

Konzept ITS / MOSY

*Von Mona Röttger (2220740),
Corinna Bohnenberger (2217555),
Tjarko Slomski (2182553)
Nico Flach (2175367)*

Projektziel

Das Projekt "QuickQuizCopter" wird im Rahmen der Veranstaltung ITS/MOSY entwickelt. QuickQuizCopter besteht aus einer Quiz-App, zwei Smartphones und zwei Flugobjekten. Die beiden Spieler spielen gegeneinander und müssen über die App Fragen beantworten. Bei jeder richtig beantworteten Frage steigt das Flugobjekt etwas nach oben. Gewonnen hat derjenige, der nach Ablauf der Zeit das höher fliegende Flugobjekt erreicht hat. Erreicht ein Spieler (auch vor Ablauf der Zeit) den höchsten Punkt, wird ein "Konfettiregen" ausgelöst und das Spiel ist damit (vorzeitig) beendet.

Anforderungsanalyse

Für dieses Projekt benötigen wir 3 Komponenten: Die Konstruktion mit den Flugobjekten und den Raspberry Pi's, die Android App auf zwei Smartphones und die Verbindung beider Komponenten über einen Server.

Bei dem Spiel geht es darum, dass zwei Spieler gegeneinander ein Quiz über eine Android-App auf ihren Smartphones spielen. Das User-Interface des Quiz' soll ähnlich wie bei QuizDuell aufgebaut sein, mit einer Frage und jeweils 4 Antwortmöglichkeiten, von denen nur eine die Richtige ist. Jeder Spieler hat eine vorgegebene Zeit, z.B. 2 Minuten (dies hängt noch von der Anzahl der Fragen und der Schnelligkeit des Flugobjektes ab, was wir praktisch testen müssen), in der er so viele Fragen wie möglich beantworten soll.

Wichtig dafür ist die Verbindung der beiden Spieler mit den jeweiligen Raspberry Pi's über den Server, sodass ein Signal bei der richtigen Beantwortung der Frage eines Spieler an den zugewiesenen Raspberry Pi gesendet wird.

Außerdem wird zu Beginn des Spiels ein Array mit Zufallszahlen (entsprechend der Anzahl der Fragen) über den Server generiert, der die Fragen aus der lokalen Datenbank der Smartphones auswählen soll, sodass beide Spieler in der App die gleiche Fragenreihenfolge angezeigt bekommen. Zusätzlich wird ein Timer gestartet (ca. 2 Minuten) und das Spiel beginnt.

Jetzt können die Spieler über die Quiz-App ihr Wissen unter Beweis stellen. Nach jeder beantworteten Frage wird die Antwort gewertet und lässt das Flugobjekt steigen oder nicht. Ein dazwischen geschalteter Zeitpuffer kann der Spieler den Spielfortschritt beobachten. Kurz danach wird die nächste Frage eingeblendet.

Für die Konstruktion werden für jeden Spieler zwei Rohre mit einer Skala auf eine Holzplatte befestigt, an denen das jeweilige Flugobjekt nach oben gleitet. Die Höhe des Objekts wird vermutlich über einen Infrarotsensor gemessen und an den Raspberry gesendet.

Technische Rahmenbedingungen

Konstruktion

Für jeden Spieler gibt es einen Aufbau mit jeweils einem Flugobjekt und einem Raspberry Pi. Das Flugobjekt besteht aus einem Motor mit Propeller, der je nach Drehgeschwindigkeit auf- und absteigen kann. Als Stabilisierung dienen zwei Rohre, die als Führung für den Motor dienen.

Die Ansteuerung des Motors erfolgt über eine H-Brücke, um die für den Motor benötigte Spannung zu erreichen.

Je nachdem, ob die Flugobjekte kabellos oder über Feststrom laufen, erfolgt die Stromversorgung über einen Akku oder ein Netzteil. Eine kabellose Konstruktion hat den Vorteil, dass sie schöner aussieht und möglicherweise weniger fehleranfällig ist. Dafür steigen die Kosten massiv. Eine weitere und etwas günstigere Lösung zur kabellosen Stromversorgung, ist die Nutzung von Schleifkontakten an den Rohren.

