Geliederung

1. Einleitung
2. Physik und Bewegung
3. Net und Sycronisierung
4. Graphik und Modell
5. UI und Interaktion
6. Einleitung

Das Spiel “Drift” ist ein Autorennspiel für mehrere Spieler in eine LAN. Der Spiel ist mit Unity-Engine programmiert. Die Modelle wie die Häuse und Autos sind aus Unity-Assetsstore.

1. Physik

Die Rechnungen der Physik des Autos sind mit Vectoren. Ein Auto hat immer paar physikalische Attribute:

Masse (float mass)

Bewegungsrichtung und Geschwindigkeit (Vector3 velocity)

Position (Vector3 position)

Forwarts Richtung (Vector3 forward)

Aufwarts Richtung (Vector3 upward)

Ein Auto hat auch paar Attribute für die Bewegung (Ein AnimationCurve ist ein kurve mit Graphik bearbeiten kann, damit kann man ein Wert von dem kurve auslesen):

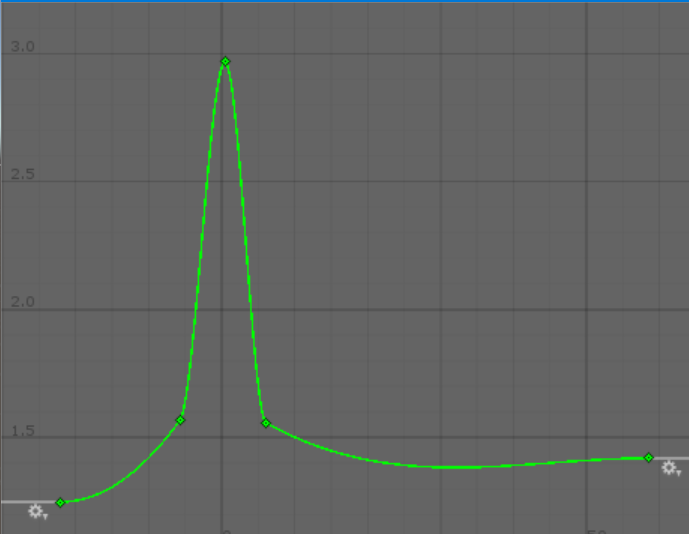
Beschleunigung/rückwarts Beschleunigung (float acceleration/backAcceleration)

Trägheit der Beschleunigung (AnimationCurve accelerationDrag)

Beschleunigung der Drehung (AnimationCurve torque)

Trägheit der Rotation(float rotateDrage)

Die accelerationDrag und torque siehen so aus:

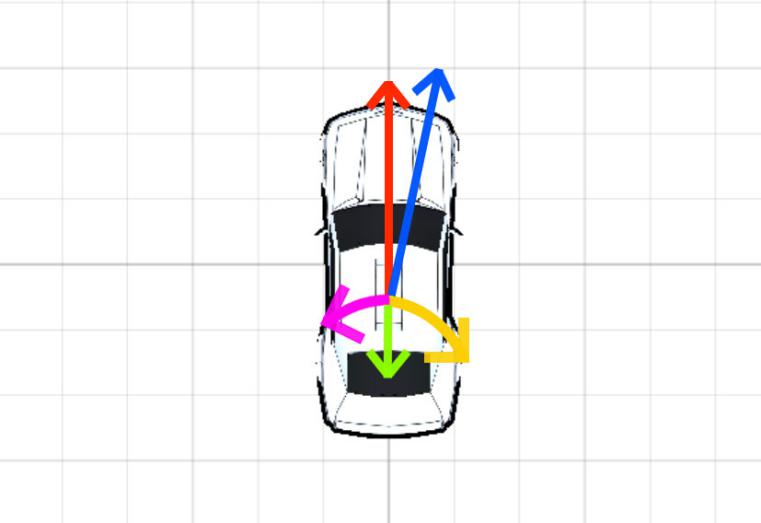
 

Die Rechnung der Bewegung ist:

velocity += InputY() \* forward \* acceleration \* deltaTime (deltaTime ist die Zeit jeder Frame, InputY/InputY ist die horizontale/vertikale Input des Spieler zwischen -1 und 1)

AddTorque (InputX() \* upward \* torque.Evaluate(velocity.magnitude) \* deltaTime)

(AddTorque() ist ein Funktion, die ein Geschwindigkeit zu dem Rotation des Autos machen. Evaluate() ist nimmt den Wert der X-Achse und gibt den Wert der Y-Achse der Kurve zurück. magnitude ist die Länge einer Vector)



Rot: anfangs velocity, Grün: accelerationDrag, Gelb: torque, Lila: torqueDrag, Blau: neue velocity