

Anforderungsspezifikationen

Projekt: Street Sim

Matteo Bentivegna, Sandra Kiefer, Jan Ningelgen, Niklas Schlögel



Hochschule RheinMain

Sommersemester 2020

Medieninformatik - Softwaretechnik

Projektgrundlagen	2
Einführung, Ziele und Rahmenbedingungen	2
Abläufe und Funktionen	3
Typische Interaktionen	3
Relevanz für Zielgruppen	4
Analyse der verfügbaren Funktionen	5
Funktionale Anforderungen	12
Nichtfunktionale Anforderungen	12
Entitäten und Beziehungsdefinitionen	13
„Gegenstandswelt“ des Systems	13
Grafische Darstellung	14
Aufbau der Szenen (Sitemap)	14
Benutzungsoberfläche (Wireframes)	14

Projektgrundlagen

Einführung, Ziele und Rahmenbedingungen

Ziel ist es einen Verkehrssimulator zu entwickeln, welcher es ermöglicht, ein individuelles Verkehrsnetz aus Straßenteilen, wie z.B. Geraden, Kreuzungen und Kurven über einen grafischen Editor zu erzeugen. Dem Verkehrsnetz sollen zusätzlich Verkehrsteilnehmer hinzugefügt werden können, welche mit individuell einstellbaren Geschwindigkeiten, das simulierte Fahren darstellen. Darüber hinaus werden Ampeln, sowie Verkehrsregeln und Kollisionserkennung berücksichtigt um die Simulation realistischer zu machen.

Abläufe und Funktionen

Typische Interaktionen

Zur Veranschaulichung werden verschiedene Anwendungsszenarien formuliert. Somit werden die Funktionen der Anwendung konkret beschrieben.

1. Fahrlehrer

Der Fahrlehrer Bernd ist auf der Suche nach einem Programm, welches seinen Fahrschülern die Abläufe im Straßenverkehr demonstriert. Durch die individuellen Anpassungen des Straßennetzes ist es ihm möglich, die unklaren Situationen seinen Fahrschülern näher zu erklären. Für ihn wäre es ebenfalls sehr wichtig und hilfreich, wenn noch andere Verkehrsteilnehmer wie Fußgänger und Fahrradfahrer hinzugefügt werden könnten. Durch das Abspeichern von bereits aufgebauten Situationen im simulierten Straßenverkehr kann er seinen Fahrschülern schneller helfen.

2. Schüler

Paul geht in die dritte Klasse und ist gerade dabei seinen „Fahrradführerschein“ zu absolvieren. Weil seine Eltern sicher gehen wollen, dass Paul den Straßenverkehr richtig einschätzen kann, zeigen sie ihm das Programm. Dabei kann er spielerisch durch individuelles Zusammensetzen der eigenen Strecken lernen, wie sich zum Beispiel Linksabbieger verhalten. Besonders viel Spaß macht es Paul, ihm bereits bekannte Straßennetze aus der Realität im Programm nachzubauen. Ebenfalls findet er es amüsant, wenn so viele Autos wie möglich platziert werden. Dadurch kann er ebenfalls nachvollziehen wie Staus entstehen können und auch die damit verbundene Unfälle.

3. Verkehrsplanung

Die Landesverkehrsministerin Andrea möchte für eine Stadt in ihrem Bundesland die optimale Beampelung finden. Sie baut in dem Simulationsprogramm einen viel Befahrenen Verkehrsbereich nach. Sie setzt entsprechend viele Autos auf der hauptsächlich befahrenen Straße ab und erwartet einen sofortigen Ablauf der Simulation mit einer maximalen Wartezeit von 4 Sekunden. Sie erkennt problematische Stellen und kann durch neues Hinzufügen von Straßenabschnitten und zusätzliche Ampeln Staus lösen und Aufnahmen des Straßennetzes an Baufirmen weiterleiten.

4. Hobby

Gustav ist ein großer Motorsport-Fan und will seine Lieblings Rennstrecke nachbauen. Gustav startet die Anwendung und landet im Home-Menü. Da er schon einen Speicherstand angelegt hatte, möchte er diesen überarbeiten und lädt ein Spiel. Beim letzten Mal hat er die Rennstrecke erfolgreich nachbauen können, ihm fehlen nur noch Rennautos. Dafür fügt Gustav drei Autos hinzu. Für jedes Auto stellt er eine andere Geschwindigkeit ein und startet die Simulation. Nach einigen erfolgreichen Runden kommt es zu einer Kollision zwischen zwei Autos. Dabei wird ihm eine Kollisionswarnung angezeigt. Gustav speichert seinen Stand und schließt das Programm.

