# Car Bowling

Game Design Document

## Goal

Create a game using a combination of WebGL, Cannon.js, three.js and Babylon.js

## Overview

Create a basic Bowling game but instead of throwing a bowling ball down the lane, you will be driving a car to knock over the pins

## Game Description:

Picture all water with an island (lane) in the middle with a ramp at one end with a little gap of water in between this ramp and another island which holds 10 pins. The plan is to make it 10 frames like a real bowling game with the same scoring system. The vehicle starts at one end of the lane and tries to hit the ramp at the perfect speed and angle to knock off all the pins from the island.

## Game Assets:

**Objects:** The game will consist of a few objects, a vehicle, pins, the lane with a ramp at the end which is connected to the island for which the pins sit on

**Environment:** the water for the surrounding world and a cubed sky.

**Audio**: Add some sounds for the car (gas and brake maybe), add some collision sounds for car and pins, add some sounds to car while in the air?

**UV maps:**  I can UV some images to the sky cube and some to the lane. The vehicle and pins can use basic material coloring

**Lighting:** Since there isn’t going to be much walls if any, a simple sun should be enough

## UI: I would enjoy keeping the camera at a set height above the car and not allow it to move lower than this point but maybe swing it left and right. The camera would then follow the car in the x and y direction.

## Game Object Mechanics

**Vehicle Mechanics:** There will be a hovering vehicle so I do not have to worry about rotating the tires correctly. The vehicle will have basic driving ability such as accelerating and braking. The car will use have a mesh box within in body to use for collision with pins.

**Pin Mechanics:** Pins will be set up in the correct 1-2-3-4 formation on the lone island. They will be influenced by the vehicle collision. The center of mass should be in the lower half of the pin and the weights will have to be tuned for gameplay.

**Lone Island Mechanics:** I will have some sort of plane that cuts through the red lines on the pins to detect which pins are not standing up after the collision with the vehicle. This will help set up keep track of current frame pin position and final score.

## States and Transitions

Launching the game goes immediately into gameplay state. An instruction screen may be added if time allows, but initial instructions will be in a simple text file. The state of the game is either in driving/bowling mode or in counting resetting pins, or game finished. It starts off each driving/bowling mode with all 10 pins set up for the first part of the frame and how many ever standing will be left for the second part of that same frame. After the player launches off the ramp a timer starts (maybe 5 secs) to allow for collision with the pins and pin on pin collision. After this timer ends, we count how many fell, and leave the others up for the second part of the frame. If the player knocks all the pins over in the first half of the frame it’s a strike and goes to the next frame. If the player knocks all the remaining pins in the second half of the frame it’s a spare and the player continues to the next frames. Strikes account for 10 pts + the next two knockdowns of pins. Spares count for 10 pts + the next knockdown of pins. Total of 10 frames. After finishing the 10 frames score is printed and you can play again?

## Milestone plan

**Basic models and scenery** – A ocean with an imported obj from blender consisting of lane a ramp that launches to a lone island of pins. The vehicle starts at the far side of the lane.

**Car dynamics** – add basic driving mechanics such as acceleration, braking and collision with pins

**Game management** – Figure out how to implement the 10 frames of the standard bowling game

## Tuning

If time allows, add things to lane such as objects, slows, boosts, etc. Maybe add a leaderboard.