Wird eine Antwort richtig beantwortet, fliegt das jeweilige Flugobjekt eine Stück (z.B. ca 5 cm) höher, jedoch maximal bis es an der höchsten Stelle (Anschlag) angekommen ist. Wird die Antwort falsch beantwortet, bleibt das Flugobjekt auf der selben Höhe stehen.

Zur Kontrolle der Höhe wird ein Infrarot-Näherungssensor verwendet. Dies wird benötigt, um die Höhe des Flugobjektes zu kontrollieren und zu steuern.

App

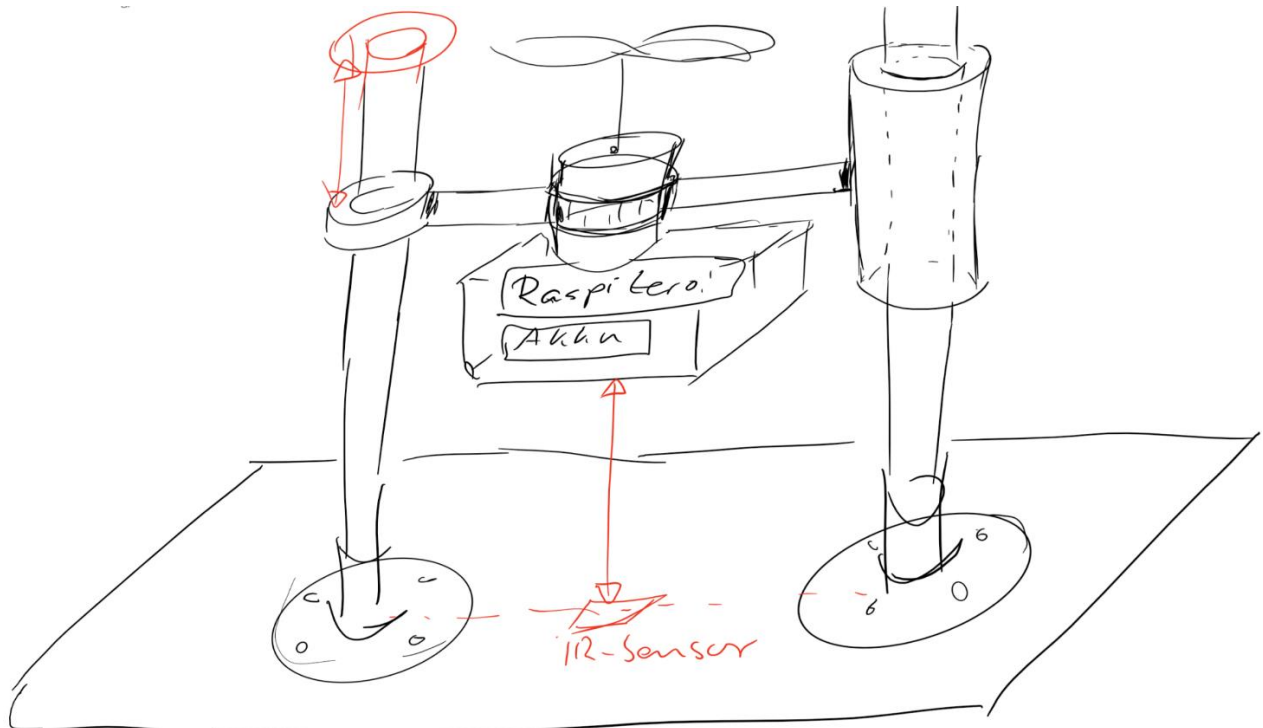
Da wir im Moment nur Zugang zu Mobiltelefonen mit der Betriebssoftware Android haben, erstellen wir mit Hilfe der Entwicklungsumgebung Android Studios eine Android-Applikation. Außerdem haben wir bereits Erfahrung mit Android Studios. Für die Datenbank werden wir die in Android-Geräten eingesetzte SQLite Database verwenden. In dieser werden die Daten lokal gespeichert.