Relevanz für Zielgruppen

Die typischen Interaktion zeigen, auf welchen Funktionen der Applikation der Schwerpunkt liegt. Die Implementierung der Geschwindigkeitsanpassung wird in diesem Fall keine hohe Priorität zugeteilt. Das Hinzufügen von Straßenabschnitte dagegen ist essentieller Bestandteil welcher unbedingt solide umgesetzt werden sollte.



Analyse der verfügbaren Funktionen

Prioritätsklasse I:

Diese Funktionen sind essentiell für jeden Nutzer der Applikation. Jeder mögliche Fehlerfall muss abgedeckt werden. Der Zugriff auf diese Funktionen sollte mit sehr wenigen Navigationsschritten möglich sein.

Klasse I.A

Titel	Hinzufügen neuer Straßenabschnitte
Akteure	Bernd, Andrea, Paul und Gustav
Fachlicher Auslöser	User Interaktion
Vorbedingungen	Anwendung und Welt ist vollständig geladen und Anwendung ist im Editiermodus
Standardablauf	<ol style="list-style-type: none"> 1. Anwender wählt aus Menü den Punkt „Neuer Straßenabschnitt“ aus 2. System listet mögliche Straßenformen (gerade Straße, Kurve, Kreuzung) 3. Anwender wählt eine der Optionen aus und zieht ein Straßenstück in die Welt 4. System prüft Position und setzt Straßenstück ab
Alternative Abläufe Fehlersituationen Sonderfälle	<ol style="list-style-type: none"> 4a. Position ist schon belegt <ol style="list-style-type: none"> 4aa. System markiert durch graphischen Hinweis (rote Färbung im „Ablagevorgang“), dass das Straßenstücke dort nicht platziert werden 4aaa. Anwender versucht eine andere Position aus (4) 4aab. Anwender bricht Vorgang ab 4ab. System platziert Straßenstück, die Straße ist jedoch ohne Verbindung zu den bereits vorhandenen Straßenteilen <ol style="list-style-type: none"> 4aba. System zeigt betroffenes Straßenstück durch rote Färbung an 4abaa. Anwender rotiert Straßenstück in richtige Position 4abab. Anwender löscht Straßenstück
Nachbedingungen/Ergebnis	System verlässt Prüfzustand
Nicht-Funktionale Anforderungen	verzögerungsfreie Menünavigation und Prüfung
Parametrisierbarkeit Flexibilität	
Nutzungshäufigkeit Mengengerüst	beliebig oft

Das Hinzufügen neuer Straßenabschnitte ist die Kernfunktion der Applikation. Sie ist relevant für ausnahmslos jeden Nutzer. Das Auslösen dieser Funktion ist während einer laufenden Simulation nicht möglich. Straßenabschnitte nachträglich verschieben zu können ist in zukünftigen Versionen geplant. Das Rotieren von Straßenabschnitten muss gewährleistet werden.

Klasse I.B

Titel	Hinzufügen von Autos
Akteure	Bernd, Andrea, Gustav und Paul
Fachlicher Auslöser	Individueller Aufbau seines eigenen Straßennetzes
Vorbedingungen	Straßennetz wurde bereits angelegt
Standardablauf	<ol style="list-style-type: none"> 1. laufende Simulation pausieren 2. Fahrzeug auswählen 3. Fahrzeug an passender Stelle auf Straßenabschnitt platzieren 4. System überprüft gewünschte Platzierung 5. Simulation starten
Alternative Abläufe Fehlersituationen Sonderfälle	<ol style="list-style-type: none"> 3a. Platzierung von Autos soll korrigiert werden <ol style="list-style-type: none"> 3aa. Platzierungsvorgang wird erneut gestartet (Fahrzeug Auswahl beibehalten) 4a. Platzierung an gewünschter Stelle kann nicht realisiert werden <ol style="list-style-type: none"> 4aa. (z.B. in Kurven, auf Kreuzungen, andere Fahrzeuge sind bereits dort) Auto wird stattdessen an nächstmöglicher Stelle platziert
Nachbedingungen/Ergebnis	System fügt das Auto dem Straßennetz hinzu
Nicht-Funktionale Anforderungen	Reaktionszeit < 2 Sekunden
Parametrisierbarkeit Flexibilität	individuelle Positionierungen
Nutzungshäufigkeit Mengengerüst	beliebig oft

Das Hinzufügen von Autos ist eine weitere sehr wichtige Funktion. Viele folgende Funktionen bauen auf ihrer zuverlässigen Funktionsweise auf, wie beispielsweise das Starten einer Simulation (Klasse II.A) oder die Kollisionserkennung (Klasse III.A). Eine besondere Schwierigkeit stellt die korrekte Positionsberechnung in Abhängigkeit von Fahrrichtung und Art des Straßenabschnittes dar.

