**Practical Exercise 3 - RL with Function Approximation**

Ido Glanz – 302568936 Matan Weksler – 302955372

**2 – Preliminary: The Mountain Car Problem and feature engineering**

1. **The systems state and action space and reward function as we are understood them are as follows:**
   1. **State space -** a tuple of [position, velocity], indicating both the position of the cart (in the range of -1.2 to 0.6) and the velocity (ranging from -0.07 to 0.07). position and velocity are both continuous in the above range. The goal state for example is [position = 0.5, any action]
   2. **Action space -** a discrete integer in the range [0, 1, 2] indicating: [0=left, 1=neutral, 2=right]. The actions are then added to the current speed (subtracting the acceleration/deceleration caused by gravity).
   3. **Reward** - a function of state (in practice only dependent on position) equaling 1 if in terminal state (position >= 0.5) and 0 otherwise.
2. **Value Function Approximation -** The advantages in encoding the state vector as an RBF (as we understood it), is that we are allowing for more complex and diverse function approximation not necessarily linearly dependent on the state vector. Specifically, RBFs map the (standardized) deviation of the state (or absolute distance) from centers we initialize and in this manner, we can generate many RBFs with different “centers” (or kernels) whereas each could later be weighted as to its “importance” to the value of the state. In control problems such as ours, these types of functions could in a way describe our distance or error to a specific goal (in either of the state variables) and hence prove to be useful.
3. **Data Standardization -** generally, standardization allows us to “treat all variables equally” as well as un-bias them hence they are centered around 0. This is useful when using RBFs as their kernels could be