

HOCHSCHULE FÜR TECHNIK UND
WIRTSCHAFT

ENTWICKLUNG VON MULTIMEDIASYSTEMEN

Krypton

Youssef Nassar Matrikelnummer: 566297

Aaron Bach Matrikelnummer: 563346

David Karaguiozian Matrikelnummer: 567196

Korbinian Hurt Matrikelnummer: 576086

Dozent:
Sebastian Keppler

11. Februar 2022

Inhaltsverzeichnis

1 Funktionale Anforderungen	3
1.1 UI für alle Komponenten (Hauptmenü/Atom/Molekül/Task)	3
1.2 Visualisierung von Atomen mit Bohr Modell	3
1.3 Atome spawnen auf Image Targets	3
1.4 Molekülerstellung bei richtigen Image Targets innerhalb einer Reichtweite	3
1.5 Moleküldespawn	3
1.6 Aufgabentypen	4
1.7 Aufgabenerstellung	4
1.8 Aufgabenliste speichern	4
1.9 Laden gespeicherter Aufgabenlisten	4
1.10 Tipps für das Lösen der Aufgaben	4
2 Anwendung	5
2.1 Vuforia Image Targets	5
2.2 Aufgaben-Modus	5
2.3 Atom-Modus	6
2.4 Molekül-Modus	7
3 Verwendete Technologien	9
3.1 Vuforia	9
3.2 Unity	9
3.3 Git	9
4 Systemarchitektur	10
5 Einschätzung	11
5.1 Aaron	11
5.2 Youssef	11
5.3 David	11
5.4 Korbinian	12
5.5 Gesamt	12

Abbildungsverzeichnis

1	Aufgaben-Modus	5
2	Atom-Modus	6
3	Molekül-Modus	7
4	Systemarchitektur	10

1 Funktionale Anforderungen

1.1 UI für alle Komponenten (Hauptmenü/Atom/Molekül/Task)

Erfüllungsgrad: 10/10

Erfolgreich für alle Menüs einheitlich erstellt.

1.2 Visualisierung von Atomen mit Bohr Modell

Erfüllungsgrad: 10/10

Erfolgreich erfüllt, vollständig dynamisch geschrieben, zudem wird der Nukleus physikalisch simuliert. Das Modell hat einen Atomkern sowie Schalen mit Elektronen die sich bewegen, alles in 3D umgesetzt. Orbit/Schalen der Elektronen um den Kern sind 2D.

1.3 Atome spawnen auf Image Targets

Erfüllungsgrad: 10/10

Spawnen erfolgreich auf den Image Targets wie gewollt.

1.4 Molekülerstellung bei richtigen Image Targets innerhalb einer Reichweite

Erfüllungsgrad: 10/10

Wenn die richtige Kombination von Atomen innerhalb der Kamera (+ eingestellter Reichweite) sind, wird ein Molekül erstellt. Es gibt Probleme mit Vuforia die Distanz abzuschätzen weswegen die Ermittlung der Distanz teilweise fehlerhaft ist.

1.5 Moleküldespawn

Erfüllungsgrad: 10/10

Molekül wird despawned sobald ein Teil der Kombination nicht mehr vorhanden ist, bzw. die Voraussetzung nicht mehr erfüllt ist. Auch kann man das Molekül despawnen mit einem Button der eingefügt wurde. Die dazugehörigen Atome spawnen wieder, sobald das Molekül verschwindet/despawned.

1.6 Aufgabentypen

Erfüllungsgrad: 10/10

Man kann die Aufgaben:

- Atom erstellen in der Atom-Szene
- Atombeschreibung anschauen
- Molekül erstellen
- Molekülbeschreibung anschauen

Erstellen so wie erfüllen sobald die Anforderungen erreicht werden.

1.7 Aufgabenerstellung

Erfüllungsgrad: 10/10

Vollständig funktionsfähig. Man kann eine Liste mit Aufgaben erstellen und abarbeiten.