Prioritätsklasse II:

Diese Funktionen sind essentiell für viele Nutzer der Applikation. Die schwerwiegendsten Fehlerfälle müssen, sämtliche weiteren sollten abgedeckt werden. Der Zugriff auf diese Funktionen sollte mit wenigen Navigationsschritten möglich sein.

Klasse II.A

Titel	Starten und Pausieren der Simulation
Akteure	Andrea, Bernd und Gustav
Fachlicher Auslöser	Verkehrssimulation soll zu einem Zeitpunkt genauer betrachtet werden
Vorbedingungen	Entwurf ist erstellt oder geladen und Autos oder Ampeln sind bereits platziert
Standardablauf	1. Nutzer wählt Option Play/Start 2. Simulation startet, Autos und Ampeln sind platziert 3. Nutzer wählt Option Pause 4. Simulation friert ein, Autos und Ampeln verweilen in jeweiliger Lage
Alternative Abläufe Fehlersituationen Sonderfälle	1a. Entwurf enthält weder Autos noch Ampeln 1aa. Hinweis (Keine Simulation möglich)
Nachbedingungen/Ergebnis	
Nicht-Funktionale Anforderungen	Real Time Starten/Stoppen der Simulation
Parametrisierbarkeit Flexibilität	
Nutzungshäufigkeit Mengengerüst	beliebig

Das Starten und Pausieren einer Simulation ist eine aufwändige und wichtige Funktion. Erst in ihrer Durchführung werden hinzugefügte Ampeln und Autos dynamisch und automatisiert auf das erstellte Straßennetz reagieren.

Klasse II.B

Titel	Hinzufügen von Ampeln
Akteure	Bernd, Andrea und Paul
Fachlicher Auslöser	Platzierung von Ampeln an Straßen oder Kreuzungen
Vorbedingungen	Straßen müssen platziert worden sein
Standardablauf	<ol style="list-style-type: none"> 1. Ampel wird ausgewählt und an gewünschter Stelle platziert 2. Überprüfung ob die Ampel an dieser Stelle platziert werden kann 3. Ampel wird akzeptiert und eingeschaltet 4. Verkehr passt sich der Regulierung der Ampel an
Alternative Abläufe Fehlersituationen Sonderfälle	<ol style="list-style-type: none"> 2a. Positionierung der Ampel wird abgelehnt <ol style="list-style-type: none"> 2aa. Position wird rot markiert 2ab. nächstmögliche Positionen werden grün markiert 4a. Verkehr ignoriert die Ampelschaltungen (Fehlersituation)
Nachbedingungen/Ergebnis	Ampel wurde erfolgreich hinzugefügt
Nicht-Funktionale Anforderungen	
Parametrisierbarkeit Flexibilität	<ul style="list-style-type: none"> - Geschwindigkeitsanpassung der verschiedenen Ampelphasen/Schaltungen - komplett individuelle Positionierung der Ampeln
Nutzungshäufigkeit Mengengerüst	Maximale Anzahl von Ampeln

Das Hinzufügen von Ampeln zum bestehenden Straßennetz ist eine häufig genutzte Funktion. Die automatisierten Schaltphasen helfen Nutzern aller Art bei der Verbesserung des Straßennetz-Designs. Dennoch ist die Applikation auch ohne diese Funktion, aufgrund der Umsetzung von Vorfahrtsregeln, uneingeschränkt nutzbar.

