

Client-Side Performance Test

Dilbar Singh Lamba

1 Introduction

This report provides an analysis of performance tests conducted on a game hosted on GitHub Pages. Initially, the page load time was approximately 1.99 seconds. After implementing optimizations such as lazy loading and image compression, the load time reduced to around 1.37 seconds. This report documents the steps taken for optimization, the resulting improvements, and other inferences based on network performance metrics.

2 Initial Load Performance

The initial performance test, shown in Figure 1, indicated a total page load time of 1.99 seconds. Each asset, including JavaScript files, CSS, and images, was loaded immediately upon page load, which increased the initial load time.

```
Total page load time for https://notdilbarsl.github.io/: 1.99 seconds
Assets with significant load times (over 0.5 seconds):
https://notdilbarsl.github.io/styles.css: 0.16 seconds
https://notdilbarsl.github.io/client/generateGrid.js: 0.14 seconds
https://notdilbarsl.github.io/client/gameState.js: 0.14 seconds
https://notdilbarsl.github.io/client/generatedirection.js: 0.14 seconds
https://notdilbarsl.github.io/client/triggerLosingAnimation.js: 0.14 seconds
https://notdilbarsl.github.io/client/triggerWinningAnimation.js: 0.14 seconds
https://notdilbarsl.github.io/client/chooseNumberOfMines.js: 0.15 seconds
https://notdilbarsl.github.io/client/chooseGridSize.js: 0.21 seconds
https://notdilbarsl.github.io/client/handleCellClick.js: 0.30 seconds
https://notdilbarsl.github.io/client/createGrid.js: 0.30 seconds
https://notdilbarsl.github.io/client/showGrid.js: 0.30 seconds
https://notdilbarsl.github.io/client/movePlayer.js: 0.15 seconds
https://notdilbarsl.github.io/client/generateRandomPath.js: 0.15 seconds
https://notdilbarsl.github.io/client/randomMinePlacement.js: 0.15 seconds
https://notdilbarsl.github.io/client/startGame.js: 0.15 seconds
https://notdilbarsl.github.io/client/checkWin.js: 0.14 seconds
https://notdilbarsl.github.io/client/additionalFunctionality.js: 0.21 seconds
https://notdilbarsl.github.io/client/checkLoss.js: 0.36 seconds
https://notdilbarsl.github.io/client/updateLeaderboard.js: 0.14 seconds
https://notdilbarsl.github.io/client/postScore.js: 0.39 seconds
https://notdilbarsl.github.io/client/calculateNewPosition.js: 0.39 seconds
https://notdilbarsl.github.io/option.webp: 0.77 seconds
https://notdilbarsl.github.io/game.webp: 0.79 seconds
https://notdilbarsl.github.io/favicon.ico: 0.11 seconds
```

Figure 1: Initial Performance Test with 1.99s Load Time

2.1 Load Time of Individual Assets

The following table summarizes the load times of various assets in the initial test:

- **JavaScript Files:** Each JavaScript file took approximately 0.14 to 0.39 seconds to load. Key files include `generateGrid.js` (0.14s), `showGrid.js` (0.30s), and `postScore.js` (0.39s).
- **Images:** The two main image files, `option.webp` and `game.webp`, took around 0.77 and 0.79 seconds, respectively. These large image files significantly contributed to the overall load time.

3 Optimization Techniques Applied

To improve load performance, the following optimizations were implemented:

- **Lazy Loading of Non-Critical Assets:** Non-essential assets were set to load only when required, reducing the initial load burden.
- **Image Compression:** The `option.webp` and `game.webp` images were compressed to reduce their file sizes, thereby decreasing the time required to download these resources.

