UDACITY

PROJECT

Classic Arcade Game Clone

A part of the Front-End Web Developer Nanodegree Program

PROJECT REVIEW

CODE REVIEW 5

NOTES

SHARE YOUR ACCOMPLISHMENT!

Meets Specifications

Hi:

Since you already got a very detailed code review, which touched many points I usually provide, I tried not to repeat and focused on other relevant points.

Congratulations, your project meets specification! Keep the good work going! :-)

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**

Awesome job! You can further enhance the game by adding features such as lives, a scoreboard, different types of enemy, levels, music et. al.

Object-Oriented Code

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

Excellent work! Remember to give a try to inheritance.

Documentation

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

Great job with your README. In case you need inspiration in future projects, you can find suggestions in the following posts:

http://stackoverflow.com/questions/2304863/how-to-write-a-good-readme
https://robots.thoughtbot.com/how-to-write-a-great-readme
http://www.wikihow.com/Write-a-Read-Me
http://docs.writethedocs.org/writing/beginners-guide-to-docs/
https://guides.github.com/features/mastering-markdown/

You can also have a look to our Writing a READMEs course.

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

Remember that there are always three places where you should write concise comments:

1. *Header comment*
   You should write at least what the code should do. You can also write your name, the date and why you wrote the code.
2. *Function Header*
   This comment should provide information about the purpose of the function. You should include at least the required parameters (if any), the transformations, and the expected output.
3. *Inline (above line) comment*
   You should write this type of comment in any part of your code you feel that no everyone will get what you are trying to achieve with certain the function or partial code.

In term of comments style, you can use JSDoc, YUIDoc or Docco.

Finally, if you may require some inspiration or advice about what or how to write better comments, you can check this blog post.

References
http://www.cs.utah.edu/~germain/PPS/Topics/commenting.html
http://stackoverflow.com/questions/6815903/what-is-the-correct-way-of-code-comments-in-javascript
https://www.thinkful.com/learn/javascript-best-practices-1/#Comment-as-Much-as-Needed-but-Not-More

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

DOWNLOAD PROJECT

5 CODE REVIEW COMMENTS ›

RETURN TO PATH

Rate this review

Student FAQ