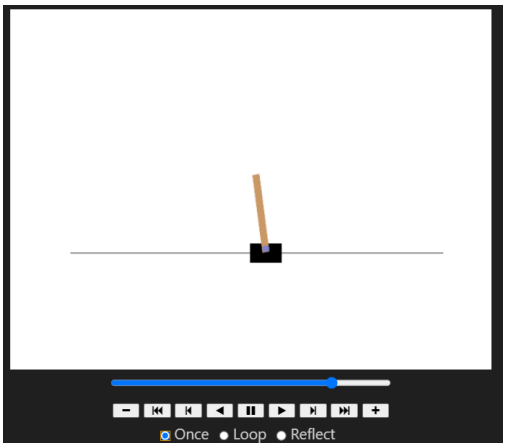
Task 2.3a)

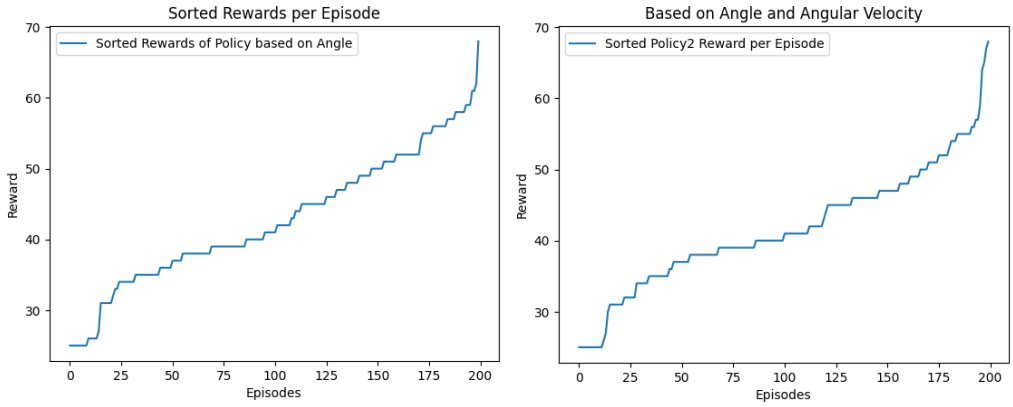
Die Zustände sind beschrieben durch die 4 Variablen: Position, Geschwindigkeit, Winkel des Poles und Geschwindigkeit des Poles.
Es gibt 2 mögliche Aktionen in jedem Zustand, die den Fahrtrichtungen links und rechts entsprechen.
Nach der Wahl einer Aktion wird die Umgebung entsprechend geupdatet und der neue Zustand (beschrieben durch die 4 Variablen) berechnet.
Für jeden Frame, in dem ein gewisser Winkel nicht überschritten wird, erhält der Algorithmus einen Reward von 1. Sollte das Balancieren fehlschlagen gibt es keinen Reward und wenn 200 Schritte erreicht sind endet das Experiment.

Der Screenshot zeigt einen Ausschnitt der Animation (im Notebook enthalten):



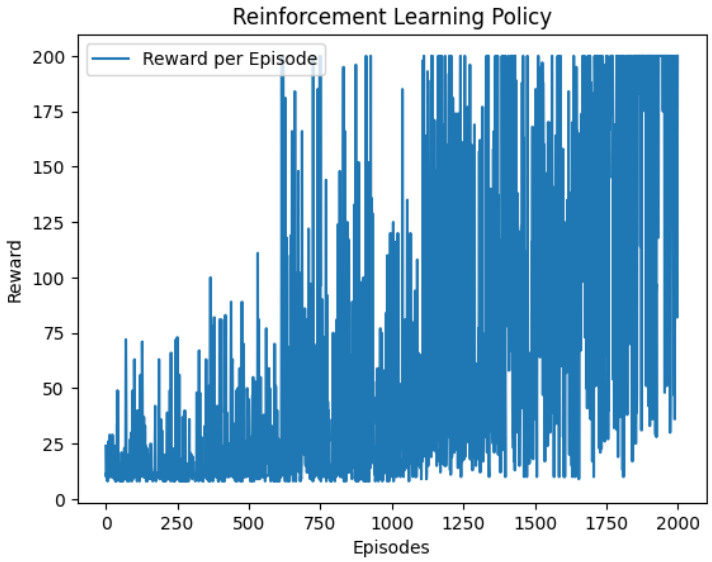
Task 2.3b)

Einfachste Strategie basiert nur auf dem Winkel, erreicht jedoch niemals einen Reward über 70. Für den zweiten Graph ist die Geschwindigkeit der Winkeländerung berücksichtigt, was jedoch zu keiner Verbesserung führt. Entsprechend kann die Aufgabe von 200 Time Steps nicht mit einer leichten Strategie basierend auf den beiden Variablen gelöst werden und es müsste mehr Information der Umgebung betrachtet werden.



Task 2.3c)

Für das Diskretisieren ist eine Definition über bins nötig, damit es nur eine endliche Anzahl an möglichen Zuständen gibt. Je nach Zustand lässt sich das Modell trainieren und man erhält folgendes Ergebnis mit dem e-Greedy Algorithmus:



Hier ist klar zu sehen, dass der Algorithmus lernt, wie er die Stange zu balancieren hat. Je länger er trainiert, desto öfter erreicht er den maximalen Reward von 200. Durch die Diskretisierung werden Informationen verloren, da nurnoch die bins betrachetet werden. Es verschlechtert die Performance aber beschleunigt die Berechnung. Entsprechend, kann es sein, dass das Modell das Problem nicht verlässlich lösen kann, wie es hier trotz 2000 Trainingsepochen der Fall ist. Dennoch ist das Modell deutlich besser als die zuvorbetracheteten intuitiven Strategien.