4 Optimized Load Performance

After applying the optimizations, a subsequent test, shown in Figure 2, revealed a reduced total load time of 1.37 seconds. This reduction indicates that the optimizations effectively decreased the loading time by approximately 0.62 seconds.

```
Total page load time for https://notdilbarsl.github.io/: 1.37 seconds
Load Time of Assets:
https://notdilbarsl.github.io/option.webp: 0.35 seconds
https://notdilbarsl.github.io/game.webp: 0.37 seconds
https://notdilbarsl.github.io/styles.css: 0.36 seconds
https://notdilbarsl.github.io/client/generateGrid.js: 0.36 seconds
https://notdilbarsl.github.io/client/generateDirection.js: 0.36 seconds
https://notdilbarsl.github.io/client/gameState.js: 0.37 seconds
https://notdilbarsl.github.io/client/triggerWinningAnimation.js: 0.37 seconds
https://notdilbarsl.github.io/client/triggerLosingAnimation.js: 0.36 seconds
https://notdilbarsl.github.io/client/chooseGridSize.js: 0.37 seconds
https://notdilbarsl.github.io/client/handleCellClick.js: 0.37 seconds
https://notdilbarsl.github.io/client/chooseNumberOfMines.js: 0.36 seconds
https://notdilbarsl.github.io/client/createGrid.js: 0.36 seconds
https://notdilbarsl.github.io/client/showGrid.js: 0.36 seconds
https://notdilbarsl.github.io/client/movePlayer.js: 0.37 seconds
https://notdilbarsl.github.io/client/generateRandomPath.js: 0.37 seconds
https://notdilbarsl.github.io/client/randomMinePlacement.js: 0.37 seconds
https://notdilbarsl.github.io/client/startGame.js: 0.36 seconds
https://notdilbarsl.github.io/client/checkLoss.js: 0.36 seconds
https://notdilbarsl.github.io/client/checkWin.js: 0.36 seconds
https://notdilbarsl.github.io/client/updateLeaderboard.js: 0.36 seconds
https://notdilbarsl.github.io/client/additionalFunctionality.js: 0.36 seconds
https://notdilbarsl.github.io/client/postScore.js: 0.36 seconds
https://notdilbarsl.github.io/client/calculateNewPosition.js: 0.24 seconds
https://notdilbarsl.github.io/favicon.ico: 0.24 seconds
```

Figure 2: Optimized Performance Test with 1.37s Load Time

4.1 Optimized Load Time of Individual Assets

After optimizations:

- **JavaScript Files:** JavaScript file load times remained similar, however, lazy loading of some files had an impact on the reduced load time due to image compression and lazy loading.
- **Images:** The load times for `option.webp` and `game.webp` significantly decreased due to compression, contributing to the improved performance.

5 Comparison of Initial and Optimized Performance

The optimizations resulted in a noticeable reduction in the total load time:

- **Initial Load Time:** 1.99 seconds
- **Optimized Load Time:** 1.37 seconds
- **Improvement:** Approximately 0.62 seconds, or a 31% reduction in load time.

6 Conclusion

The performance tests highlighted that implementing lazy loading for non-essential assets and compressing large images are effective techniques for reducing load times on GitHub Pages. These optimizations reduced the total load time from 1.99 seconds to 1.37 seconds, improving user experience and performance efficiency. Future improvements could include code splitting for JavaScript files and caching strategies for further load time reduction.

Name	Status	Type	Initiator	Size	Time
notdlbars!.github.io	200	document	Other	1.6 kB	350 ms
option.webp	200	webp	(index).6	80.0 kB	314 ms
game.webp	200	webp	(index).7	73.6 kB	358 ms
styles.css	200	stylesheet	(index).8	2.2 kB	243 ms
generateGrid.js	200	script	(index).87	467 B	356 ms
generateDirection.js	200	script	(index).88	315 B	356 ms
gameState.js	200	script	(index).89	667 B	356 ms
triggerLosingAnimation.js	200	script	(index).90	521 B	253 ms
triggerWinningAnimation.js	200	script	(index).91	477 B	255 ms
chooseGridSize.js	200	script	(index).92	399 B	356 ms
chooseNumberOfMine.js	200	script	(index).93	497 B	255 ms
handleCellClick.js	200	script	(index).94	422 B	253 ms
createGrid.js	200	script	(index).95	678 B	340 ms
showGrid.js	200	script	(index).96	670 B	257 ms
movePlayer.js	200	script	(index).97	523 B	340 ms
randomMinePlacement.js	200	script	(index).98	605 B	355 ms
generateRandomPath.js	200	script	(index).99	624 B	253 ms
startGame.js	200	script	(index).100	899 B	339 ms
checkLoss.js	200	script	(index).101	521 B	339 ms
checkWin.js	200	script	(index).102	555 B	339 ms
additionalFunctionality.js	200	script	(index).103	518 B	253 ms
updateLeaderboard.js	200	script	(index).104	615 B	339 ms
postScore.js	200	script	(index).105	757 B	339 ms
calculateNewPosition.js	200	script	movePlayer.js:6	363 B	242 ms
favicon.ico	404	text/html	Other	5.7 kB	247 ms

Figure 3: Optimized Load Times for Key Assets