1.8 Aufgabenliste speichern

Erfüllungsgrad: 10/10

Listen mit deren Aufgaben können gespeichert auf der Festplatte gespeichert werden.

1.9 Laden gespeicherter Aufgabenlisten

Erfüllungsgrad 10/10

Erfolgreiches Laden der jeweils gespeicherten Aufgabenlisten ist möglich. Es ist nur 1 aktive Liste gleichzeitig möglich.

1.10 Tipps für das Lösen der Aufgaben

Erfüllungsgrad: 10/10

Der Benutzer kann sich jederzeit Tipps anzeigen lassen, die Ihm beim Bearbeiten als Hilfestellung dient.

2 Anwendung

Krypton ist eine AR-Anwendung, die dem Benutzer hilft atomare und molekulare Strukturen zu verstehen und zu visualisieren. Die Anwendung wurde entwickelt, um das Erlernen der besagten Strukturen zu fördern und zu erleichtern, und bietet 3 verschiedene Modi, die der User nutzen kann.

2.1 Vuforia Image Targets

In der gesamten App verwenden Benutzer Vuforia Image Targets als Mittel zur Kommunikation mit der Anwendung. Jedes Atom hat eigene Image Targets, dass als Repräsentation eines Atoms oder einer Molekülstruktur für die Anwendung dient. Diese Image Targets müssen in die Sicht der Kamera gebracht werden, um die Erstellung eines relativen Atom-/Molekülmodells auszulösen. Gegenwärtig kann jedes Atom Image Target anhand seiner äquivalenten Abkürzung im Ordner “Imagetargets” im Projekt-Verzeichnis gefunden werden.

2.2 Aufgaben-Modus



Abbildung 1: Aufgaben-Modus

Der Aufgabenmodus ermöglicht die Erstellung verschiedener Benutzeraufgaben, die in den anderen 2 Visualisierungsmodi der Anwendung abgeschlossen werden können.

In diesem Modus können Benutzer sowohl Erstellungsaufgaben als auch Informationsaufgaben erstellen. Erstellungsaufgaben, die entweder für Atom-

oder Molekülstrukturen erstellt werden können, können durch klicken auf die Buttons "Create Atom" oder "Create Molecule" und anschließende Auswahl des entsprechenden Atom-/Molekülmodells aus der Liste unten erstellt werden. Ebenso können Informationsaufgaben, die ebenfalls für Atome und Molekülmodelle erstellt werden können, durch Klicken auf die Buttons "Info Atom" oder "Info Molecule" und Auswahl des entsprechenden Moleküls aus der Liste erstellt werden. Sobald alle gewünschten Aufgaben erstellt wurden, können die Benutzer mit der Erledigung der Aufgaben beginnen, indem sie auf "Save + Start" klicken, oder ihre Aufgabenliste einfach durch klicken auf "Save" speichern.

