Project Overview

You will be provided visual assets and a game loop engine; using these tools you must add a number of entities to the game including the player characters and enemies to recreate the classic arcade game Frogger.

Why this Project?

Games have a lot of objects and those object do a lot of different things; but sometimes those object do some very similar things as well. This creates a great opportunity to practice object-oriented programming, an important programming paradigm that influences your application architecture as well as provides performance optimizations.

What will I Learn?

You will learn JavaScript's object oriented programming features to write eloquently designed classes capable of creating countless instances of similarly functioning objects. You will discover a variety of ways inheritance and delegation can be used to create well architected and performant applications.

How does this Help my Career?

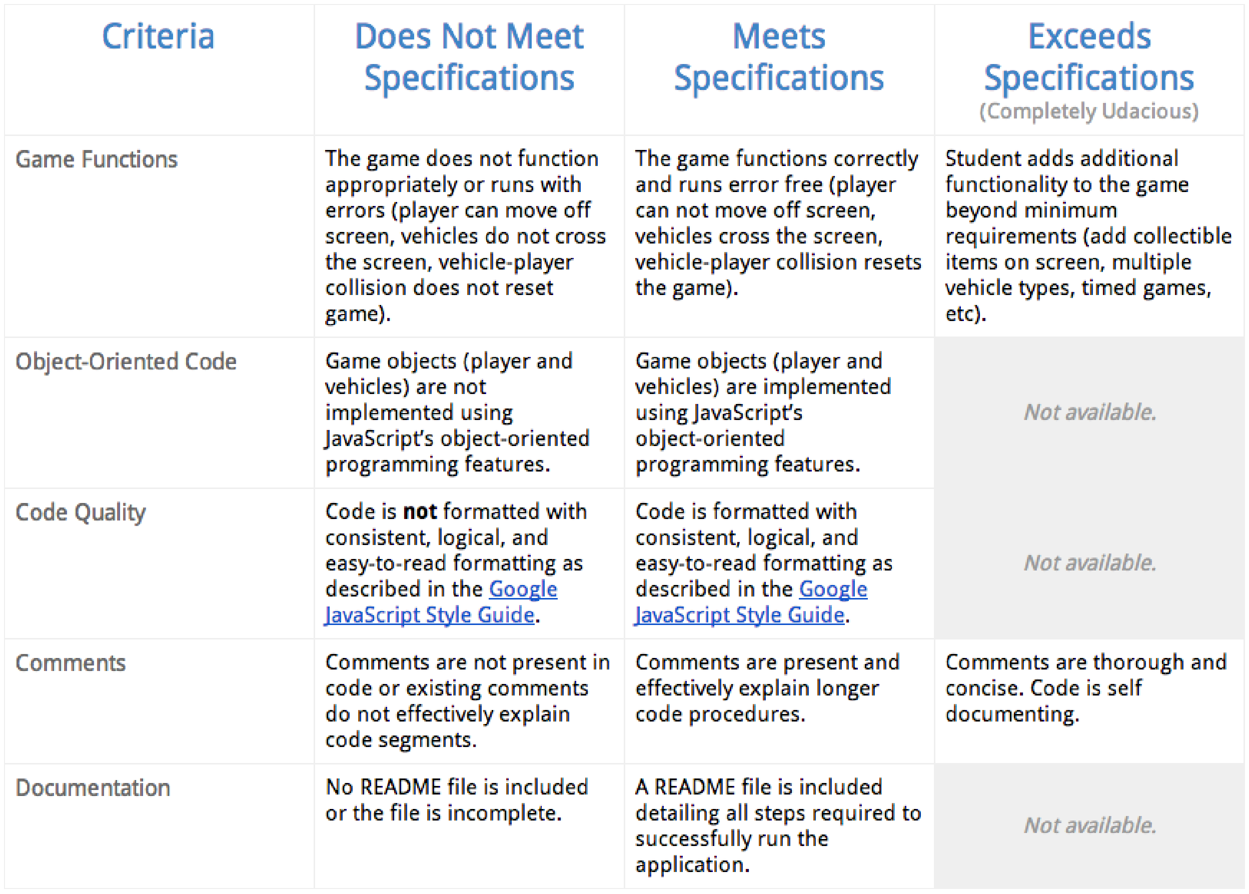
- JavaScript enables the development of complex applications on the web.
- JavaScript runs on normal web browsers, which makes it one of the most accessible and flexible programming languages.
- Complex applications that must be "broken down" into simpler entities that manage their own properties and functionality

How do I Complete this Project?

1. If you need a refresher on Object Oriented JavaScript, review our course; if you'd like a more detailed explanation as to how the game engine works, see our HTML5 Canvas course.
2. Download the art assets and provided game engine.
3. Review the video of the completed game and take note of the game's rules.
4. Review the code and comments provided in app.js
5. Identify the various classes you will need to write.
6. Identify and code the properties each class must have to accomplish its tasks.
7. Write the functions that provide functionality to each of your class instances.

Evaluation

Your project will be evaluated by a Udacity reviewer according to the rubric below. Be sure to review it thoroughly before you submit. All criteria must "meet expectations" in order to pass.



Criteria	Does Not Meet Specifications	Meets Specifications	Exceeds Specifications (Completely Udacious)
Game Functions	The game does not function appropriately or runs with errors (player can move off screen, vehicles do not cross the screen, vehicle-player collision does not reset game).	The game functions correctly and runs error free (player can not move off screen, vehicles cross the screen, vehicle-player collision resets the game).	Student adds additional functionality to the game beyond minimum requirements (add collectible items on screen, multiple vehicle types, timed games, etc).
Object-Oriented Code	Game objects (player and vehicles) are not implemented using JavaScript's object-oriented programming features.	Game objects (player and vehicles) are implemented using JavaScript's object-oriented programming features.	*Not available.*
Code Quality	Code is **not** formatted with consistent, logical, and easy-to-read formatting as described in the Google JavaScript Style Guide.	Code is formatted with consistent, logical, and easy-to-read formatting as described in the Google JavaScript Style Guide.	*Not available.*
Comments	Comments are not present in code or existing comments do not effectively explain code segments.	Comments are present and effectively explain longer code procedures.	Comments are thorough and concise. Code is self documenting.
Documentation	No README file is included or the file is incomplete.	A README file is included detailing all steps required to successfully run the application.	*Not available.*

Submission 1 -
https://github.com/bahalps/frontend-nanodegree-arcade-game

Submission 2 -
https://github.com/heyunen/frontend-nanodegree-arcade-game