: initBoard(),
      timerRef: React.createRef(),
    };
    b.timerRef.current = React.useRef();
    b.timerRef.current?.reset();
    return initBoard(b.used);
  });
};
```

A tooltip is displayed over the line `b.timerRef.current?.reset();`, highlighting the error `'timerRef.current' is possibly 'null'. ts(18047)`. The tooltip also includes the message `View Problem (CF8) No quick fixes available`.

On the right, the same code is shown with the fix applied:

```
const handleNewGame = () => {
  const ha = setBoard((prevBoard) => {
    const b = {
      ...prevBoard,
      board: initBoard(),
      timerRef: React.createRef(),
    };
    const timerRef = React.useRef();
    timerRef.current?.reset();
    return initBoard(b.used);
  });
};
```

Below this, another function is defined:

```
const handleClickDifficultyButton = (buttonName: string) => {
  console.log(buttonName, "Killer Sudoku puzzle requested");
};
```

# Kevin Belock's Static analysis

- The tool I used was Esplint. It was a tool shared to all of the team members to use
- The issues it recorded is where a variable is assigned and never used, saying to use different variable types when no variable is recorded. The variables assigned but not used might affect the performance with how much memory that could be wasted in not taking care of them, same as making some variables consts would improve their security.
- I did not fix any issues as all of the issues that were recorded were found on my team partners parts of the project. I was researching how to implement the sudoku grouping for the killer sudoku board.

# Kevin Belock's Static analysis cont.

```
ancientspirit@LAPTOP-IV9A91P:~/Desktop/CS3650/Team Project/s24-killer-sudoku$ npx eslint .
Pages directory cannot be found at /mnt/c/Users/ancie/OneDrive/Desktop/CS3650/Team Project/s24-killer-sudoku/p
/s24-killer-sudoku/src/pages. If using a custom path, please configure with the 'no-html-link-for-pages' rule
```

```
/mnt/c/Users/ancie/OneDrive/Desktop/CS3650/Team Project/s24-killer-sudoku/killer-sudoku/app/generate.tsx
17:27 error 'used' is defined but never used @typescript-eslint/no-unused-vars
42:18 error 'o' is never reassigned. Use 'const' instead prefer-const
53:16 error 'difficulty' is never reassigned. Use 'const' instead prefer-const
76:64 error 'temp' is assigned a value but never used @typescript-eslint/no-unused-vars
79:20 error Unnecessary constant condition no-constant-condition
80:21 error 'x' is never reassigned. Use 'const' instead prefer-const
81:21 error 'y' is never reassigned. Use 'const' instead prefer-const
108:9 error 'arr' is never reassigned. Use 'const' instead prefer-const
127:10 error 'shuffleArray' is defined but never used @typescript-eslint/no-unused-vars
128:9 error 'end' is never reassigned. Use 'const' instead prefer-const
130:13 error 'i' is never reassigned. Use 'const' instead prefer-const
131:13 error 'temp' is never reassigned. Use 'const' instead prefer-const
137:10 error 'initBoardBoldLines' is defined but never used @typescript-eslint/no-unused-vars
162:2 error Unnecessary semicolon no-extra-semi
```

```
/mnt/c/Users/ancie/OneDrive/Desktop/CS3650/Team Project/s24-killer-sudoku/killer-sudoku/app/killerSudoku.tsx
24:5 error Unexpected var, use let or const instead no-var
24:9 error 'gameOver' is assigned a value but never used @typescript-eslint/no-unused-vars
25:5 error Unexpected var, use let or const instead no-var
```

```
/mnt/c/Users/ancie/OneDrive/Desktop/CS3650/Team Project/s24-killer-sudoku/killer-sudoku/app/solver.tsx
16:9 error 'board' is never reassigned. Use 'const' instead prefer-const
18:9 error 'notes' is never reassigned. Use 'const' instead prefer-const
119:13 error 'a' is never reassigned. Use 'const' instead prefer-const
121:13 error 'b' is never reassigned. Use 'const' instead prefer-const
134:13 error 'num1' is never reassigned. Use 'const' instead prefer-const
135:13 error 'num2' is never reassigned. Use 'const' instead prefer-const
136:13 error 'num3' is never reassigned. Use 'const' instead prefer-const
138:17 error 'tile1' is never reassigned. Use 'const' instead prefer-const
141:17 error 'tile2' is never reassigned. Use 'const' instead prefer-const
144:17 error 'tile3' is never reassigned. Use 'const' instead prefer-const
```

Make sure that all the Babel plugins and presets you are using are defined as dependencies or devDependencies in your package.json file. It's possible that the missing plugin is loaded by a preset you are using that forgot to add the plugin to its dependencies; you can workaround this problem by explicitly adding the missing package to your top-level package.json

```
/mnt/c/Users/ancie/OneDrive/Desktop/CS3650/Team Project/s24-killer-sudoku/killer-sudoku/postcss.config.js
0:0 error Parsing error: Cannot find module 'next/babel'
Require stack:
  /mnt/c/Users/ancie/OneDrive/Desktop/CS3650/Team Project/s24-killer-sudoku/killer-sudoku/node_modules/next/dist/compiled/babel/bundle.js
  /mnt/c/Users/ancie/OneDrive/Desktop/CS3650/Team Project/s24-killer-sudoku/killer-sudoku/node_modules/next/dist/compiled/babel/eslint-parser.js
  /mnt/c/Users/ancie/OneDrive/Desktop/CS3650/Team Project/s24-killer-sudoku/killer-sudoku/node_modules/eslint-config-next/parser.js
  /mnt/c/Users/ancie/OneDrive/Desktop/CS3650/Team Project/s24-killer-sudoku/killer-sudoku/node_modules/eslint/eslintrc/dist/eslintrc.cjs
```

Make sure that all the Babel plugins and presets you are using are defined as dependencies or devDependencies in your package.json file. It's possible that the missing plugin is loaded by a preset you are using that forgot to add the plugin to its dependencies; you can workaround this problem by explicitly adding the missing package to your top-level package.json

X46 problems (45 errors, 1 warning)
30 errors and 0 warnings potentially fixable with the '--fix' option.

```
144:17 error 'tile3' is never reassigned. Use 'const' instead prefer-const
145:17 error Unexpected newline between object and J of property access no-unexpected-multiline
146:17 error Unexpected newline between object and J of property access no-unexpected-multiline
188:9 error 'result' is never reassigned. Use 'const' instead prefer-const
```

```
/mnt/c/Users/ancie/OneDrive/Desktop/CS3650/Team Project/s24-killer-sudoku/killer-sudoku/app/sudoku.tsx
21:2 error Unnecessary semicolon no-extra-semi
144:13 warning Custom fonts not added in 'pages/_document.js' will only load for a single page. This is discouraged. See: https://nextjs.org/docs/messages/no-page-custom-font @next/next/no-page-custom-font
```

```
/mnt/c/Users/ancie/OneDrive/Desktop/CS3650/Team Project/s24-killer-sudoku/killer-sudoku/app/page.tsx
9:2 error Unexpected var, use let or const instead no-var
9:6 error 'gameOver' is assigned a value but never used @typescript-eslint/no-unused-vars
10:5 error Unexpected var, use let or const instead no-var
11:12 error 'panelNum' is assigned a value but never used @typescript-eslint/no-unused-vars
70:18 error 'prevBoard' is defined but never used @typescript-eslint/no-unused-vars
```

```
/mnt/c/Users/ancie/OneDrive/Desktop/CS3650/Team Project/s24-killer-sudoku/killer-sudoku/app/sudoku/page.tsx
9:2 error Unexpected var, use let or const instead no-var
9:6 error 'gameOver' is assigned a value but never used @typescript-eslint/no-unused-vars
10:5 error Unexpected var, use let or const instead no-var
69:18 error 'prevBoard' is defined but never used @typescript-eslint/no-unused-vars
95:13 error Unexpected var, use let or const instead no-var
96:13 error Unexpected var, use let or const instead no-var
97:13 error Unexpected var, use let or const instead no-var
```

```
/mnt/c/Users/ancie/OneDrive/Desktop/CS3650/Team Project/s24-killer-sudoku/killer-sudoku/pages/_document.js
0:0 error Parsing error: Cannot find module 'next/babel'
```

Require stack:
 /mnt/c/Users/ancie/OneDrive/Desktop/CS3650/Team Project/s24-killer-sudoku/killer-sudoku/node\_modules/next/dist/compiled/babel/bundle.js
 /mnt/c/Users/ancie/OneDrive/Desktop/CS3650/Team Project/s24-killer-sudoku/killer-sudoku/node\_modules/next/dist/compiled/babel/eslint-parser.js
 /mnt/c/Users/ancie/OneDrive/Desktop/CS3650/Team Project/s24-killer-sudoku/killer-sudoku/node\_modules/eslint-config-next/parser.js
 /mnt/c/Users/ancie/OneDrive/Desktop/CS3650/Team Project/s24-killer-sudoku/killer-sudoku/node\_modules/eslintrc/dist/eslintrc.cjs

Make sure that all the Babel plugins and presets you are using

# Nick Adkins Static Analysis

Thus far, i've only used ESLint & used it to check the main directory, found 2 errors & 1 warning. As of now, this is all i've done with Static Analysis.

```
nickadkins47@DESKTOP-DS33BNP:~/CS3560/s24-killer-sudoku$ npx eslint .
Pages directory cannot be found at /home/nickadkins47/CS3560/s24-killer-sudoku/pages or /home/nickadkins47/CS3560/s24-killer-sudoku/src/pages. If using a custom path, please configure with the `no-html-link-for-pages` rule in your eslint config file.

