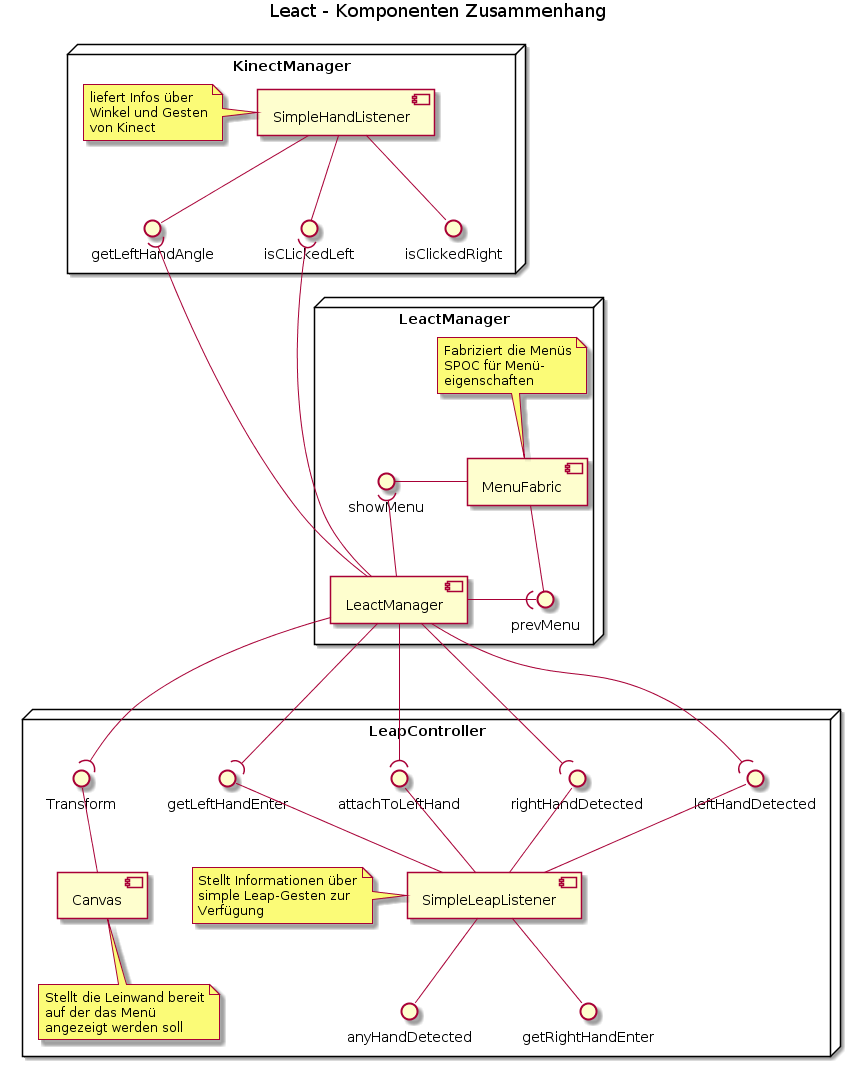
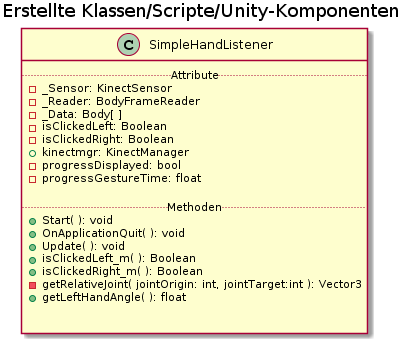
Die Idee der Leact-Erweiterung für Unity ist es, Daten von Kinect und Leap Motion zu vereinen um möglichst wenig Interaktion mit echten Kontrollern zu haben. Es geht darum allein mit dem Körper in VR mit der Umgebung zu interagieren. Der Schwerpunkt lag hier auf Menüführung mit Kinect-Gesten und Interaktion mit Objekten durch Leap Motion. Dazu mussten Die Kontroller für Kinect und Leap Motion erweitert werden und ein komplett neuer Kontroller – der beide Daten fusioniert – erstellt werden.

Die Leact-Erweiterung ist eine Erweiterung für Unity. Um die Erweiterung einzubinden, muss das Script SimpleHandListener.cs dem GameObject Kinectmanager hinzugefügt werden. Das Script SimpleLeapListener.cs muss dem LeapController hinzugefügt werden. Der Kinect Manager sorgt für Schnittstellen mit der Kinect und sorgt dafür das Körper getrackt wird und in VR verfügbar ist. Der LeapController ist der Controller der Leap Motion. Dem LeapController muss zudem noch ein Canvas (Leinwand) hinzugefügt werden. Diese Sorgt dafür das das Manü immer direkt im Zentrum des Sichtfeldes des Spielers ist.



~~Die Scripte sind im Unity-Kontext als Komponenten zu verstehen~~. Der SimpleHandListener im Kinect Manager erweitert die von Kinect bereit gestellten Standardgesten. Der SimpleLeapListener erweitert das Standard Gestenset von Leap Motion. Der LeactManager ist der neue Kontroller der die neuen Gesten definiert und eine Factory für Menüs stellt. In der Menüfabrik wird die Menüstruktur definiert und das Layout sowie Funktionalität festgelegt

Der SimpleHandListner stellt einfache Gesten zur Verfügung. Er stellt Informationen wie ein Click mit der Hand bereit. Er stellt auch den Winkel bereit, den die linke Hand zur linken Schulter hat. Die Linke Hand sowie die Linke Schulter sind Gelenke die die Kinect erkennen kann. Es hat sich herausgestellt das es die Bedienung deutlich angenehmer macht wenn der Winkel zwischen Schulter und Hand bestimmt wird. Der bestimmte Winkel zwischen Schultermitte und Hand sorgte für Überstreckung des Körpers, was unangenehm war und zu Fehlern in der navigation durch Menüs führte.