2.3 Atom-Modus

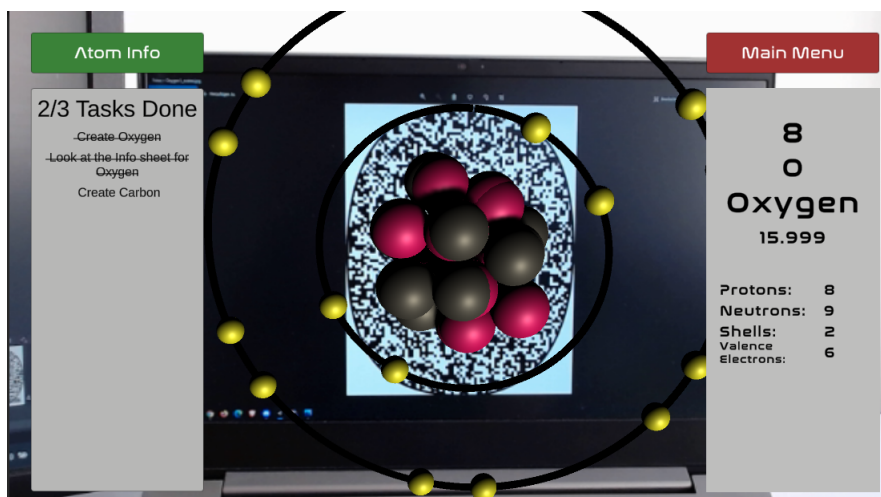


Abbildung 2: Atom-Modus

Der Atommodus ist eines von 2 Werkzeugen zur Visualisierung chemischer Strukturen. In diesem Modus ist die Kamera des Benutzers aktiviert, um die Verfolgung der Vuforia-Atom Image Targets zu ermöglichen. Auf diesem Bildschirm sehen die Benutzer nur eine Taste "Show Atom Info", die inaktiv ist, bis ein Atom erzeugt wird, und eine Taste "Main Menu", die den Benutzer zum Hauptmenü zurückbringt. Wenn der TaskMode aktiv, erscheint auf der linken Seite eine zusätzliche Aufgabenliste, in der die zu erledigenden Aufgaben dargestellt werden. Wenn der Benutzer nicht weiß, wie er die Aufgabe erledigen soll, kann er auf die einzelnen Aufgabenpunkte klicken und erhält einen Hinweis, wie die Aufgabe erledigt werden sollte. Um ein Atom-AR-Modell zu erstellen, muss der Benutzer eines der Vuforia

Atom Image Targets in das Blickfeld der Kamera bringen. Sobald es vom Vuforia SDK erkannt wird, erscheint ein Atommodell über dem entsprechenden Image Target. In dieser Prototyp-Version können Benutzer Stickstoff-, Sauerstoff-, Kohlenstoff- und Wasserstoff-Atommodelle erstellen.

Nach der Erstellung eines Atoms wird jede für dieses Atom offene Aufgabe "Create Atom" als abgeschlossen markiert.

Nach der Erstellung eines Atoms wird die Taste Show Atom-Info angezeigt, und durch Anklicken dieser Taste wird das Atom-Info-Canvas auf der rechten Seite aktiviert. Hier sehen die Benutzer ein Canvas mit relevanten Informationen über das Atom, wie z. B. die Ordnungszahl, die Anzahl der Protonen usw... Durch die Anzeige dieses Atom-Info-Canvas werden alle Aufgaben von Atom-Info angezeigt für das betreffende Atom abgeschlossen.