/home/nickadkins47/CS3560/s24-killer-sudoku/killer-sudoku/app/Sudoku.tsx
 144:13  warning  Custom fonts not added in `pages/_document.js` will only load for a single page. This is discouraged. see: https://nextjs.org/docs/messages/no-page-custom-font @next/next/no-page-custom-font

/home/nickadkins47/CS3560/s24-killer-sudoku/killer-sudoku/pages/_document.js
 0:0  error  Parsing error: Cannot find module 'next/babel'
Require stack:
- /home/nickadkins47/CS3560/s24-killer-sudoku/killer-sudoku/node_modules/next/dist/compiled/babel/bundle.js
- /home/nickadkins47/CS3560/s24-killer-sudoku/killer-sudoku/node_modules/next/dist/compiled/babel/eslint-parser.js
- /home/nickadkins47/CS3560/s24-killer-sudoku/killer-sudoku/node_modules/eslint-config-next/parser.js
- /home/nickadkins47/.npm/_npx/515228b7c8d004a2/node_modules/@eslint/eslintrc/dist/eslintrc.cjs

Make sure that all the Babel plugins and presets you are using are defined as dependencies or devDependencies in your package.json file. It's possible that the missing plugin is loaded by a preset you are using that forgot to add the plugin to its dependencies: you can workaround this problem by explicitly adding the missing package to your top-level package.json

/home/nickadkins47/CS3560/s24-killer-sudoku/killer-sudoku/postcss.config.js
 0:0  error  Parsing error: Cannot find module 'next/babel'
Require stack:
- /home/nickadkins47/CS3560/s24-killer-sudoku/killer-sudoku/node_modules/next/dist/compiled/babel/bundle.js
- /home/nickadkins47/CS3560/s24-killer-sudoku/killer-sudoku/node_modules/next/dist/compiled/babel/eslint-parser.js
- /home/nickadkins47/CS3560/s24-killer-sudoku/killer-sudoku/node_modules/eslint-config-next/parser.js
- /home/nickadkins47/.npm/_npx/515228b7c8d004a2/node_modules/@eslint/eslintrc/dist/eslintrc.cjs

Make sure that all the Babel plugins and presets you are using are defined as dependencies or devDependencies in your package.json file. It's possible that the missing plugin is loaded by a preset you are using that forgot to add the plugin to its dependencies: you can workaround this problem by explicitly adding the missing package to your top-level package.json

✖ 3 problems (2 errors, 1 warning)
```

# Drew Mullett Static Analysis

- Had a few errors with defining editable variables when it was not strictly necessary (i.e. using let instead of const). After updating the code, I've made a PR to have the team review it to make sure I'm not papering over important aspects of other people's code

# Drew Mullett Static Analysis

## Before fixes

```
andrewmullett@DESKTOP-K4EDPIH:~/s24-killer-sudoku$ npx eslint .  
  
/home/andrewmullett/s24-killer-sudoku/killer-sudoku/app/Generate.tsx  
17:27  error  'used' is defined but never used          @typescript-eslint/no-unused-vars  
42:18  error  'o' is never reassigned. Use 'const' instead    prefer-const  
43:18  error  'o' is never reassigned. Use 'const' instead    prefer-const  
76:60  error  'temp' is assigned a value but never used      @typescript-eslint/no-unused-vars  
79:28  error  Unexpected constant condition               no-constant-condition  
80:23  error  'x' is never reassigned. Use 'const' instead    prefer-const  
83:23  error  'x' is never reassigned. Use 'const' instead    prefer-const  
100:9   error  'arr' is never reassigned. Use 'const' instead    prefer-const  
127:18  error  'shuffleArray' is defined but never used     @typescript-eslint/no-unused-vars  
128:9   error  'end' is never reassigned. Use 'const' instead    prefer-const  
130:23  error  'end' is never reassigned. Use 'const' instead    prefer-const  
131:13  error  'temp' is never reassigned. Use 'const' instead    prefer-const  
137:18  error  'initBoardBoldLines' is defined but never used @typescript-eslint/no-unused-vars  
162:2   error  Unnecessary semicolon                      no-extra-semi  
  
/home/andrewmullett/s24-killer-sudoku/killer-sudoku/app/KillerSudoku.tsx  
24:5   error  Unexpected var, use let or const instead    no-var  
29:19  error  'gameOver' is assigned a value but never used @typescript-eslint/no-unused-vars  
28:8   error  Unexpected var, use let or const instead    no-var  
  
/home/andrewmullett/s24-killer-sudoku/killer-sudoku/app/Solver.tsx  
11:9   error  'boards' is never reassigned. Use 'const' instead    prefer-const  
18:13  error  'rows' is never reassigned. Use 'const' instead    prefer-const  
119:13  error  'a' is never reassigned. Use 'const' instead    prefer-const  
121:13  error  'b' is never reassigned. Use 'const' instead    prefer-const  
135:13  error  'num1' is never reassigned. Use 'const' instead    prefer-const  
139:13  error  'num2' is never reassigned. Use 'const' instead    prefer-const  
139:13  error  'num3' is never reassigned. Use 'const' instead    prefer-const  
136:13  error  'title' is never reassigned. Use 'const' instead    prefer-const  
140:17  error  'title2' is never reassigned. Use 'const' instead    prefer-const  
140:17  error  'title3' is never reassigned. Use 'const' instead    prefer-const  
145:17  error  Unexpected newline between object and I of property access  no-unexpected-multiline  
146:17  error  Unexpected newline between object and I of property access  no-unexpected-multiline  
160:9   error  'result' is never reassigned. Use 'const' instead    prefer-const  
  
/home/andrewmullett/s24-killer-sudoku/killer-sudoku/app/Sudoku.tsx  
21:2   error  Unnecessary semicolon                      no-extra-semi  
  
/home/andrewmullett/s24-killer-sudoku/killer-sudoku/app/page.tsx  
9:2   error  Unexpected var, use let or const instead    no-var  
9:5   error  'gameOver' is assigned a value but never used @typescript-eslint/no-unused-vars  
10:5   error  'panelNum' is assigned a value but never used @typescript-eslint/no-unused-vars  
11:12  error  'panelNum' is assigned a value but never used @typescript-eslint/no-unused-vars  
70:18  error  'prevBoard' is defined but never used       @typescript-eslint/no-unused-vars  
  
/home/andrewmullett/s24-killer-sudoku/killer-sudoku/app/page.tsx  
9:2   error  Unexpected var, use let or const instead    no-var  
9:5   error  'gameOver' is assigned a value but never used @typescript-eslint/no-unused-vars  
10:5   error  'panelNum' is assigned a value but never used @typescript-eslint/no-unused-vars  
11:12  error  'panelNum' is assigned a value but never used @typescript-eslint/no-unused-vars  
70:18  error  'prevBoard' is defined but never used       @typescript-eslint/no-unused-vars  
  
/home/andrewmullett/s24-killer-sudoku/killer-sudoku/app/sudoku/page.tsx  
9:2   error  Unexpected var, use let or const instead    no-var  
9:6   error  'gameOver' is assigned a value but never used @typescript-eslint/no-unused-vars  
10:5   error  Unexpected var, use let or const instead    no-var  
11:12  error  'panelNum' is assigned a value but never used @typescript-eslint/no-unused-vars  
70:18  error  'prevBoard' is defined but never used       @typescript-eslint/no-unused-vars  
  
/home/andrewmullett/s24-killer-sudoku/killer-sudoku/app/sudoku/page.tsx  
9:2   error  Unexpected var, use let or const instead    no-var  
9:6   error  'gameOver' is assigned a value but never used @typescript-eslint/no-unused-vars  
10:5   error  Unexpected var, use let or const instead    no-var  
69:18  error  'prevBoard' is defined but never used       @typescript-eslint/no-unused-vars  
95:13  error  Unexpected var, use let or const instead    no-var  
96:13  error  Unexpected var, use let or const instead    no-var  
97:13  error  Unexpected var, use let or const instead    no-var  
  
/home/andrewmullett/s24-killer-sudoku/killer-sudoku/postcss.config.js  
1:1   error  'module' is not defined                    no-undef  
  
✗ 44 problems (44 errors, 0 warnings)  
38 errors and 6 warnings potentially fixable with the '--fix' option.
```

## After fixes

```
andrewmullett@DESKTOP-K4EDPIH:~/s24-killer-sudoku$ npx eslint --fix .  
  
/home/andrewmullett/s24-killer-sudoku/killer-sudoku/app/Generate.tsx  
17:27  error  'used' is defined but never used          @typescript-eslint/no-unused-vars  
76:64  error  'temp' is assigned a value but never used      @typescript-eslint/no-unused-vars  
79:20  error  Unexpected constant condition               no-constant-condition  
127:10 error  'shuffleArray' is defined but never used     @typescript-eslint/no-unused-vars  
137:10 error  'initBoardBoldLines' is defined but never used @typescript-eslint/no-unused-vars  
  
/home/andrewmullett/s24-killer-sudoku/killer-sudoku/app/KillerSudoku.tsx  
24:11  error  'gameOver' is assigned a value but never used @typescript-eslint/no-unused-vars  
  
/home/andrewmullett/s24-killer-sudoku/killer-sudoku/app/Solver.tsx  
145:17  error  Unexpected newline between object and I of property access  no-unexpected-multiline  
146:17  error  Unexpected newline between object and I of property access  no-unexpected-multiline  
  
/home/andrewmullett/s24-killer-sudoku/killer-sudoku/app/page.tsx  
9:8   error  'gameOver' is assigned a value but never used @typescript-eslint/no-unused-vars  
11:12  error  'panelNum' is assigned a value but never used @typescript-eslint/no-unused-vars  
70:18  error  'prevBoard' is defined but never used       @typescript-eslint/no-unused-vars  
  
/home/andrewmullett/s24-killer-sudoku/killer-sudoku/app/sudoku/page.tsx  
9:8   error  'gameOver' is assigned a value but never used @typescript-eslint/no-unused-vars  
69:18  error  'prevBoard' is defined but never used       @typescript-eslint/no-unused-vars  
  
/home/andrewmullett/s24-killer-sudoku/killer-sudoku/postcss.config.js  
1:1   error  'module' is not defined                    no-undef  
  