Netzwerk / Server / Broker

Für die Verbindung zwischen Smartphone und Raspberry Pi wird der von der HAW bereitgestellten Server verwendet. Von der Android-Seite aus wird der Server über Java-Klassen und mithilfe des MQTT Protokolls angesprochen. Das Raspberry Pi kommuniziert mittels Python Skripte auch über MQTT mit dem Server. Die Teilnehmer können ihre Daten austauschen, indem sie Topics abonnieren oder schreiben. Dies haben wir bereits in der Vorlesung kennengelernt, müssen uns allerdings noch genauer einlesen.

Technisches Konzept

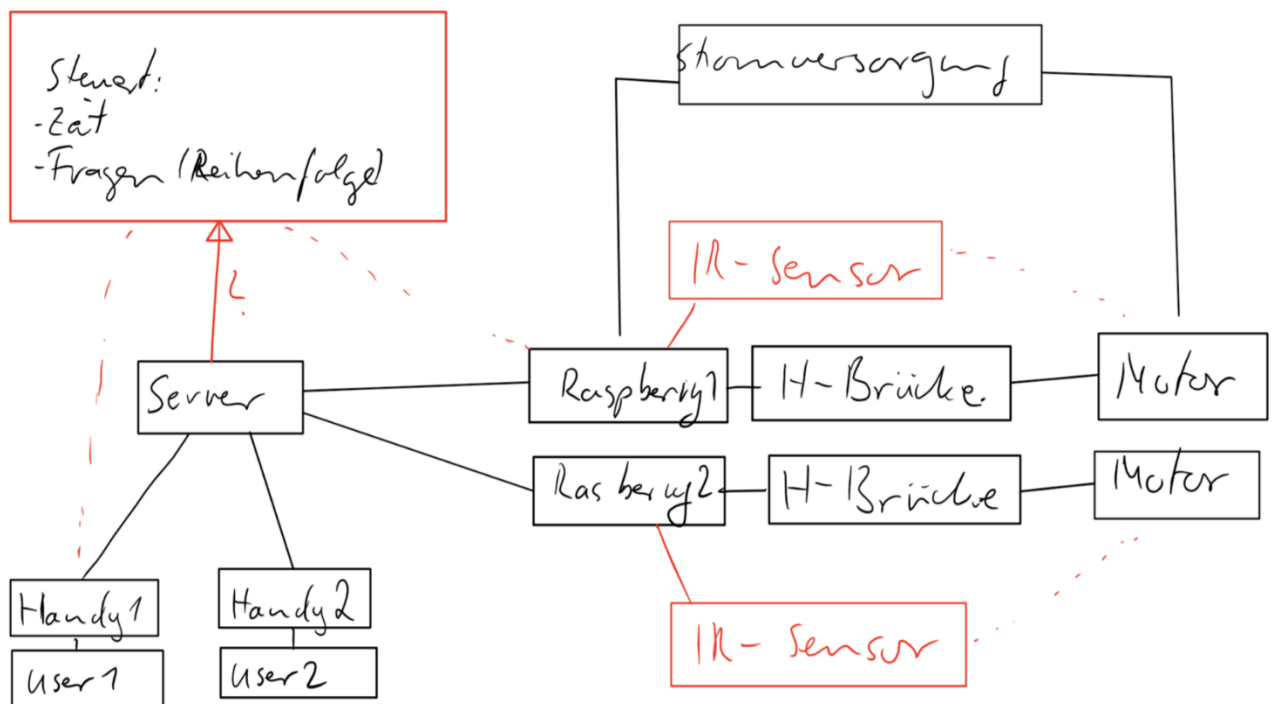
Abb. 1: Skizze Konstruktion



Die grobe Skizze verdeutlicht den Aufbau des Flugobjektes.