2.4 Molekül-Modus

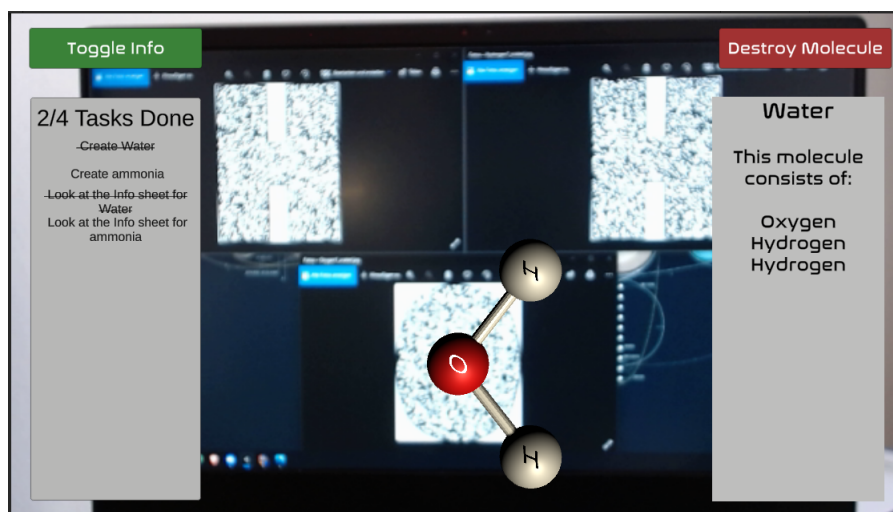


Abbildung 3: Molekül-Modus

Der Molekülmodus ist das zweite von 2 Tools zur Visualisierung chemischer Strukturen. In diesem Modus ist die Kamera des Benutzers aktiviert, um die Verfolgung der Vuforia Image Targets zu ermöglichen. Auf diesem Bildschirm sieht der Benutzer nur einen Button "Spawn", der inaktiv ist, bis ein Molekül erzeugt wird, und einen Button "Main Menu", der den Benutzer zum Hauptmenü zurückführt. Wenn gerade Aufgaben ausgeführt werden, erscheint auf der linken Seite eine zusätzliche Aufgabenliste, in der die Aufgaben dargestellt werden. Wenn der Benutzer nicht weiß, wie er die Aufgabe erledigen soll, kann er auf die einzelnen Aufgabenpunkte klicken und erhält

einen Hinweis, wie die Aufgabe erledigt werden sollte.

Um ein Molekül-AR-Modell zu erstellen, müssen die Benutzer die entsprechende Anzahl von Image Targets in das Blickfeld der Kamera bringen. Die Anzahl der erforderlichen Image Targets wird durch die Molekülstruktur des Moleküls bestimmt. Für ein CO₂-Molekül werden beispielsweise 1 Kohlenstoff- und 2 Sauerstoff- Image Targets benötigt. Jedes erkannte Target wird mit einer 3D-Kugel in der Mitte markiert, um dem Benutzer zu zeigen, dass es registriert wurde.

Sobald sich die erforderliche Anzahl von Image Targets im Blickfeld der Kamera befindet, wird die Schaltfläche Molekül erstellen aktiviert. Durch Anklicken dieser Schaltfläche wird eine 3D-Moleküldarstellung des Moleküls erstellt. Wenn für das Molekül Aufgaben zum Erstellen von Molekülen offen sind, werden diese als abgeschlossen markiert.

Sobald ein Molekül erstellt wurde, wird der Button "Toggle Info" aktiviert, und durch Anklicken dieser Schaltfläche wird die Molekül Informationsfläche auf der rechten Seite aktiviert. Hier sehen die Benutzer eine Canvas mit den relevanten Informationen für das Molekül. Durch das Betrachten der Molekül-Info-Canvas werden alle Aufgaben zum Anzeigen von Molekül-Informationen für das betreffende Atom abgeschlossen.

3 Verwendete Technologien

3.1 Vuforia

Vuforia ist ein Augmented-Reality-Softwareentwicklungskit (SDK), dass die Erstellung von Augmented-Reality-Anwendungen ermöglicht. Es verwendet Computer-Vision-Technologie, um planare Bilder und 3D-Objekte in Echtzeit zu erkennen und zu verfolgen. Diese Bildregistrierung Fähigkeit ermöglicht es Entwicklern, virtuelle Objekte wie 3D-Modelle und andere Medien in Bezug auf reale Objekte zu positionieren und auszurichten, wenn sie durch die Kamera eines mobilen Geräts betrachtet werden. Das virtuelle Objekt verfolgt dann die Position und Orientierung des Bildes in Echtzeit, so dass die Perspektive des Betrachters auf das Objekt mit der Perspektive auf das Ziel übereinstimmt. Es scheint somit, dass das virtuelle Objekt ein Teil der Szene der realen Welt ist.[1]

3.2 Unity

ist eine von Unity Technologies entwickelte plattformübergreifende Spiele-Engine, die erstmals im Juni 2005 auf der Worldwide Developers Conference von Apple Inc. als exklusive Mac OS X-Spiele-Engine angekündigt und veröffentlicht wurde. Die Engine wurde seitdem schrittweise erweitert, um eine Vielzahl von Desktop-, Mobil-, Konsolen- und Virtual-Reality-Plattformen zu unterstützen. Nennenswerte Spiele wie Pokémon Go, Beat Saber und Cyphead wurden mit Unity umgesetzt. Ebenfalls ist Unity beliebt für die Entwicklung von Indie-Spielen.[2]