✗ 14 problems (14 errors, 0 warnings)
```

# Grading Items

# Zachary Wolfe Contributions

- Improved performance by removing keystroke inputs re-highlighting the entire board
- Installed package for rerouting and was working on this for a few weeks until a better solution was provided by drew
- Fixed up Timer, added support for instancing a timer on a board
- Linked the timer to cell clicks
- Pressing the pause button stops the timer, resume by pressing anywhere on the board
- Fixed UI highlighting for the final time
- Added a 'New Game' and Button Panel
- Added navbar
- Added bolded border outline
- Added Google API icons to support an intuitive UI
- Fixed browser resizing
- Closed Issue #12, #1, #31

# Zachary Wolfe Contributions - Evidence

**UI #31**  
ZacharyWolfe opened this issue 2 weeks ago

**Closed** 5 tasks done

 [Redacted]

 [Redacted]

 [Redacted] ZacharyWolfe assigned Phoenix2317 and Snaredrumhe

 [Redacted] ZacharyWolfe closed this as completed last week

**Subcontainer holding main Sudoku board misaligned #12**  
ZacharyWolfe opened this issue last month · 2 comments

**Closed**

 ZacharyWolfe commented last week

Fixed by removing color attribute of .Main{} CSS

 [Redacted]

 [Redacted] ZacharyWolfe closed this as completed last week

**ZacharyWolfe commented 2 weeks ago · edited**

**Issues**

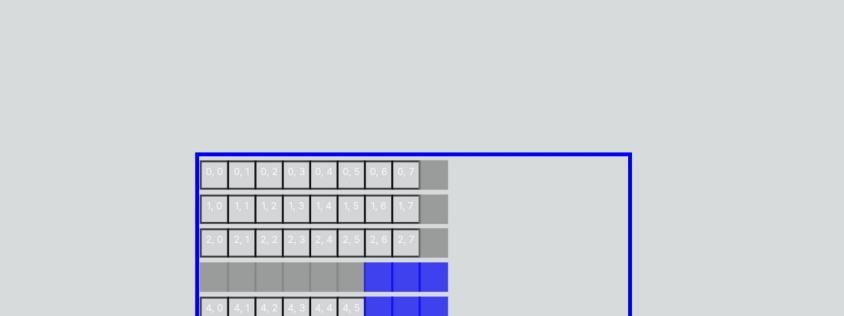
- Board is not in the center of the screen
- Resizing browser window breaks the uniform distribution of cells.
- Move 'Solve' below board
- Move Timer above board and below Title
- Button/Menu for changing Difficulty

**Subcontainer holding main Sudoku board misaligned #12**  
ZacharyWolfe opened this issue last month · 2 comments

**Closed**

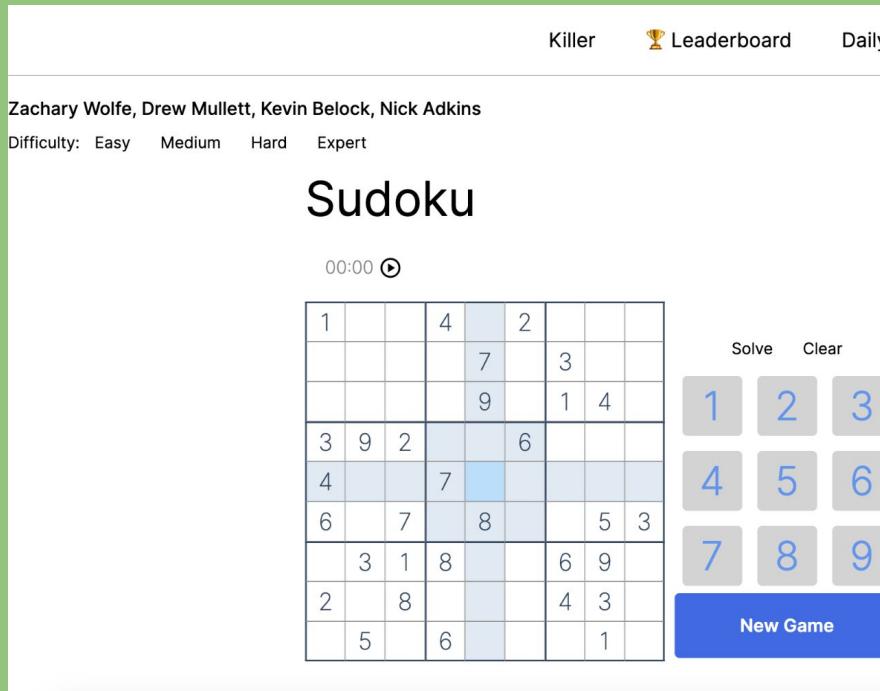
 ZacharyWolfe commented last month

I need this board to be aligned on the center of the screen (primarily inside of the blue container)



# Zachary Wolfe Contributions - Evidence cont.

- Navbar
- Panel
- New Game
- Bold outlines
- Buttons hover
- Timer start-stop
- Changed whole layout of website to support intuitive UI



# Drew Mullett Contributions

- Added functionality for server to make http GET request along with difficulty options
- Created site segments for sudoku and killer sudoku
- Created Dockerfile for project

# Drew Mullett Contributions - Evidence

The image displays four screenshots arranged in a 2x2 grid, illustrating contributions to two web-based Sudoku games.

**Top Left Screenshot:** A screenshot of the "Killer" game interface from <https://sudoku.drewmullett.net/killer>. It shows a 9x9 grid with some numbers filled in. Below the grid are buttons for "Solve", "Clear", and "New Game".

**Top Right Screenshot:** A GitHub issue titled "How to separate Client and Server data? #29" ([View Issue](#)). The issue was opened by Snaredrumhero and has one comment. The comment discusses separating client and server data, mentioning completed puzzle, leaderboard, and puzzle generation features for the server, and incomplete puzzle and validation for the client. The issue is marked as closed.

**Bottom Left Screenshot:** A screenshot of the "Classic" game interface from <https://sudoku.drewmullett.net/classic>. It shows a 9x9 grid with some numbers filled in. Below the grid are buttons for "Solve" and "Clear".

**Bottom Right Screenshot:** A GitHub pull request titled "Websolver #44" ([View PR](#)). It shows a merge from "webservice" into "main". The pull request was merged by Zachary Wolfe last week. The commit message indicates it adds 7 commits. The pull request has no reviews or assignees.

# Nick Adkins Contributions

- Added function to Solve the sudoku board (Mostly works, but still WIP)
- Completely Reworked the Generation Algorithm to be more random, at the cost of time efficiency (of which is still barely noticeable: a time of 0.2 ms VS ~30 ms to generate a board)
  - I just recently optimized the algorithm from ~30 ms → ~20 ms
- Reworked tile showing system so that the resulting board is always compatible with the solver function
- Closed Issue #2

# Nick Adkins Contributions - Evidence

(Both Generate.tsx & Solver.tsx are now about 200 lines each, can't show all of it here)

```
killer-sudoku > app > Generate.tsx > initBoard
1  /**
2   * @file      Generate.tsx
3   * @author    Nicholas Adkins (na761422@ohio.edu)
4   * @brief     Generates a valid full sudoku board
5   * @date      February 26, 2024
6   */
7
8   import { SpaceButtonProperties, HandleHighlighting, SaveBoardState } from "..."
9   import { solve_str, isAvailable, isValid, makeBoard, copyBoard } from "..."
10
11  /**
12   * @brief Initializes the board to be a 2d array, generates a board full
13   *       data with SpaceButtonProperties, and highlights the origin to
14   *       @param used (WIP)
15   *       @returns A 9x9 board
16   */
17  export function initBoard(used: number): SpaceButtonProperties[][] {
18
19    console.log("initBoard: Start");
20
21    let recNumber: number = 0;
22    let board: string[][] = makeBoard();
23    generate(board);
24
25    function generate(inputs: string[][]): boolean {
26
27      ...
28
29      return true;
30    }
31
32    return board;
33  }
34
```

