‹ Return to "Front-End Web Developer Nanodegree" in the classroom

Classic Arcade Game Clone

REVIEW | CODE REVIEW 12 | HISTORY

Meets Specifications

⭐ 🌟 🎉
CONGRATULATIONS💥

Hello, Udacian 👋🏻

I'm very impressed with the extra mile you went to make your project look amazing 💥
The sweet alert element is my personal favourite final touch.
Your project works perfectly fine 😄

It would be really helpful for me if you could leave a feedback to tell me what is helpful and what to improve from this review 😄

I really enjoyed looking at your project ✔️ It was different from the usual ones I review in every small little detail 💥
Keep learning and stay Udacious 💥

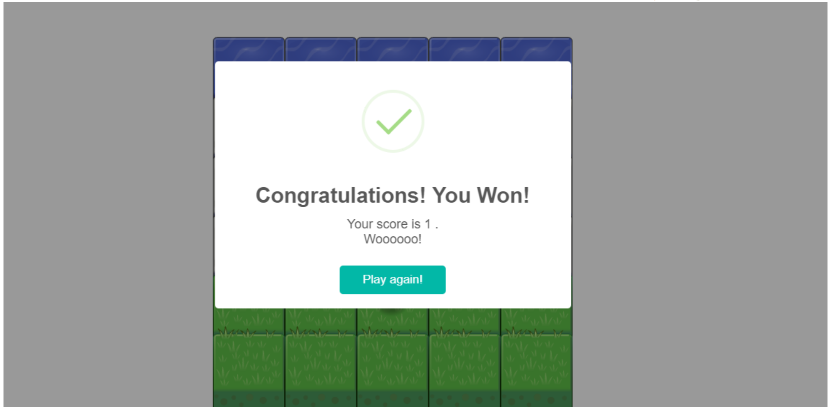
Game Functions

✓ **The game functions correctly and runs error free**

- **Player can not move off screen**
- **Vehicles cross the screen**
- **Vehicle-player collisions happen logically (not too early or too late)**
- **Vehicle-player collision resets the game**
- **Something happens when player wins**

Your player is contained to the screen and your enemies move across the screen ✨
Also,Great work with keeping track of the score 😄
It is these details which make the project stand out!

My personal favourite is using sweet alert to display the winning message ✨ 💫



Congratulations! You Won!

Your score is 1 .
Woooooo!

Object-Oriented Code

✓ **Game objects (player and vehicles) are implemented using JavaScript object-oriented programming features.**

Your code makes good use of prototype functions and follows all object oriented principles 🎉
✨ Amazing job using the concepts of inheritance ✨
Small details like these make the code more better 😄
💡 You might want to add collectables like gems, stars to make the game even more fun 😄
Also increasing the difficulty levels could make the game more challenging.

Documentation

✓ **A `README` file is included detailing all steps required to successfully run the application.**

You have completely fixed the issues as told by your previous reviewers. Good job 😄

✓ **Comments are present and effectively explain longer code procedures. As a rule of thumb: describe what all custom functions and object methods do.**

Good Work including comments above functions as well as inline comments.
I would encourage you to customize the comments to reflect the changes you have made to the code.

Comments are helpful both for the people who will be reading your code (reviewers, collaborators, etc.) but also for your future self. You may want to revisit and even borrow your past code. Good comments will help you remember what you were doing and avoid reinventing your brilliant ideas.

There are three places where comments are usually found:

At the top of the file: this comment should explain what the code does. You can also include your name, contact info, the date etc.
Above functions: these comments should help the reader understand the purpose of the function.
Inline or above lines of code: these comments explain what is happening at this point in the code. If used carefully they can provide an explanatory map for the reader.

✓ **Code is formatted with consistent, logical, and easy-to-read formatting as described in the Udacity JavaScript Style Guide.**

Code is nicely formatted ⭐ 🌟

DOWNLOAD PROJECT

12 CODE REVIEW COMMENTS ›

RETURN TO PATH