3.3 Git

Ist eine freie Software zur verteilten Versionsverwaltung von Dateien.[3] Zu den Zielen gehören Geschwindigkeit, Datenintegrität und Unterstützung für verteilte, nichtlineare Arbeitsabläufe (Tausende von parallelen Zweigen, die auf verschiedenen Systemen ausgeführt werden).[4]

4 Systemarchitektur

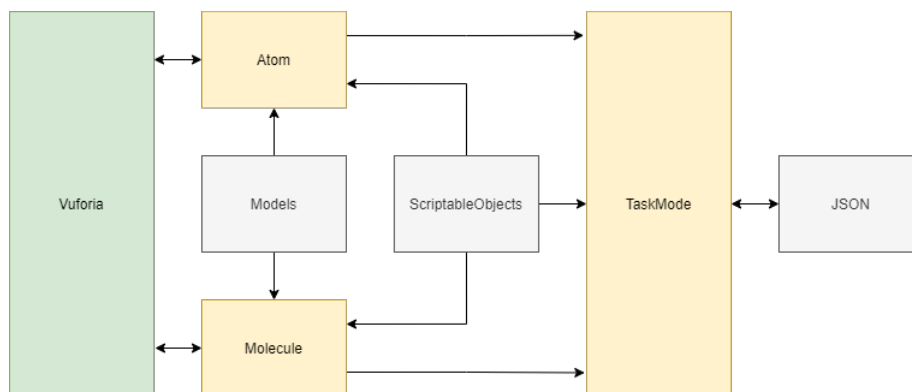


Abbildung 4: Systemarchitektur

Das System besteht aus 4 Hauptkomponenten: Vuforia, Atom, Molecule und Taskmode.

Vuforia(Erklärung siehe 3.1) wird zum Tracking der Atome/Moleküle genutzt, verwendet wird dazu die in Vuforia Integrierte Technologie ImageTarget - wir haben für jedes Atom mehrere Image Targets entwickelt. Atom und Molecule erstellen/zerstören dann die zugehörigen Atome über den Image Targets sobald diese für die Kamera sichtbar/nicht mehr sichtbar sind und die Positionen von Image Targets in der Szene (+dazugehörigen Atomen) werden von Vuforia getracked.

Atom visualisiert Atome mit dem Bohr - Modell.

Die dafür benötigten Daten werden aus ScriptableObjects gelesen und die zur Visualisierung benötigte Sphere (Elektronen/Protonen/Neutronen) wird als Prefab geladen. Die Anordnung der Elektronen/Protonen/Neutronen passiert dynamisch auf Basis der Daten aus dem Scriptable Object.

Molecule ist ein Molekülbaukasten.

Wie auch in der Atom-Komponente werden die Daten aus ScriptableObjects gelesen. Die Visualisierung des Moleküls ist etwas fester geschrieben, das Modell für das Molekül liegt als Prefab vor.

TaskMode ist ein Aufgabenmodus.

Die Aufgaben (Auflistung siehe 1.6) greifen auf dieselben ScriptableObjects zu wie Atom und Molecule. Die Aufgabenlisten werden als JSON - Datei

gespeichert und wieder geladen (mit Hilfe von JSONUtility, einer Unity - Library). Über ein Skript mit dem Name "TaskModeManager" kommunizieren Atom und Molecule mit dem TaskMode und 'berichten' was in der Anwendung passiert.

5 Einschätzung

5.1 Aaron

Ich habe mir mit Youssef die Aufgaben für die Moleküle zusammen bearbeitet, welches wir hauptsächlich in Pair-Programming gemacht haben.