```
killer-sudoku > app > Solver.tsx > ...
1  /**
2   * @file      Solver.tsx
3   * @author    Nicholas Adkins (na761422@ohio.edu)
4   * @brief     Function to solve a board / determine if it's solvable
5   * @date      March 8, 2024
6   */
7
8   // TODO: Re-do Comments and such
9
10
11  /**
12   * @brief takes input board & tries to solve it
13   * @param board input board of string[][] trying to be solved
14   * @returns tuple of a boolean (did it succeed or not) & the resulting board, or null if failed
15   */
16  export function solve_str(input: string[][]): [boolean, string[][]] {
17    let board: string[][] = copyBoard(input);
18
19    let notes: boolean[][][] = [];
20    for (let x = 0; x < 9; x++) {
21      notes[x] = [];
22      for (let y = 0; y < 9; y++) {
23        if (board[x][y] != '') continue;
24        notes[x][y] = [];
25        for (let n: number = 1; n <= 9; n++) {
26          ...
27        }
28      }
29    }
30
31    let success: boolean = true;
32    for (let x = 0; x < 9; x++) {
33      for (let y = 0; y < 9; y++) {
34        if (board[x][y] == '') {
35          if (notes[x][y].length == 0) {
36            success = false;
37            break;
38          }
39        }
40      }
41    }
42
43    if (!success) return [false, null];
44
45    let result: string[][] = copyBoard(board);
46    for (let x = 0; x < 9; x++) {
47      for (let y = 0; y < 9; y++) {
48        if (board[x][y] == '') {
49          let possibilities: number[] = notes[x][y];
50          if (possibilities.length == 1) {
51            result[x][y] = String(possibilities[0]);
52          }
53        }
54      }
55    }
56
57    return [true, result];
58  }
59
```

# Nick Adkins Contributions - Evidence (cont.)

I Copied over my code into a test file, and logged the time it took to generate the board (using the performance.now() function).

Post-Optimization: I output the average time (ms) of 40 generation attempts, repeated 20 times, into a text file, as seen on the right.

I don't have an image of what the numbers were Pre-Optimization, but they were higher, around 30 ms or so.

*Note: These numbers were from my desktop; it generates a bit slower on my laptop*

1	
2	24.3467625
3	24.5544875
4	26.572747500000002
5	33.740042499999994
6	31.995249999999977
7	24.30849249999999
8	22.4601325
9	24.020654999999994
10	21.196372500000003
11	17.491824999999995
12	22.855915
13	18.117650000000005
14	33.35680999999999
15	21.53533749999994
16	19.0486925
17	24.3334275
18	22.84449999999999
19	23.442057500000004
20	29.643119999999993
21	30.24444250000001
22	20.023084999999988
23	

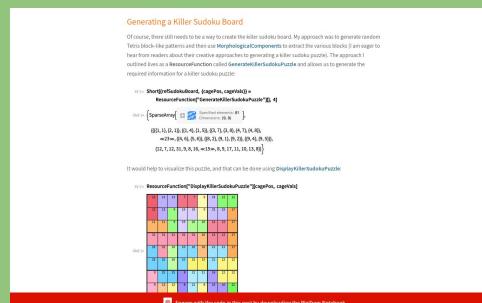
# **Kevin Belock's contributions**

I researched ways to implement the cage / grouping for the killer sudoku side of the board. The hardest thing to make will be implementing the shapes and outlining them properly without any overlap.

# Kevin Belock's contributions evidence

Websites I have visited for research;

- <https://markheath.net/post/typescript-tetris>
- <https://resources.jointjs.com/tutorials/joint/tutorials/ts-shape.html>
- <https://blog.wolfram.com/2020/06/02/using-integer-optimization-to-build-and-solve-sudoku-games-with-the-wolfram-language/>



The screenshot shows a blog post titled "Porting Tetris to JavaScript with TypeScript" by Kevin Belock. The post includes a Tetris game board at the top, followed by several code snippets and explanatory text.

**The HTML**

There wasn't much that needed to be done in the HTML, except to create a `HTML5 canvas` object for us to draw on. Probably there is some cool trick web developers use to pick the optimal size for the game board based on your browser size, but I just went for a fixed size canvas for now.

**The Shape Classes**

In my original Java code I had a `Shape` base class, with methods like `move`, `drop`, `rotate` etc, and a series of classes derived from it

**Custom Shape With TypeScript**

We often get asked how to incorporate TypeScript with JointJS. As JointJS is a standard JavaScript library, the integration process is quite simple and straightforward. In the following tutorial, we are going to create our very own custom shape using TypeScript, and also try to provide you with some useful information along the way.

**A basic shape**

To get started, we create a separate `shape.ts` file. We will define our custom shape here, and later you can import it in your main file. To define our custom shape in TypeScript, we are going to extend the `dia.Element` class. The syntax is quite simple, and will seem familiar to those of you who have used JavaScript classes before. In the following code, you can see our custom element `MyShape`.

```
import { dia } from 'jointjs';

export class MyShape extends dia.Element {
  constructor() {
    super();
    this.$el.attr('class', 'myshape');
  }
}
```

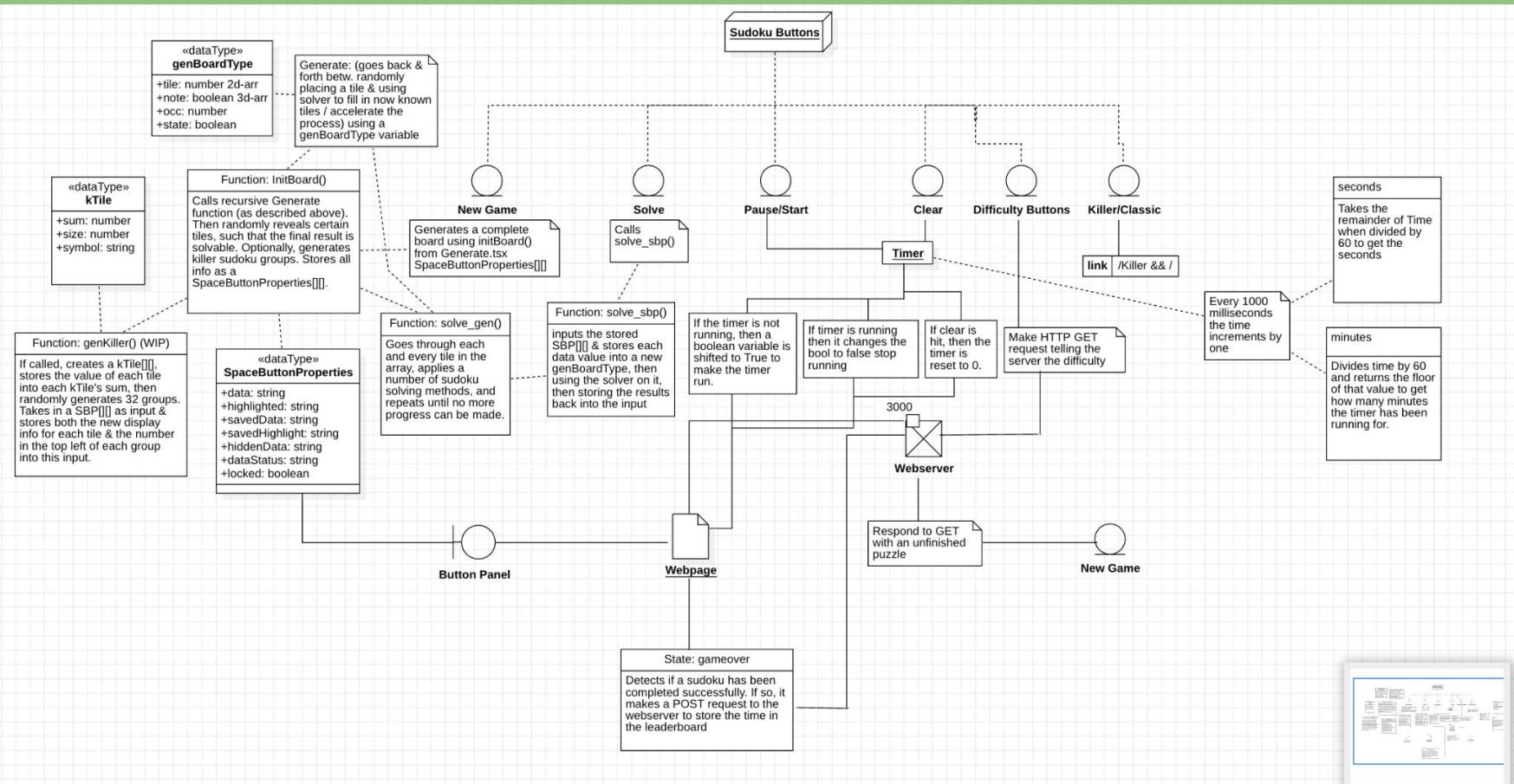
The `defaultts` function will return an object that contains the attributes for our model. It is possible to use an object for our `defaults`, but as objects are referenced, not copied in JavaScript, our `favorites` will return a different object each time.

In JavaScript classes, you may be used to working with `super`. In our use case, we want our child subtype to take attributes from its parent type. Using `...super.defaults`, if an attribute is undefined in the child, the parent attribute will be assigned instead. Similarly, once a property is set in the child, additional values of the same property from the parent are replaced.

This is the most basic boilerplate for a custom shape, but you have to agree it's not very exciting, so let's add some more attributes.

# Start of Checkpoint 4

# UML Diagram



# Meeting Attendance

- Group Meeting (Checkpoint 4) 3/30/2024
  - Present: All members
- Group Meeting 4/2/2024
  - Present: All members

# Work Summary

## Zachary

- Successfully made cages for individual cells
- Added support for a sum in the top left of cells
- Closed issues #66, #65, #60, #59
- Hacky getaround for bolded border outline
- Added Game Ending logic

## Drew

- Added eslint static analysis to current build by fixing dependency errors
- Added Webserver section to group .mdj display

## Nick

- Reworked the generation Algorithm again (now its much faster)
- Implemented Killer Sudoku group generation & helped with the UI aspects of it.
- Posted a ton of bugs on GitHub as issues, and also closed issue #62

## Kevin

- Working on finding the average ratio of killer sudoku grouping.
-

# Next Checkpoint Plan

## Zachary

- Fix minor bugs noticed along the way to alpha testing
- Add extra features like leaderboard
- Explore authentication and storing user data
- Implement a victory popup overlay?
- Save time, name, and which sudoku puzzle to leaderboard scores

## Drew

- Try to regroup after BlossomHack and figure out where my efforts should best be put
- Fix Dockerfile installation errors
- Clean up git branches with file hangs preventing checkouts between branches

## Nick

- Make the solver work better with killer sudoku
- Help implement Difficulty options/menu, and help merge the difficulty with killer sudoku mode.
- Work on fixing some bugs we just discovered in the live demo showcase today

## Kevin

- Helping with fixing any issues in grouping and other minor issues.

# Road Blocks

Zachary

- Time to manage 3 HWs, 3 Quizzes, and the project on top of all other classes assignments

Drew

- Difficulty managing BlossomHack along with increased workload
- Strange errors with Docker installation preventing debugging Dockerfile

Nick

- Likewise, am having difficulty finding the time to work on this with all of the other assignments i have elsewhere

Kevin

- Time to work with all other class stuff

# Grading Items

# Zachary Wolfe Contributions

- Successfully made cages for individual cells
- Added support for a sum in the top left of cells
- Closed issues #66, #61, #60, #59
- Hacky getaround for bolded border outline
- Changed directory pathing to display Killer Sudoku on the home page instead of Sudoku
- Fixed UI sizing for QOL changes
- Fixed persistent panel input bug
- Added Game Ending logic

# Zachary Wolfe Contributions - Evidence

The image shows a GitHub pull request interface with two main comments from user 'ZacharyWolfe'.

**Comment 1:** ZacharyWolfe commented 2 days ago

Just needed to set the icon, good catch! Solve with [#67](#)

**Comment 2:** ZacharyWolfe commented 4 days ago • edited

These issues were all fixed in my branch a while back, just needed to merge.

**Closing:** ZacharyWolfe closed this as completed 2 days ago

**Comment 3:** ZacharyWolfe commented 4 days ago

This was a known bug that was fixed a long time ago, been waiting to merge because I needed to fix the panel more.

**Closing:** ZacharyWolfe closed this as completed 4 days ago

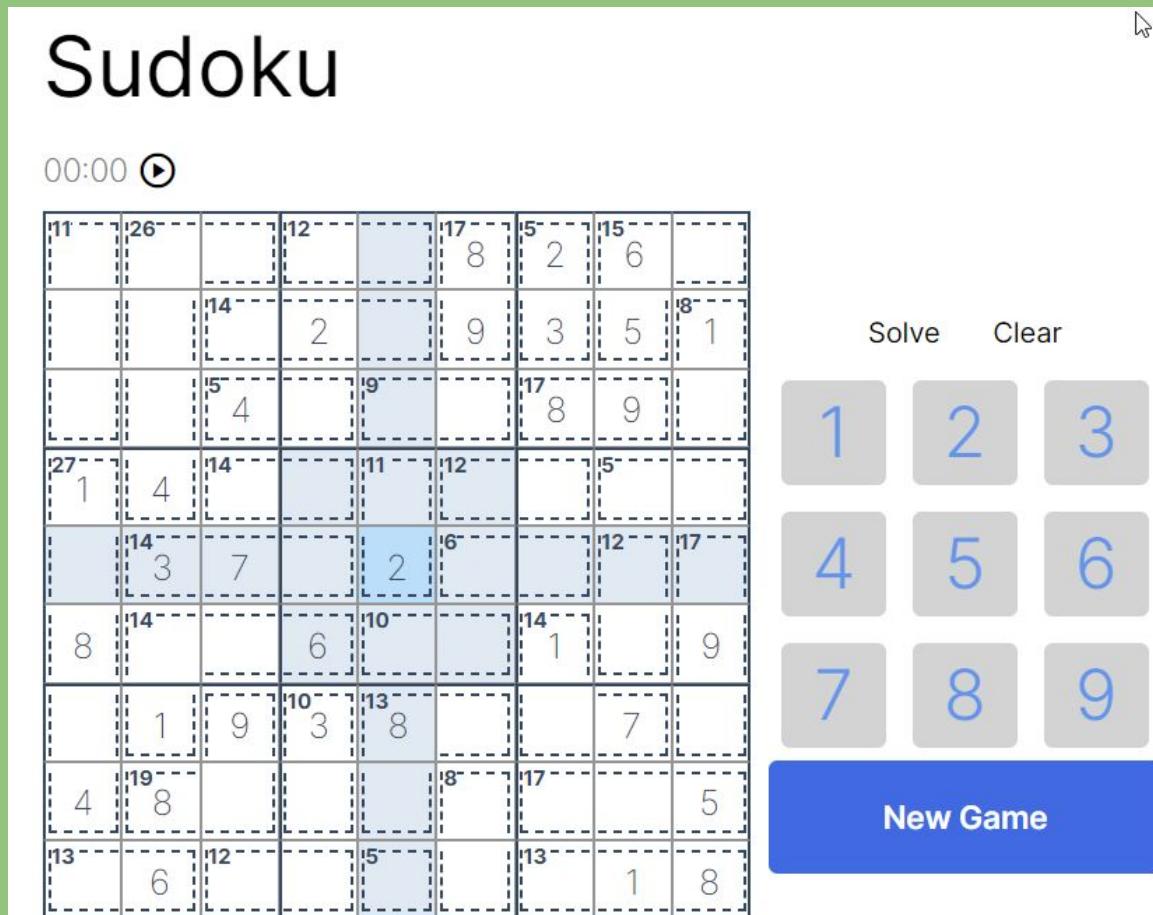
# Zachary Wolfe Contributions - Evidence cont.

The screenshot shows a GitHub pull request thread with the following contributions from Zachary Wolfe:

- ZacharyWolfe commented 4 days ago • edited**  
To comment on what's happening in the middle, it appears that there is a space in the (4, 3) cell, so when it's comparing it's actually bringing up false. I can't reproduce the (4, 3) cell 'space' issue but I will put out a fix for the highlighting. Good catch!
- ZacharyWolfe commented 4 days ago**  
Will ensure a fix when [#62](#) is closed.
- ZacharyWolfe mentioned this issue 2 days ago**  
Fix Solve and Clear highlights [#65](#)  
↳ Merged
- ZacharyWolfe commented 2 days ago**  
Solved with [#65](#)
- ZacharyWolfe closed this as completed 2 days ago**

# Zachary Wolfe Contributions - Evidence cont.

- All the visual aspects in this photo



# Drew Mullett Contributions

- Added eslint static analysis to current build by fixing dependency errors
- Added Webserver section to group .mdj display
- Attempted to sort through Docker dependencies
- Added unit tests for timer object