Klasse II.C

Titel	Erstellen / Laden / Speichern von Entwürfen
Akteure	Andrea, Bernd und Gustav
Fachlicher Auslöser	
Vorbedingungen	verfügbare Dateien und bereits platzierte Objekte
Standardablauf	<ul style="list-style-type: none"> - Erstellen <ol style="list-style-type: none"> 1. das Programm wird gestartet 2. gewünschtes Objekt wird ausgewählt (Straße, Auto, ...) 3. Objekt wird an gewünschter Stelle platziert 4. Schritt (2) und (3) beliebig oft wiederholen - Speichern <ol style="list-style-type: none"> 1. nach dem Erstellen die Option speichern auswählen 2. Datei benennen 3. Speichern bestätigen - Laden <ol style="list-style-type: none"> 1. nach dem Programmstart Option „Datei Laden“ auswählen 2. entsprechender Datei aus verfügbaren Dateien auswählen 3. Auswahl bestätigen 4. Entwurf wird geladen und angezeigt
Alternative Abläufe Fehlersituationen Sonderfälle	<ul style="list-style-type: none"> - Erstellen <ol style="list-style-type: none"> 3a. an dieser Stelle befindet sich bereits ein Objekt <ol style="list-style-type: none"> 3aa. System fragt ob überschrieben werden soll - Speichern <ol style="list-style-type: none"> 3a. Datei existiert bereits <ol style="list-style-type: none"> 3aa. System verlangt eindeutigen Namen 3b. Datei ist leer <ol style="list-style-type: none"> 3ba. System bricht Speichervorgang mit Hinweis ab - Laden <ol style="list-style-type: none"> 2a. es existieren keine kompatiblen Dateien <ol style="list-style-type: none"> 2aa. System bietet standardmäßig Abbruch an 3a. Auslesen der Datei schlägt fehl <ol style="list-style-type: none"> 3aa. Hinweis und Abbruch 3ab. „Datei Laden“ Option weiterhin möglich
Nachbedingungen/Ergebnis	Entwurf erstellt / Entwurf geladen / Entwurf gespeichert
Nicht-Funktionale Anforderungen	
Parametrisierbarkeit Flexibilität	
Nutzungshäufigkeit Mengengerüst	immer

Das Speichern und Laden von Entwürfen ist eine reine Komfortfunktion. Sie wird von Nutzern sehr geschätzt, beeinträchtigt die Funktionsweise der Applikation jedoch in keiner Weise.

Prioritätsklasse III:

Diese Funktion ist essentiell für wenige Nutzer der Applikation: Die schwerwiegendsten Fehlerfälle müssen abgedeckt werden. Der Zugriff auf diese Funktion sollte mit nachvollziehbaren Navigationsschritten möglich sein.

Klasse III.A

Titel	Kollisionserkennung
Akteure	Gustav und Paul
Fachlicher Auslöser	Kollisionen zwischen Verkehrsteilnehmer
Vorbedingungen	Verkehrsteilnehmer auf Straßen und unterschiedliche Geschwindigkeiten
Standardablauf	1. Verkehrsteilnehmer fahren auf Straßen 2. Kollision zwischen zwei Verkehrsteilnehmer 3. Verkehrsteilnehmer tragen Schaden 4. kleiner Stau wird gebildet 5. Stau wird schnell gelöst
Alternative Abläufe Fehlersituationen Sonderfälle	2a. Kollision auf Kreuzung 2aa. Stau an mehreren Straßen 2ab. Stau braucht länger um gelöst zu werden 2b. Verkehrsteilnehmer fahren ineinander rein (Fehlersituation)
Nachbedingungen/Ergebnis	System macht User auf Kollision aufmerksam
Nicht-Funktionale Anforderungen	kollidierte Verkehrsteilnehmer rauche/ brennen
Parametrisierbarkeit Flexibilität	- Kollision hängt von der Hitbox der Verkehrsteilnehmer ab - Schaden hängt von der Geschwindigkeit der Verkehrsteilnehmer ab - Stau hängt von der Dauer der Auflösung des Unfalls ab
Nutzungshäufigkeit Mengengerüst	mindestens zwei Autos müssen auf den Straßen platziert worden sein

Die Kollisionserkennung kommt nur bei Fehlern des automatisierten Fahrens oder falsch geschalteten Ampeln zum tragen. In zukünftigen Versionen zudem noch bei zufälliger Unachtsamkeit der Fahrer oder beim eigenständigen Lenken eines Fahrzeuges.