Die Moleküle waren teilweise schwierig zu dynamisch zu schreiben, und es gab auch oft Probleme mit der Implementierung von den Funktionen anderer (z.B. Atomspawn), welches aber schnell durch gute Kommunikation von allen schnell gelöst wurde. Die Erstellung war anfangs sehr statisch und hätte/sollte von Grundauf direkt dynamisch gestaltet werden sollen, allerdings gab es anscheinend bei Youssef sowie bei mir sehr viel Druck von anderen Fächern und deren Abgaben. Persönlich fand ich die Arbeit von meiner Seite aus zufriedenstellend, großer Aha-Moment dank Korbinians Hilfestellung bei der Umsetzung zur dynamik.

Tatsächlich war die größte Hürde meiner Meinung nach die Probleme die Vuforia verursacht hat, wie multiple Image Target tracking, sinnvolle implementierung von Vuforia im Quellcode.

5.2 Youssef

Ich habe mit Aaron an der Molecule Szene gearbeitet. Die Zusammenarbeit ist für mich ist generell problemlos gewesen. Wenn Schwierigkeiten gab, haben wir uns gegenseitig unterstützt. Für mich war die Implementierung der Logik für die Molecule Szene eine Herausforderung. Dabei hat uns Korbinian geholfen, um eine Lösung zu finden.

Da das Projekt relativ groß war und beim Erstellen des Projekts verschiedene Technologien genutzt wurden, habe ich viel neues gelernt.

5.3 David

Ich war mit dem Projekt persönlich und der Gruppendynamik insgesamt sehr zufrieden. Ich habe mich auf die Erstellung der Atom Szene konzentriert und konnte alles so umsetzen, wie ich es mir vorgestellt hatte, bevor ich begann. Bei der Arbeit an den Atomen hatte ich manchmal Probleme mit falschen Spawn-Punkten, konnte das Problem aber durch Diskussionen mit meiner Gruppe lösen. Meine Zeitplanung hätte etwas besser sein können, da ich kurz

vor der Präsentation noch an einigen Fehlerkorrekturen gearbeitet habe, aber insgesamt hat alles gut geklappt.

5.4 Korbini

Ich hatte einen guten Anteil am Projekt. Die Aufgaben waren soweit einfach zu erledigen und ich bin mit der Qualität relativ zufrieden.

Es gibt allerdings auch Bereiche wo ich noch etwas mehr Zeit hätte investieren können / ich mit der Lösung nicht ganz zufrieden bin:

Aufbau der Klassen "Task"/"TaskList", Tasklist saving, Lesbarkeit TaskModeManager, Schriftart der Tasklist im UI, Design der Image Targets.

Ich denke mein Input in den Gruppenmeetings war zielführend.

Meine Zeiteinteilung hätte sicherlich etwas besser sein können - ich bin glücklicherweise aber ohne all-nighter ausgekommen.

5.5 Gesamt

Das gesamte Projekt bewerten wir positiv.

Die Entwicklung verlief relativ flüssig - jegliche Probleme die aufgetreten sind konnten wir gemeinsam lösen. Wir haben uns seit Projektbeginn wöchentlich getroffen um den aktuellen Stand zu bereden.

Wir haben die Aufgaben unter uns sinnvoll verteilt und uns gegenseitig bei der Problembewältigung geholfen.

Die Gruppendynamik war für ein Uni - Projekt sehr gut.

Mit dem Endresultat sind wir zufrieden, es übertrifft sogar unsere Erwartungen.

Literatur

- [1] Unity. [https://de.wikipedia.org/wiki/Unity_\(Spiel-Engine\)](https://de.wikipedia.org/wiki/Unity_(Spiel-Engine)).
- [2] Vuforia. https://en.wikipedia.org/wiki/Vuforia_Augmented_Reality_SDK.
- [3] Git Deutsch. <https://de.wikipedia.org/wiki/Git>.
- [4] Git. <https://en.wikipedia.org/wiki/Git>.