```
andrewmullett@DESKTOP-K4EDPIH:~/s24-killer-sudoku/killer-sudoku$ npm test

> killer-sudoku@0.1.0 test
> jest

PASS  app/timer.test.tsx
  ✓ two plus two is four (2 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        2.841 s, estimated 3 s
Ran all test suites.

andrewmullett@DESKTOP-K4EDPIH:~/s24-killer-sudoku/killer-sudoku$ |
```

# Drew Mullett Contributions - Evidence

Added eslint dependencies and fixes #70

[Open](#) Snaredrumhero wants to merge 1 commit into `generation` from `generation-fix`

Conversation 0 Commits 1 Checks 0 Files changed 7

Snaredrumhero commented yesterday

The dependencies wouldn't be added in because of some git weirdness that was happening with missing files being tracked but weren't able to be found. You'll still need to run this in `s24-killer-sudoku` to get eslint's dependencies installed

```
npm install --save-dev eslint typescript @typescript-eslint/parser @typescript-eslint/eslint-plugin
```

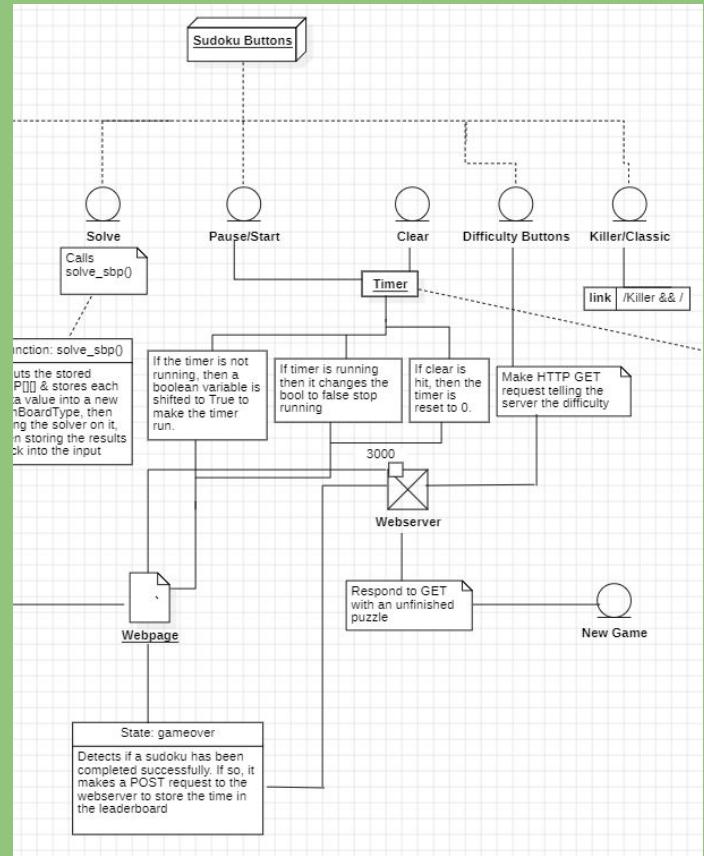
After that just go in the `killer-sudoku/app` directory and run

```
npx eslint .
```

And it'll perform static analysis. I merged because there were some simple variable assignment issues it ran into, so I figured I'd just pass those your way since you were having issues with eslint. Let me know if you have any questions

Added eslint dependencies and fixes

Snaredrumhero requested a review from nickadkins47 yesterday



# Nick Adkins Contributions (1/3)

Reworked the generation Algorithm again (now it's much faster)

- Board generation process is now down from ~20 ms (with a rough maximum value of ~200 ms) down to <1 ms (maxing out at roughly ~10 ms)
- Part of the new algorithm pictured on the right, but i'd definitely insist on looking through every function to get the full picture here on how it works.
- Compared to previous version, its largely organized in a similar manner but just that significant changes were made to optimize it.