Klasse III.B

Titel	Anpassung der Geschwindigkeit
Akteure	Gustav
Fachlicher Auslöser	vordefinierte Geschwindigkeitseinstellung ist für die individuelle Nutzung nicht geeignet
Vorbedingungen	verschiedene Verkehrsteilnehmer sind bereits auf dem Straßennetz platziert worden und gestartet worden
Standardablauf	<ol style="list-style-type: none"> 1. laufende Simulation pausieren 2. anzupassendes Fahrzeug auswählen 3. eingestellte Geschwindigkeit verringern oder erhöhen 4. Abspeichern der eingestellten Geschwindigkeit
Alternative Abläufe Fehlersituationen Sonderfälle	<ol style="list-style-type: none"> 3a. eingestellte Geschwindigkeit ist bereits Maximum/Minimum <ol style="list-style-type: none"> 3aa. gewünschte Anpassung der Geschwindigkeit kann nicht realisiert werden 4a. eingestellte Geschwindigkeit soll doch nicht übernommen werden <ol style="list-style-type: none"> 4aa. vorherige Einstellungen werden wieder geladen und beibehalten
Nachbedingungen/Ergebnis	System übernimmt eingestellte Geschwindigkeit für das ausgewählte Fahrzeug
Nicht-Funktionale Anforderungen	Wertebereich im realistischen Bereich halten
Parametrisierbarkeit Flexibilität	eingeschränkter Wertebereich zur Anpassung
Nutzungshäufigkeit Mengengerüst	beliebig oft

Die Anpassung der Geschwindigkeit einzelner Autos ist für wenige Nutzer interessant. Zu beachten ist das automatisierte Drosseln bei zu nahem Auffahren.

Funktionale Anforderungen

Die Funktionalen Anforderungen fassen alle wesentlichen Dienste und Funktionen des Systems zusammen. Darunter findet man das Hinzufügen neuer Straßenabschnitte, Autos und Ampeln, das Anpassen der Geschwindigkeit von Autos und deren Kollisionserkennung, die Erstellung, das Laden und Speichern von Entwürfen, und das Starten und Pausieren der Simulation.

Das System sollte sich den einzelnen Situationen entsprechend richtig verhalten. Dazu gehören das richtige Abbiegen in eine Straße und die dazugehörigen Vorfahrtsregelungen.

Falls es zu einer Kollision kommt, sollten die beteiligten Verkehrsteilnehmer einen Schaden davon tragen, wie z.B. langsames Fahren. Alle anderen Verkehrsteilnehmer sollten ebenso davon beeinflusst werden, indem es z.B. kurzzeitig zu einem Stau kommt.

Nichtfunktionale Anforderungen

Nichtfunktionale Anforderungen gehen über die funktionalen Anforderungen hinaus und beschreiben, inwiefern das System eine Leistung erbringt.

In diesem Fall beinhaltet das das Rauchen von Autos bei Kollisionen, die generelle Animation von Autos und Ampeln, und das Zusammenstellen des Straßennetzes per Drag and Drop.

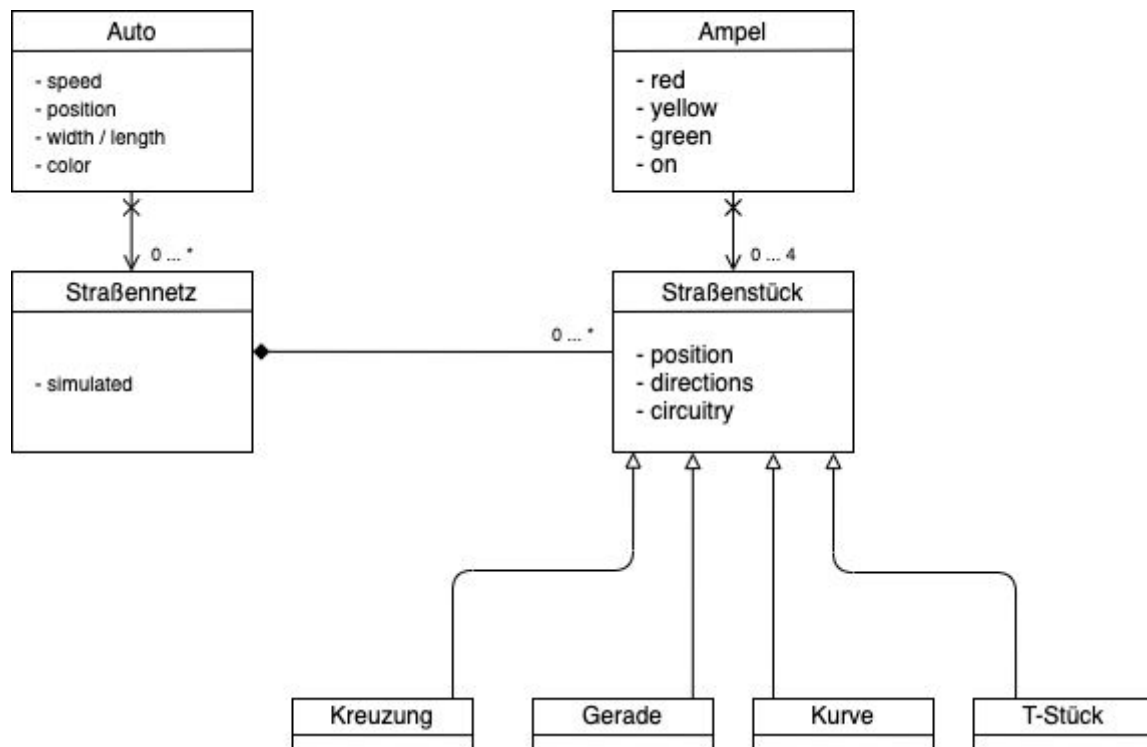
Zu den technologischen Anforderungen zählen die Mindestanforderung, welche bestimmen, wie das System ausgeführt werden kann.

