Steven Hart
README - Planet Jumper Game Demo
Lumosity - Mobile Game Engineer Take Home

SOLUTION:

To solve this challenge using HTML5 canvas with JavaScript, I used a standard 'update/draw' sequence iteration. The script continually updates each game object and redraws them on the canvas.

I built this solution according to a small class system, including PLANET, PLAYER, and ITEMS - which can be assigned as other objects and doodads within the game. Level layout description is ALL localized at the top of the code and clear enough to be easily understood, configured, and extended by someone without programming expertise. Almost every other value in the game is configurable and commented in the same area. I'm confident that this solution accomplishes all of the functions outlined in the design document as well as some extra features I included to display its extendability.

EXTRA FEATURES:

- This solution is coded to be easily configured and extended. Instead of hardcoding the level layout, the **levelList** feature allows multiple levels to be designed and included in the same playthrough.
- I altered the planet art asset for functional reasons - it is now clear which planets are rotating even if they have no Items on the outside (which was difficult to discern with a flat green planet.

POSSIBLE NEXT STEPS:

- **More specific collision detection** - The current collision detection does not account for non-circular objects, but is appropriate to display the functions of the game in this stage.
- **Polished/detailed art assets** - The final version of the game could benefit from more detailed art assets. The provided assets are completely capable of showing the functionality of the game, so I focussed on the mechanics over changing the art.

EXTRA EXTRA IDEAS:

I had the following ideas to extend the functionality of the game, but did not include them because I thought they would disrupt the original gameplay described the design document. I have identified how they could be easily introduced in the code:

- Scale gravity based on the current planet size - easily done with small alterations to the planet object class.
- Black holes which pull the player mid-jump if he gets too close - can be extended from the planet object class
- Asteroids player must avoid in space as well as spikes you avoid on planets, this would allow the designer to discourage jumping to a planet even if its close - can be extended from Items object class