```
let iter: number = 0;

const generate = (board: genBoardType): boolean => {
    if (iter++ > 50) {iter = 0; return true;}

    //Calls solver & records all changes it made
    const changes: [number,number][] = solve_gen(board,2);
    if (board.state) {
        if (board.occ == 81) return true;

        let x: number = 0, y: number = 0;
        do {
            x = rand(0,8);
            y = rand(0,8);
        } while (board.tile[x][y] != 0);

        //Look through every available option
        for (let val of randomOptions(board.note[x][y])) {
            boardAdd(board,val,x,y);
            if (generate(board)) return true;
            boardRem(board,x,y);
        }
    }

    //If board is unsolvable, undo all solver changes & return false
    for (let ch of changes) boardRem(board,ch[0],ch[1]);
    board.state = true;
    return false;
}
```

# Nick Adkins Contributions (2/3)

Implemented Killer Sudoku group generation & helped with the UI aspects of it.

- Also including a way to turn on/off killer group generation in the initBoard() function (still kinda WIP)
- Implemented in GenKiller.tsx, including most of the killer sudoku board generation, and a ton of documentation too, with a little bit elsewhere

```
GenKiller.tsx
killer-sudoku > app > GenKiller.tsx > genKiller
20  * will have the same symbol attached to it)
21  * @note Elsewhere, I take advantage of the fact that objects are always
22  * passed by reference in typescript, so that I have multiple spots
23  * in a kTile[][] point to the same kTile, thus not needing to worry about
24  * updating all tiles, since those spots would just point to the same object
25 */
26 type kTile = {
27   sum: number,
28   size: number,
29   symbol: string
30 };
31 /**
32  * @brief Creates groups for killer sudoku based on the given board
33  * @param {SpaceButtonProperties[][]} board board to write dashedBorder & topLeftNumber values to
34  * @return None
35 */
36 export function genKiller(board: SpaceButtonProperties[][]): void {
37   //# of killer groups (vary on difficulty?)
38   const AmountTotal: number = 32;
39
40   let groups: kTile[][] = [];
41   for (let i = 0; i < 9; i++) {
42     groups[i] = [];
43     for (let j = 0; j < 9; j++) {
44       groups[i][j] = {sum: Number(board[i][j].data), size: 1, symbol: '.'};
45     }
46   }
47 }
```

# Nick Adkins Contributions (3/3)

Posted a ton of bugs on GitHub as issues, and also closed issue #62, as Pictured right & below

- New Game -> Pause broken / already ticking**  
#66 by nickadkins47 was closed 3 days ago
- When putting in a number, it doesnt get rid of the blank space that was there** bug good first issue ui
- Incorrectly Place Tile -> Solve Button -> Tile highlighting is still red on fixed tiles** bug ui
- Big number Buttons fill every unlocked tile with that number** bug ui
- Pause button bugs** bug ui  
#59 by nickadkins47 was closed 4 days ago 2 tasks done
- Solve button doesnt work properly if there are incorrectly placed tiles**  
#55 by nickadkins47 was closed 5 days ago

The screenshot shows a GitHub issue page for a bug titled "When putting in a number, it doesn't get rid of the blank space that was there" (issue #62). The issue is marked as "Closed" with a purple checkmark icon. The author, "nickadkins47", opened the issue 5 days ago and has 3 comments. In the comments section, "nickadkins47" commented 4 days ago: "ok with the latest changes in (either spaces-board or generation branch), this is now fixed". Below this comment, "nickadkins47" closed the issue as "completed" 4 days ago. The GitHub interface includes standard navigation elements like "Member", "Author", and a three-dot menu.

Closed

**When putting in a number, it doesn't get rid of the blank space that was there** #62  
nickadkins47 opened this issue 5 days ago · 3 comments

**nickadkins47** commented 4 days ago

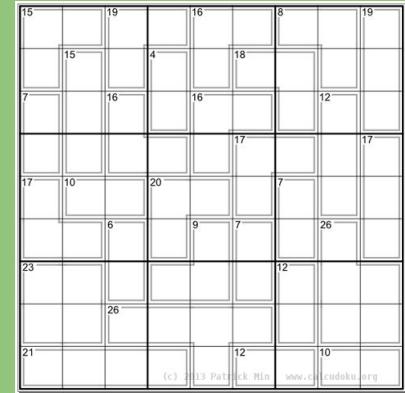
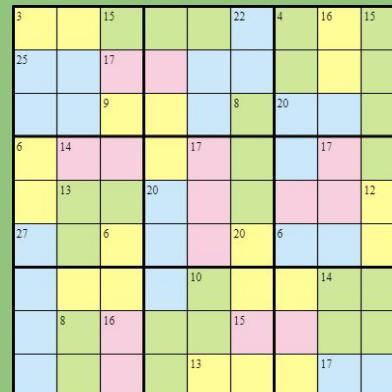
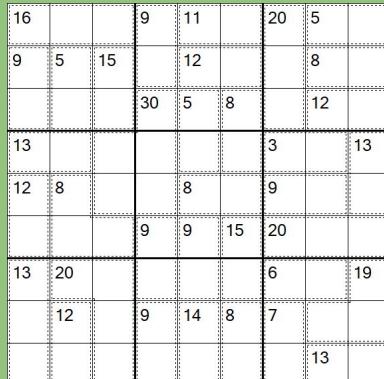
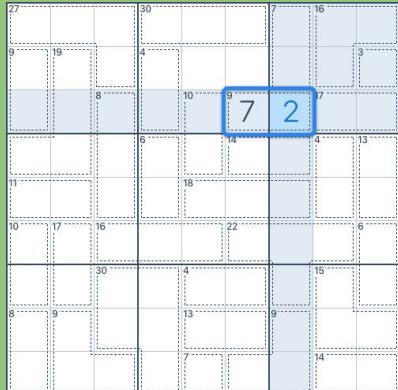
ok with the latest changes in (either spaces-board or generation branch), this is now fixed

**nickadkins47** closed this as completed 4 days ago

# **Kevin Belock Contributions**

Researched what the most common cell grouping in a sudoku board and what cell groupings are generated when the difficulty is changed. I have found that 2 and 3 group cages are more common than 1 and 4 group cages. A 5 group cage is rarely generated.

# Kevin Belock Contributions - Evidence



# Start of Checkpoint 5

# Meeting Attendance

- Group Meeting 4/9/2024
  - Present: All members
- Group Meeting 4/13/2024
  - Present: All members

# Work Summary

## Zachary

- Fixed errors revolving around the pause button wiping the board
- Added .babelrc, babelconfig, Jest config, and dependencies for Jest.

## Drew

- Added dependencies and example for testing software Jest
- Added dependencies and example for documentation software TSDoc
- Wrote tests for generating under different difficulties

## Nick

- Worked on the Solver Algorithm to be much faster, still not as good as it needs to be though, but im happy with my work.
- Helped implement Jest testing files
- Added a target to the makefile to call the jest testing

## Kevin

- Added my unit test cases for the timer using jest.
- Got mocked timers working through jest and learned how to advance the fake timers with a function call through jest.

# Road Blocks

Zachary

- Will need redux for Auth/Storing user data, will not be able to implement the Leaderboard without it. Very unlikely that we get to the user data part of this project by the end of the year.

Drew

- Surprising lack of documentation on receiving HTTP requests in React, so any kind of leaderboard would be difficult to send data to

Nick

- Solver Algorithm is still kinda slow / might need to use an entirely different way of doing it.

Kevin

- Road block we ran into was Jest not working with react and needing to write separate tests code for the tests to run.

# Reflection - Zachary

## Goals from week 5 (Checkpoint 1)

- I can confidently say I learned Typescript
- I can confidently say I learned how to work with React
- I can confidently say I learned how to write a web-app with HTML5 and Tailwind CSS
- I can confidently say I strengthened interpersonal connections through teamwork
- I can confidently say I learned to build modern tech stack skills (React, Next.js, Node.js)
- I can confidently say I learned how to build an intuitive and snappy UI.
- I can confidently say I learned how to work with a team on a large project

Throughout the course of the semester, I had the opportunity to take my skills to a new level. I took full advantage of this opportunity, and exercised it all the way until the end. I overcame challenges nested in React, HTML, CSS, page re-routing, state variables, and understanding the process of user workflow. Additionally, I got to do something I always wanted to, work with a team on a project we were all excited to work on. Our team started off extremely strong and persistently worked up until the very end. Coming into the project, we set high goals for this app like a Leaderboard, Authentication, and mistakes a user made; we later realized that we hadn't the time and couldn't complete those additional features we so desired. Although the semester has ended I still would like to implement these features as it was part of the incoming goal. Had we more time in a perfect world, this project could have been slammed out of the park without a doubt.

# Reflection - Drew (Goals/tools)

## Goals from week 5 (Checkpoint 1)

- Learned how to operate and maintain a webserver
- Greatly improved Git proficiency in the terminal
- Gained familiarity with Typescript and npm tools
- Improved my ability to coordinate and work on a team
- Gained proficiency with Ubuntu terminal

I'm incredibly happy with how this project turned out. I got great experience with coordinating with a team, especially when coordinating the dependencies needed for the project to function. As we got further along the project, testing project functionality became more and more important. I wish that we could have gotten Docker integrated into the project from the start because that would have made checking branches before merging much easier. Npm was a solid tool to use for installing our dependencies because it worked quickly and was generally easy to troubleshoot, but there were still some pretty glaring issues with its use, especially when trying to install fixes for vulnerable packages. Jest also performed its job well, but its inability to work with react without extra downloads was difficult to troubleshoot and implement, resulting in us having to use some hacky solutions to get everything in on time. TSDoc is far and away the tool I most enjoyed using. The documentation was very easy to traverse and it did a great job emulating Doxygen.

# Reflection - Drew (Teamwork)

I'm really happy with the work of my teammates, everyone was firing on all cylinders and it reflects in our final product.

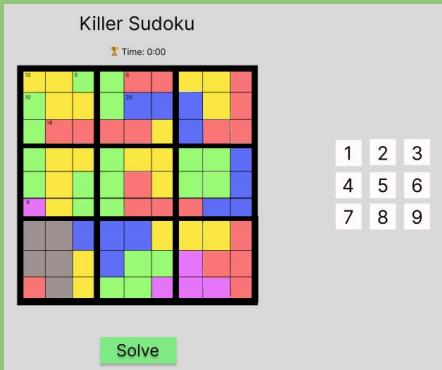
- Got started almost immediately
- Had a decently wide knowledge background and support network
- Great communication through Git Issues, Discord, and Pull Requests

In the future:

- Reserve the more beginner friendly tasks. It was difficult to try and merge with the React team after working on the server for two weeks
- Have large or all hands meetings for plannings and implementation can spread out from there to work on individual tasks
  - Everyone is on the same page, the view for the project is more singular, and we get multiple perspectives on how to solve a problem

# Reflection - Drew (Semester Contribution)

- Configured our webserver
- Collaborated with Zach for our Fibble prototype
- Interviewed IT professional for codebase inspiration
- Added network section to our UML diagram
- Installed static analysis, documentation, and test dependencies
- Created static analysis, documentation, and test examples



KillerSudoku  
00:00

Zachary Wolfe, Drew Mullett, Kevin Belock, Nick Adkins

Difficulty: Easy Medium Hard Expert

Solve Clear

New Game

Killer Sudoku

# Reflection - Nick

## Goals from week 5 (Checkpoint 1)

- Never touched much on React, but i most definitely did learn Typescript
- Learned a bit about various different tools used for these nodejs projects, such as Jest
- Gained quite a bit of experience working with teams

Overall i'm actually quite happy with how everything turned out. Even though i ended up with 99% of my work concentrated in a very specific part of the code, i still genuinely enjoyed working on the project. It sucks that i didn't have the time to really sit down & perfect my solving algorithm, but i'm still satisfied with the end result. In fact, i may end up working more on it over the summer.

# Reflection - Kevin

- I learned some of how to use typescript and react
- Learned how to help team members even if it is a small help
- Noticed That I could have done more when it came to helping with more of the coding then what I have already
- Enjoyed working with the team.

Throughout the project I was constantly fighting with myself to do anything helpful for the team, I was able to implement my timer code without too many issues and learned how to export it for it to display on the page properly.

I should of done more when it came to the actual coding but I did do my best in helping research what parts of the killer cages needed to be generated, how big they got, and how rarely they were updated. I enjoyed working with this team for the project and would happily work with them again if I get the chance. They were all very helpful in learning how to use typescript, react, and designing the unit tests.

# Grading Items

# Release

[https://github.com/OU-CS3560/s24-killer-sudoku/releases  
/tag/release](https://github.com/OU-CS3560/s24-killer-sudoku/releases/tag/release)

# Testing

Zachary

- I used Jest to test, it was simple, had little to no complication. It also allowed me to write tests quickly.
- I wrote these tests to ensure that the highlighting was choosing the correct squares to start the algorithm. As well as to test that the HideBoard function was correctly working because there were lots of issues around pausing the game.
- I chose the input of a random space on the board to find where the algorithm would start. I chose the input to HideBoard to be the initBoard algorithm because I wanted to simulate a real game.

**All testing on next page**

# Testing

The screenshot shows a terminal window in VS Code displaying the output of a Jest test run. The terminal tab is titled 'testOutput.txt' and contains the following log entries:

