

Konzeptioneller Ansatz

Carrier Tracking

Unity Version zu Projektbeginn: 2019.4.11

Im Laufe des Projektes umgestiegen auf 2019.4.13f1 durch die Vermutung auf Fehlerquelle in Zusammenhang mit JSON, für das Speichern und Laden der Game- und SettingsData.

Microsoft .NET Framework zu Projektbeginn: 2.X

Um die Funktionen der ZXing Bibliothek für die QR-Code Verarbeitung nutzen zu können, wurde auf die Version 4.X geupdatet.

Interessante Punkte:

- Das Anlegen eines virtuellen Raumes in Unity, um die entsprechenden QR-Codes in Form von Models – zunächst nur simplen Rechtecken – darzustellen.
- Testen von Bildern aus der realen Umgebung, in welcher die QR-Codes in verschiedenen Größen, Lichtverhältnissen und Distanzen angebracht waren, um dem Endkunden reale und begründete Anforderungen an die Kamerabilder präsentieren zu können.
- Serialisierung von Objekten, welche von den Benutzern während der Nutzung der Software erstellt wurden.
- Neueinstieg in eine Entwicklungsumgebung, welche sowohl 3D, als auch 2D Entwicklung unterstützt.
- Anwendung klassischer Softwarepatterns wie z.B. dem MVC, oder Singleton
- Parallele Softwareentwicklung mit Hilfe von SCRUM durch mehrere Developer

Weitere Punkte welche im Backlog vorgesehen waren aber aus zeitlichen oder technischen Gründen nicht umgesetzt wurden:

- Import eigener 3D Modelle für Ladungsträger zur besseren Nachverfolgung dieser in der Stationsansicht
- Speicherort der Kamerabilder, sowie QR-Codes frei durch den Benutzer wählbar
 - Sowohl Kamerabilder, als auch QR-Codes werden im Programmpfad in den entsprechend dazu automatisch erstellten Ordnern abgelegt.
- Zur visuellen Unterstützung, Wände frei einzeichnen, um den Raum, bzw. die Trennung zweier Stationen, welche durch eine Kamera abgedeckt werden, zu verdeutlichen.

Herausforderungen und Probleme:

Als zentrale Schnittstelle zum Austausch des jeweils aktuellen Softwarestandes wurde GitHub gewählt, um das parallele Arbeiten durch mehrere Entwickler gleichzeitig zu ermöglichen. Hierbei wurden jedoch mehrfach Commits verfälscht und resultierten in verlorenen Code-Snippets, oder Funktionen, die manuell nachgetragen werden mussten. Es hat sich als sehr empfehlenswert und von Vorteil erwiesen, auf lokale Backups zu setzen, um ggfs. Änderungsvorschläge seitens GitHub-Desktop zu verwerfen und aus dem Backup nachzuführen.

Die Kenntnisstände der Gruppenmitglieder waren teils sehr unterschiedlich, vor allem im Bezug auf Erfahrungen in der Softwareentwicklung, oder dem Arbeiten in größeren Gruppen. Es war demnach ein grundlegender Aspekt und daher eine Schwierigkeit, alle auf einem Stand einzupendeln, damit eine Zusammenarbeit möglichst reibungslos funktionieren kann.

Aufgrund der momentanen Coronasituation wurden Präsenzveranstaltungen zu einem schwierigen Thema und konnten sehr selten durchgeführt werden. Zu Beginn des Projektes hat sich das komplette Team getroffen, um ein einheitliches Bild von der Problemstellung und entsprechenden Lösungsansätzen zu erörtern. Nach diesem Treffen begrenzte sich jedoch die Zusammenarbeit auf Online-Meetings und Absprachen durch Messenger-Dienste. So konnten wir auch die eigentlichen Einsatzorte für die Software nicht besichtigen und mussten uns aus den Informationen, die uns das EvoBus-Team zur Verfügung gestellt hat, ein eigenes Bild über die Räumlichkeiten machen.

Da im bisherigen Studienverlauf, die meisten Projekte in Java geschrieben werden mussten, war der Umstieg auf C# zu Beginn nicht für alle einfach. Die Eingewöhnungszeit verzögerte sich leicht durch die Tatsache, dass Java und C# über unterschiedliche Syntax und Konventionen verfügen.

Eine weitere Hürde bot sich dem Team durch den Vorsatz, die Software nach Abschluss auf verschiedenen Betriebssystemen anbieten zu können. So hatten wir früh gemerkt, dass MacOS, Probleme mit der Pfadangabe aus Windows hat und mussten eine Lösung finden, die die freie Pfadangabe auf beiden Systemen ermöglicht. Da ein Mitglied des Entwicklungsteams, Unity auf MacOS betrieben hat wurden uns solche Hürden schnell aufgewiesen, jedoch war die Lösung auf diese nicht immer offensichtlich und hat teilweise mehr Zeit beansprucht, als anfangs vorgesehen.