Die Qualitätsanforderungen sind ebenso nichtfunktionale Anforderungen und bestehen aus kurzen Ladezeiten von Speicherständen, der real-time Platzierung von Straßennetz, Ampeln und Autos, Menünavigation, und das Starten/Stoppen der Simulation.

Entitäten und Beziehungsdefinitionen

„Gegenstandswelt“ des Systems

Hierbei handelt es sich um das Domänenmodell, was die zueinander in Beziehung stehenden Entitäten unseres Systems darstellt.



Die Beziehung zwischen Auto und Straßennetz zeigt, dass beliebig viele Autos zu einem Straßennetz hinzugefügt werden können, aber kein Straßennetz zu Autos.

Die Beziehung zwischen Ampel und Straßenstück zeigt, dass keine, oder maximal vier Ampeln zu einem Straßennetz hinzugefügt werden können, aber kein Straßenstück zu Ampeln.

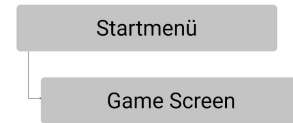
Die Beziehung zwischen Straßennetz und Straßenstück zeigt, dass beliebig viele Straßenstücke zu einem Straßennetz hinzugefügt werden können und diese Existenzabhängig vom Straßennetz sind.

Ein Straßenstück kann eine Kreuzung, Gerade, Kurve oder ein T-Stück sein.

Grafische Darstellung

Aufbau der Szenen (Sitemap)

Jeder Nutzer startet auf dem Startmenü. Von dort aus gelangt man auf das Spielfeld, auf dem weitere Funktionen getätigt werden können.



Benutzungsoberfläche (Wireframes)

Startmenü

Das Startmenü zeigt das Logo des Spiels und die Aufforderung einen beliebigen Knopf zu betätigen, damit das Spiel startet. Anschließend werden dem Benutzer zwei Möglichkeiten angeboten.

Entweder kann der Benutzer ganz neu anfangen und sein eigenes Straßennetz mit verschiedenen Komponenten aufbauen.

Oder der Benutzer hat bereits ein Projekt erstellt und möchte nun dieses laden, damit er es nach Belieben verändern und anpassen kann.

Nach Abschluss dieser möglichen Interaktionen gelangt der Benutzer zum Spielfeld.



Spielfeld

Im folgenden Bild wird ein beispielhafter Aufbau eines Spielfeldes dargestellt. Der Hintergrund ist ein statisches Bild. Darauf können unabhängig die verschiedenen Elemente des Spiels platziert werden. Zu sehen sind verschiedene Straßenabschnitte. An den Kreuzungen wurden ebenfalls Ampeln platziert. Auf den Straßen fahren verschiedene Autos. Am link unteren Bildschirmrand befindet sich die Navigation des Spiels. Hiermit kann das Spiel pausiert werden, verschiedene Elemente gelöscht werden und das Spiel geschlossen werden. Hierbei erfolgt die Frage, ob das aktuelle Spielgeschehen gespeichert werden soll. Durch ein Popup auf der rechten Seite des Bildschirms können die verschiedenen Elemente im Spielfeld ausgewählt und platziert werden.