Auf der Bodenplatte sind zwei Rohre montiert, die als Führung für das Flugobjekt dienen. Das Flugobjekt besteht aus Motor und Propeller und ist mittels Schellen und Stangen in der Führung eingehängt.

Abb. 2: Skizze App/Verbindungen



Die Skizze verdeutlicht die Verbindungen der einzelnen Komponenten als eine Art Flussdiagramm.

Abb. 3: Wireframes

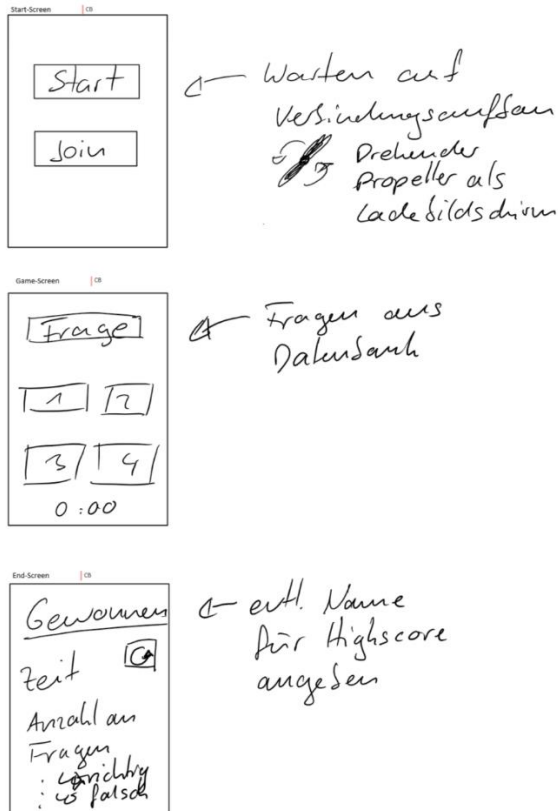
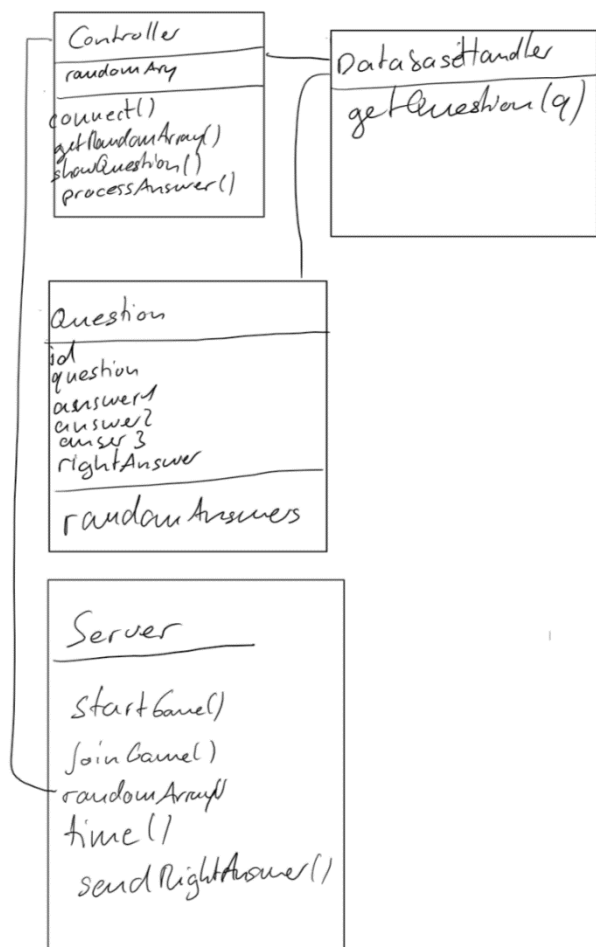