```
1
2 > killer-sudoku@0.1.0 test
3 > jest
4
5   console.log
6     initBoard: Start
7
8       at log (app/testing/initBoard.ts:22:13)
9
10  console.log
11    initBoard: Randomization complete
12
13    at log (app/testing/initBoard.ts:60:13)
14
15  console.log
```

Below the terminal, the status bar shows the command: `killer-sudoku git:(spaces-board) x npm test &> testOutput.txt`. The bottom left corner of the terminal shows a yellow progress bar.

VS Code interface elements include tabs for 'Sudoku.test.tsx', 'Sudoku.tsx M', 'testOutput.txt', and 'page.tsx'. The bottom navigation bar includes 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', 'PORTS', and 'GITLENS'. The bottom right corner features a terminal icon with a yellow progress bar.

# Testing Cont.

# Documentation

Zachary

- I used TSDoc, it was simple and was built by Microsoft specifically to support Typescript documentation.

The screenshot shows the TSDoc playground interface. On the left is a code editor containing a TypeScript file named `HandleHighlighting.ts`. The code defines a function `HandleHighlighting` that handles row and column highlights as well as 3x3 highlighting. The function takes three parameters: `row`, `col`, and `newBoard`. The code uses nested loops to iterate through the board and update highlight states. On the right is a summary panel titled "Summary" which provides a brief description of the function and its parameters. Below the summary are tables for "Parameters" and "Properties".

```
1  /**
2   * @brief A function that handles row and column highlights as well as 3x3 highlighting
3   * @param row - the row of the cell that was clicked
4   * @param col - the column of the cell that was clicked
5   * @param newBoard - the board we want to change
6   */
7  export function HandleHighlighting(row: number, col: number, newBoard: SpaceButton[][]) {
8      if (!isPaused) {
9          try {
10              // Clear the board of highlights
11              for (let j = 0; j < 9; j++) {
12                  for (let k = 0; k < 9; k++) {
13                      if (newBoard[j][k].highlighted !== 'spaceNumberTaken') {
14                          newBoard[j][k].previousHighlight = newBoard[j][k].highlighted;
15                          newBoard[j][k].highlighted = 'space';
16                          console.log(`j: ${j}, k: ${k} highlighted with ${newBoard[j][k].highlighted}`);
17                      }
18                  }
19                  if (newBoard[j][k].marked) {
20                      newBoard[j][k].marked = false;
21                  }
22              }
23          } catch (error) {
24              console.error(error);
25          }
26      }
27  }
```

Name	Description
row	the row of the cell that was clicked
col	the column of the cell that was clicked
newBoard	the board we want to change

Summary  
A function that handles row and column highlights as well as 3x3 highlighting

Name	Description
row	the row of the cell that was clicked
col	the column of the cell that was clicked
newBoard	the board we want to change

© 2024 Microsoft

# Testing - Drew

Drew

- I used Jest to test due to its popularity with JavaScript and Typescript developers and easy to read documentation
- I had initial issues with the configuration file that the documentation recommended. We had to adjust our files since getting Jest to work with Typescript was too difficult with the time we had left
- I wrote tests to ensure that the generate function would reveal the correct number of spaces. In our released code, we have it so that the solver is called on the revealed numbers to see if the solver would be able to work for the user. However, this means that loading harder difficulties of our App would be slow to impossible, so I created this function to better understand what would happen if we took out that checker

**All testing on next page**

# Testing - Drew

```
PASS app/testing/Difficulty.test.tsx
```

```
● Console

  console.log
    initBoard: Start
      at log (app/testing/GenerateDifficulty.ts:21:13)
  console.log
    initBoard: Randomization complete
      at log (app/testing/GenerateDifficulty.ts:58:13)
  console.log
    initBoard: Difficulty: Medium. numShown: 31
      at log (app/testing/GenerateDifficulty.ts:83:13)
  console.log
    initBoard: Tile showing complete
      at log (app/testing/GenerateDifficulty.ts:103:13)
  console.log
    initBoard: Initialization complete
      at log (app/testing/GenerateDifficulty.ts:127:13)
  console.log
    save board state
      at log (app/SudokuFuncs.ts:192:13)
  console.log
    initBoard: Start
      at log (app/testing/GenerateDifficulty.ts:21:13)
  console.log
    initBoard: Randomization complete
      at log (app/testing/GenerateDifficulty.ts:58:13)
  console.log
    initBoard: Difficulty: K-Expert. numShown: 0
      at log (app/testing/GenerateDifficulty.ts:83:13)
  console.log
    initBoard: Tile showing complete
      at log (app/testing/GenerateDifficulty.ts:103:13)
  console.log
    initBoard: Initialization complete
      at log (app/testing/GenerateDifficulty.ts:127:13)
  console.log
    save board state
      at log (app/SudokuFuncs.ts:192:13)
  console.log
    initBoard: Start
      at log (app/testing/GenerateDifficulty.ts:21:13)
  console.log
    initBoard: Randomization complete
      at log (app/testing/GenerateDifficulty.ts:58:13)
  console.log
    initBoard: Difficulty: Gibberish. numShown: 81
      at log (app/testing/GenerateDifficulty.ts:83:13)
  console.log
    initBoard: Tile showing complete
      at log (app/testing/GenerateDifficulty.ts:103:13)
  console.log
    initBoard: Initialization complete
      at log (app/testing/GenerateDifficulty.ts:127:13)
  console.log
    save board state
      at log (app/SudokuFuncs.ts:192:13)

Test Suites: 3 passed, 3 total
Tests:    101 passed, 101 total
Snapshots: 0 total
Time:    0.421 s, estimated 1 s
Ran all test suites.
andrewmullett@DESKTOP-K4EDPIH:~/s24-killer-sudoku/killer-sudoku$ |
```

```
console.log
  save board state
  at log (app/SudokuFuncs.ts:192:13)

  console.log
    initBoard: Start
    at log (app/testing/GenerateDifficulty.ts:21:13)

  console.log
    initBoard: Randomization complete
    at log (app/testing/GenerateDifficulty.ts:58:13)

  console.log
    initBoard: Difficulty: Medium. numShown: 31
    at log (app/testing/GenerateDifficulty.ts:83:13)

  console.log
    initBoard: Tile showing complete
    at log (app/testing/GenerateDifficulty.ts:103:13)

  console.log
    initBoard: Initialization complete
    at log (app/testing/GenerateDifficulty.ts:127:13)

  console.log
    save board state
    at log (app/SudokuFuncs.ts:192:13)

Test Suites: 3 passed, 3 total
Tests:    101 passed, 101 total
Snapshots: 0 total
Time:    0.421 s, estimated 1 s
Ran all test suites.
andrewmullett@DESKTOP-K4EDPIH:~/s24-killer-sudoku/killer-sudoku$ |
```

```
describe("Test board generation", () => {
  test("Check if correct number of tiles are revealed for a regular sudoku puzzle", () => {
    const killer: boolean = false;
    const used: number = 0;
    const difficulty: string = "Medium";
    const testboard: SpaceButtonProperties[][] = initBoard(killer, used, difficulty);

    let count = 0;
    for (let i = 0; i < 9; i++) {
      for (let j = 0; j < 9; j++) {
        if (testboard[i][j].data != "") {
          count++;
        }
      }
    }
    expect(count).toBe(31);
  });

  test("Check for incorrect difficulty", () => {
    const killer: boolean = false;
    const used: number = 0;
    const difficulty: string = "Gibberish";
    const testboard: SpaceButtonProperties[][] = initBoard(killer, used, difficulty);

    let count = 0;
    for (let i = 0; i < 9; i++) {
      for (let j = 0; j < 9; j++) {
        if (testboard[i][j].data != "") {
          count++;
        }
      }
    }
    expect(count).toBe(81);
  });

