

Protokoll 20.11.2018

Zusammenfassung der letzten Sitzung:

- 2D-Editor-Struktur

- eine Timeclass -> im Hinblick auf Physik/Animation

 - Timeslices

 - Framerate belassen (bzw von System bestimmen lassen)

Aktuell:

Präsentation Thomas:

Frage: Linkshändiges/Rechtshändiges System?

Schwerpunkt auf 3D Physik -> Anlehnung an „downscaling“ in Unity 3D/2D

Vereinfachung von Meshobjekten zur Vereinfachung der Berechnung

Auftrag: Wörterbuch anlegen (für das fudge Dev-Team)

=> Collider oder Proxy? => Collider

Implizite/Explizite Collider

Darstellung <--> Physik

Timemanagement

- Gametime skalierbar

- >Simulationtime abhängig von Gametime

- >Frametime unabhängig von Gametime

Hinweise zu Form der Dokumentation

Frage: Umsetzung von Physik im Browser (/in javascript)?

Arten der Krafteinwirkung:

- Force

- Impuls

- Geschwindigkeit

- Beschleunigung

⇒ Verbindung (Addition) von Krafteinwirkung (deLambert Algorithmus)

Klärung des Ablaufs physikalischer Kollision

->Colliderwahl an Situation anpassen (vom Gamedeveloper wählbar machen)

Frage: Wie sieht eine geeignete Schnittstelle für die Übergabe der Informationen aus der Physikklasse aus?

Auftrag: Untersuchung der Physik in ThreeJS (implementierte Physikengine)

Auftrag: Physik mit Webassembly?

Präsentation Kathrin:

WebXR

->Präsentation AR-Applet aktueller Stand

Darstellung des Problems benötigter ChromeFlags

-> Deviceerkennung

Abfrage der Device Pose (Position und Orientation im Verhältnis zum Ankerpunkt eines zu rendernden Meshes)

-> Übergabe der Daten an den Renderer

Recherche: Flaghacks für Chrome (Chromium/Elektron)

Auftrag: Typescript.d.ts erstellen

ARCore Dokumentation recherchieren