Abb. 4: UML - Diagramme

Die Klasse Controller erhält von der Klasse DatabaseHandler ein Array mit zufallsgenerierten Zahlen, sobald sich beide Spieler verbunden haben. Über die Zufallszahlen (ID) können die dazugehörigen Fragen aus der Datenbank gelesen werden. Wurde eine Antwort gewählt, wird diese über den Server an den Raspberry Pi gesendet und entsprechend weiterverarbeitet (Flugobjekt steigt oder auch nicht). In der Abbildung sieht man erste Gedanken zum Aufbau und Zusammenhang der Klassen



Bedienkonzept

Um an dem Spiel teilnehmen zu können benötigt der Nutzer ein Smartphone und eine von uns erstellte Software/Applikation. Diese Hardware und Software bekommen die Spieler in der Ausstellung gestellt. Die Applikation ist auf diesen Geräten bereits installiert und benötigt keine weiteren Einstellungen, sodass sie direkt mit dem Spiel starten können.

Öffnet der Spieler die App, wird er zunächst auf den Start-Screen geleitet. Dort kann er entweder ein Spiel starten oder einem bereits von einem anderen User gestarteten Spiel beitreten. Während des Verbindungsaufbaus wird ein Ladebildschirm mit einem sich drehenden Propeller angezeigt. Sind zwei Spieler einem erstellten Spiel beigetreten, bekommen beide die gleichen Fragen freigeschaltet. Der Timer wird gestartet und das Quiz beginnt.

Jetzt müssen die Spieler so viele Fragen wie möglich in einer vorgegebenen Zeit beantworten. Die noch verbleibende Zeit wird dem Spieler auf dem Display mitgeteilt. Die Fragen bekommen die Spieler unabhängig von einander über den Game-Screen angezeigt. Es wird jeweils eine Frage mit den dazugehörigen vier Antwortmöglichkeiten dargestellt. Der Spieler wählt nun seiner Ansicht nach die richtige Antwort aus. Ist diese tatsächlich korrekt, steigt sein Flugobjekt an. Damit der Spieler dies beobachten kann, gibt es nach jeder Frage einen kleinen Zeitpuffer. Danach wird die nächste Frage eingeblendet. Falls die Frage falsch beantwortet wurde, bleibt das Flugobjekt auf dem aktuellen Stand.

Ist die Zeit abgelaufen oder das Flugobjekt eines Spielers ist noch vor Ablauf der Zeit oben angekommen, stoppen diese und der End-Screen wird eingeblendet. Der Spieler mit dem höher angestiegenen Flugobjekt hat gewonnen. Zusätzlich erhalten die Spieler über den End-Screen die Information, ob sie gewonnen oder verloren haben. Außerdem erhält der Spieler Angaben über die Fragen z.B. wie viele Fragen hat er beantwortet und wie viele davon richtig. Ein Button auf dem End-Screen ermöglicht einen Neustart.

Zeitplan

Aufwand: Ca 90 Std. pro Person. Also bei 3 Personen 360 Std.

	AKTIVITÄT	ANFANG	ENDE	NOTIZEN
	Projektanfang	02.05.17		
	Konzept abgeben	02.05.17	02.05.17	
	Material besorgen	02.05.17	09.05.17	
	Grober Aufbau (Hardware & Software)	09.05.17	16.05.17	
	Bewegung (Auf/Ab) von Raspi + Datenbank	09.05.17	23.05.17	
	Datenbank Verbindung App	09.05.17	23.05.17	
	Probeaufbau "fertig"	16.05.17	23.05.17	
	Prototyp	23.05.17	30.05.17	
	Alle Verbindungen App->Server->Raspi	23.05.17	10.06.17	
	Verbesserungen	10.06.17	20.06.17	

	Generalprobe	20.06.17	20.06.17	
	Erstellung Plakat/Aufsteller/...	20.06.17	12.07.17	
	Jahresausstellung/Präsi	12.07.17	14.07.17	
	Doku ausarbeiten	14.07.17	20.07.17	
	Projektende/Abgabe	20.07.17	20.07.17	