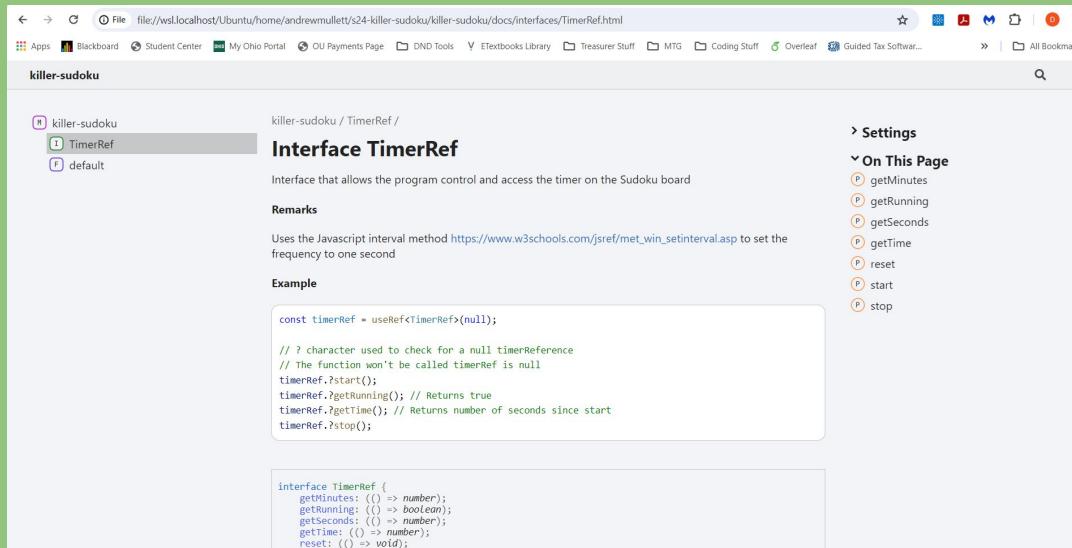
  test("Test for high difficulty", ()=>{
    const killer: boolean = false;
    const used: number = 0;
    const difficulty: string = "K-Expert";
    const testboard: SpaceButtonProperties[][] = initBoard(killer, used, difficulty);

    let count = 0;
    for (let i = 0; i < 9; i++) {
      for (let j = 0; j < 9; j++) {
        if (testboard[i][j].data != "") {
          count++;
        }
      }
    }
    expect(count).toBe(0);
  });
});
```

# Documentation

Drew

- I used TSDoc. It had amazing documentation and worked very similarly to Doxygen which is why we chose it. I wrote documentation for the TimerRef interface, giving an overview of the functions inside



# Testing - Nick

- As mentioned, i helped implement the Testing stuff & getting it to work alongside our main files, (which was somehow surprisingly difficult)
- Once i got it working, i went ahead and implemented testing for:
  - A: The main generation algorithm, making sure all appropriate values get modified whenever a tile is added to a genBoard object.
  - B: Killer generation, to make sure that all resulting killer groups all are in the valid range of sizes & max sizes (between 1 & 5 inclusive).
  - C: The solver, to make sure that each of the solver methods are correctly called whenever the circumstances are right for it.

**All testing on next 2 pages**

# Testing - Nick (Cont.)

```
//Does adding a tile to a genBoard  
// change every internal value accordingly?  
describe("When Adding to a genBoard tile", ()=>{  
    let board = new genBoard;  
    board.add(1,4,4);  
  
    test("tile gets updated", ()=>{  
        expect(board.tile[4][4]).toBe(1);  
    });  
    test("note in tile is turned off", ()=>{  
        expect(board.note[4][4][1]).toBe(false);  
    });  
    test("occ is now incremented", ()=>{  
        expect(board.occ).toBe(1);  
    });  
    describe("notesPerTile", ()=>{  
        test("...is now 0 for this tile", ()=>{  
            expect(board.notesPerTile[4][4]).toBe(0);  
        });  
    });  
});
```

```
[3,2,1,9,7,5,4,8,6],  
[4,7,8,2,6,1,9,3,5],  
[6,5,9,3,8,4,7,2,1]  
];  
let kBoard: kTile[][] = genKiller(input);  
test("1 <= kBoard size <= 5", ()=>{  
    for (let row of kBoard) {  
        for (let val of row) {  
            expect(val.curSize).toBeGreaterThanOrEqual(1);  
            expect(val.curSize).toBeLessThanOrEqual(5);  
            expect(val.maxSize).toBeGreaterThanOrEqual(1);  
            expect(val.maxSize).toBeLessThanOrEqual(5);  
        }  
    }  
});
```

# Testing - Nick (Cont.)

```
    board.add(1,0,0);
    board.add(2,1,0);
    board.add(3,2,0);
    board.add(4,2,1);
    board.add(5,2,2);
    board.add(6,1,2);
    board.add(7,0,2);
    board.add(8,0,1);
    solve_gen(board); //Should solve 9@(1,1)
    expect(board.tile[1][1]).toBe(9);
});
test("Method 2 Works", ()=>{
    let board = new genBoard;
    board.add(2,0,0);
    board.add(2,3,1);
    board.add(2,6,5);
    board.add(2,8,7);
    solve_gen(board); //Should solve 2@(7,2)
    expect(board.tile[7][2]).toBe(2);
});
```

```
at log (app/SudokuFuncs.ts:150:25)
```

```
console.log
  save board state
```

```
at log (app/SudokuFuncs.ts:192:13)
```

```
console.log
  hide board
```

```
at log (app/SudokuFuncs.ts:182:13)
```

```
PASS app/testing/Generation.test.tsx (6.425 s)
PASS app/testing/Timer.test.tsx (14.395 s)
```

Test Suites: 4 passed, 4 total

Tests: 104 passed, 104 total

Snapshots: 0 total

Time: 15.167 s

Ran all test suites.

```
nickadkins47@TheTaylors:~/CS3560/s24-killer-sudoku$ █
```

# Documentation - Nick

I also used TSDoc. As mentioned earlier its very similar to Doxygen, and its easy to use & implement.

I used <https://microsoft.github.io/tsdoc/#> to show what it would look like.

The screenshot shows the Microsoft TSDoc interface. On the left is a code editor with the following TypeScript code:

```
1  /**
2   * @info solves the board: copies data values onto a genBoardType, solves that, then converts back
3   * @param {SpaceButtonProperties[][]} boardSBP input board to be solved
4   * @returns {void} None (input is passed by reference)
5  */
6 export function solve_sbp(boardSBP: SpaceButtonProperties[][]): void {
7     let board = new genBoard;
8     for (let x = 0; x < 9; x++) {
9         for (let y = 0; y < 9; y++) {
10            const tile = boardSBP[x][y];
11            //also fixes incorrect tiles so solver works properly
12            if (tile.data == tile.hiddenData) {
13                board.add(toNum(tile.data),x,y);
14            }
15        }
16    }
17    solve_gen(board); // Uses the solve function in Solver.tsx
18    for (let x = 0; x < 9; x++) {
19        for (let y = 0; y < 9; y++) {
20            boardSBP[x][y].data = toStr(board.tile[x][y]);
21        }
22    }
}
```

On the right is the generated documentation. It includes a navigation bar with tabs for HTML, DOM, Lines, AST, and Emitter. The HTML tab is selected, showing the following content:

**Summary**

solves the board: copies data values onto a genBoardType, solves that, then converts back

**Parameters**

Name	Description
boardSBP	input board to be solved

**Return Value**

{void} None (input is passed by reference)



267

PASS app/testing/Timer.test.tsx

# Testing - continued

Kevin

```
8 ✓ test("Checks how many seconds have passed using advance time", () => {
9     jest.useFakeTimers();
10
11     const Time = timer();
12
13     expect(Time.seconds == 0 + 's');
14
15     jest.advanceTimersByTime(1000);
16
17     expect(Time.seconds == 1 + 's');
18
19
20});
```

```
22 test("Checks how many minutes have passed using advance Time", () => {
23
24     jest.useFakeTimers();
25
26     const Time = timer();
27
28     expect(Time.minutes == 0 + ' minutes');
29
30     jest.advanceTimersByTime(60000);
31
32     expect(Time.minutes == 1 + ' minutes');
33
34
35});
```

```
test("Checks if it resets correctly", () => [
    jest.useFakeTimers();

    const Time = timer();

    expect(Time.seconds == 0 + 's');
    expect(Time.minutes == 0 + ' minutes');

    jest.advanceTimersByTime(63000);

    expect(Time.minutes == 1 + ' minutes');
    expect(Time.seconds == 3 + 's');

    Time.reset = true;

    expect(Time.minutes == 0 + ' minutes');
    expect(Time.seconds == 0 + 's');

]);
```

# Documentation - Kevin

- I used TSDoc to do my documentation of my code, it was simple to set up and use

The screenshot shows the TSDoc playground interface. On the left, there is a code editor with the following TypeScript code:

```
1  |
2  |
3  /**
4   * @brief A function that returns a timer and all of its corresponding functions
5   * @returns The timers seconds and minuets to be checked by the unit tests
6   */
7
8  export function timer() {
9
10 /**
11  * @brief Starts the timer when it is first initiated.
12  */
13
14  let timeStart = new Date().getTime();
15  return {
16
17    /**
18     * @brief Returns the seconds the timer has counted since it started
19     * @returns Timer seconds calculated from miliseconds
20     *
```

On the right, the generated documentation is displayed under the "HTML" tab:

- Summary**  
A function that returns a timer and all of its corresponding functions
- Return Value**  
The timers seconds and minuets to be checked by the unit tests