Protokoll 01/11/2017 Anwesend: Moritz, Selim

Option 1: Actor Portierung, Diagramm umstellen

* Pro: weniger Aufwand, Implementation von ARToolKit effizient
* Contra: Abweichung vom alten Diagramm

Option 2: Tracking selber implementieren, Diagramm bleibt gleich

* Pro: -
* Contra: Deutlich mehr Aufwand, weniger Effizient, doppelte Buchhaltung der Objekte

**Option 1 wurde gewählt**.

Grundkonzept Initialisierung:

-> startet

Startskript (stirbt nach dem Start) -> Controller -> ARToolKit Controller -> (Codes)

->Gameobjects

(leere MonoBehaviorskripts ohne Start/Update)

Welche Änderungen müssen gemacht werden?

* Markerlogik
  + Updatefunktion muss in den Controller importiert werden
  + SignaltrackingUpdate()
* Locks müssen als Actors implementiert werden
* DLL als Actor?
* Portierung von Controller(evtl. vom Objekt lösen), Video(bleibt Objekt gebunden), Marker (bleibt Objekt gebunden)
* Start und Update Message muss für jeden Actor verfügbar sein

Plan für die nächste Zeit:

1. Dieses Protokoll /Pseudotemplate für Tests
2. Tests schreiben (TDD) auch für Konstruktoren
3. Actor Grundgerüst aufbauen / Konstruktoren
4. ARToolKit für Actor model bereit machen

Sonstiges besprochenes:

* *Die Verwendungsart der Dictionary in der Markerklasse führen zu einer Unity spezifische Verbindung. Soll durch eine Message ersetzt werden -> neue Data*
* UpdateTrackingInfo() weist ein Actorverhalten auf. Ist im Markerskript. Sendet ein leeres Message Objekt
* bool signalTrackingUpdate, scheint in mehreren Skripts benutzt zu werden und hat eine globale Funktion. Möglichkeit die Variable zu entfernen indem man während des Updates gleich beides aus dem Controller wie auch dem Markerskript ausführt.
* Interlocked = Funktionalität die ein Lock simuliert
* Ref latestTrackingInfo wird für eine Compare & switch benutzt -> Frage: Wie bekommen die einzelnen Actors ihre Informationen
* Geteilter Zugriff auf die DLL?
  + Maximal 3 Marker im Blickfeld
  + Beschränkung das nur ein Prozess gleichzeitig auf DLL zugreifen darf
    - Test: Setze bool auf true und falls das System dann nicht mehr funktioniert, ist gezeigt dass es eine notwendige Beschränkung ist.
* InitAsync() startet neuen Thread auf welchem dann die Updates gemacht werden
* OnFrameReceived() soll eine Message zum Controller in Form einer Bitmap
  + Soll auch TrackingUpdate aktivieren
* Ideen zu den Test die aufgekommen sind
  + Erstellen einer Generic Class z.B. class easyTestClass<T1,T2> -> testen ob dieser korrekt erstellt wurde.
  + Zuerst Objekte zum Senden erstellen, dann testen. Konstruktoren werden nicht der problematische Teil sein.
  + **Tests für Konstruktoren schreiben und dann diese programmieren, ohne zu diskutieren und unnötig Zeit zu verlieren.**