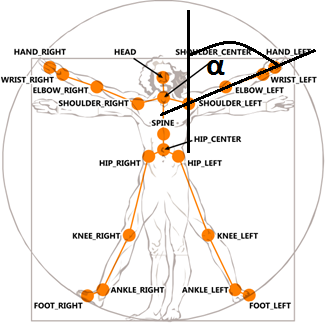
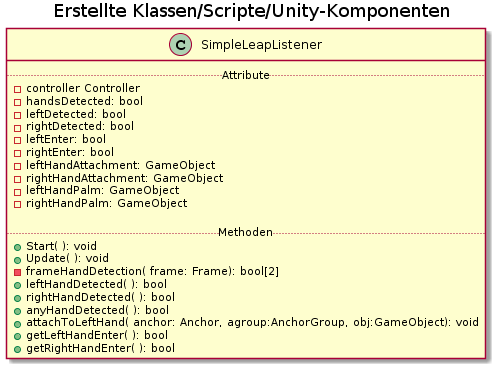
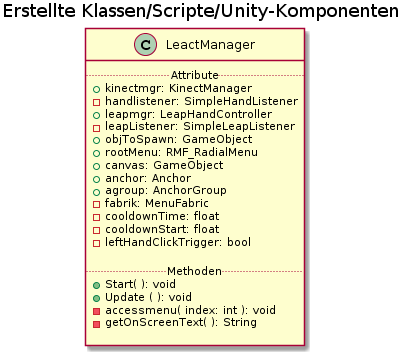


Abbildung 3 https://possiblywrong.wordpress.com/2012/11/04/kinect-skeleton-tracking-with-visual-python/



Der SimpleLeapListener stellt Gesten für Leap Motion bereit. Er stellt fest ob Hände zeitlich betrachtet vom letzten Frame bis zu diesem den Sichtbereich (FOV, field of view) betreten. Dadurch wurde eine Selektions-Geste realisiert, mit der es möglich war eine Selektion aus dem Menü ins Bild zu ziehen. Per Menü wird z.b. gewählt was für ein Objekt erstellt werden soll und per „reinziehen“-Geste ist das Objekt an der Hand verankert. Der Anker der verwendet wurde ist ein Anker aus einem Leap Motion beispiel. Objekte die durch die Leap Motion Hände manipuliert werden können sollen, sowie im Anker verankert werden sollen, müssen bestimmte Komponenten besitzen. Bewährt hat sich ein bestehendes Prefab zu kopieren und an die eigenen Bedürfnisse anzupassen.



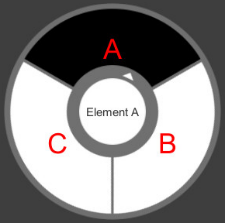
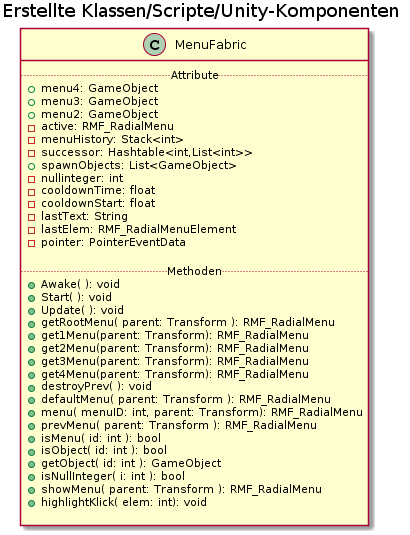
Der LeactManager bezieht die vom SimpleHandListener und SimpleLeapListener bereitgestellten Gesten und Daten und kombiniert sie zu neuen Gesten. Hier ist definiert wie das Menü aufgerufen wird, wie im menü navigiert wird und wie man Objekte in der Welt erstellt. Mit einem einfachen Klick am linken Bildschirmrand wird das Menü aufgerufen. Mit Rotationsbewegungen des Linken Arms können im Radialmenü (Abb. 6) Elemente Gewählt werden. Die Hände dürfen nicht in den Sichtbereich der Leap Motion erscheinen denn dann wird das Menü ausgeblendet um Interaktionen mit der Umgebung nicht zu stören. Mit erneutem Klick auf ein Menüelement wird das dazugehörige Untermenü aufgerufen. Es wird ein neues Menü mit entsprechenden Elementen generiert und an Stelle des alten Menüs angezeigt. Um ein Objekt aus dem Menü in der Welt zu erzeugen wird das Menüelement in den Sichtbereich der LeapMotion gezogen. Das Objekt ist dann an der linken Hand verankert und kann dort mit der Rechten Hand entnommen werden. Um in Menüs zurück zu vorherigen menüs zu navigieren, ist das oberste Element standadmäßig mit der Zurück-Funktion belegt.

Abbildung 6

In der MenüFactory ist die Komplette Menüstrucktur und Verhalten definiert.



Es ist festgelegt wie viele Elemente jedes Menü hat und ob sie auf ein weiteres Menü oder auf ein Element zeigen. Änderungen oder Erweiterungen des Menüs müssen nur hier vorgenommen werden. Die Factory merkt sich den bisherigen Menüpfad, sodass zurückgehen bis zum ersten Aufruf des Menüs möglich ist.

Das Menü wird in der Factory generiert, was hohe Austauschbarkeit der Menüs gewährleistet. Blah blah blah ich hab kein bock mehr

Das Verwendete Menü ist aus dem Asset Store von Unity Bezogen. Das Menüscript wurde geringfügig erweitert um es mit der MenuFactory benutzbar zu machen.

