

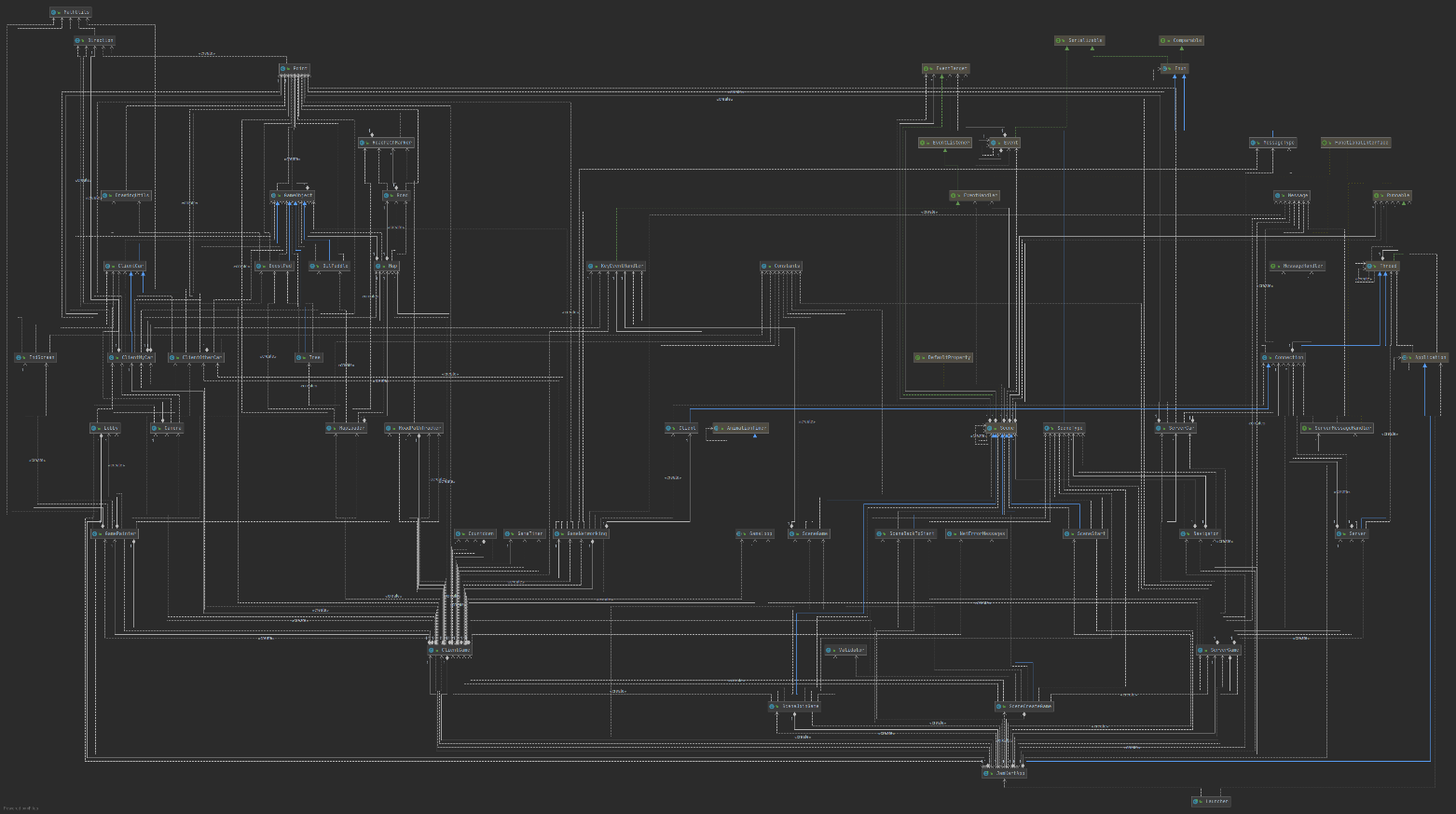
JAM-Kart

Klassendiagramm

Version 1.0.0, 17. Februar 2021 | Projektteam: Jan Henke, Alain Roser, Marino von Wattenwyl

# OO Analyse- / Design

## Klassendiagramm



## Beschreibungen

### BoostPad

Ein GameObjekt, das sich auf der Strecke befindet und in eine bestimmte Richtung gedreht ist. Autos, die darüberfahren, werden davon in die Richtung beschleunigt, in die es gerichtet ist.

### Camera

Repräsentiert eine virtuelle Kamera. Eine Kamera hat einfach eine Position und wird durch die «update»-Methode auf das Zentrum des eigenen Autos ausgerichtet. Beim Zeichnen wird die Kamera dann dadurch umgesetzt, dass die gesamte Welt entgegen der Position der Kamera verschoben wird.

### Client

Ein Client ist eine Connection, doch man gibt im Konstruktor die Ziel-IP-Adresse und den Zielport mit anstelle eines Sockets. Der Client erstellt dann selbst den Socket und gibt ihn an den Superkonstruktor mit.

### ClientCar

Repräsentiert ein Auto auf dem Client. Diese Klasse beinhaltet Eigenschaften und Methoden, die alle Autos auf dem Client gemeinsam haben wie einen Namen, eine ID, ein Bild etc.

### ClientGame

Das ClientGame ist der Startpunkt eines Spiels auf dem Client. Hier werden alle Objekte erstellt, die für ein Game auf dem Client notwendig sind wie der Painter, der Loop, die Map oder das eigene Auto.

### ClientMyCar

Der ClientMyCar ist das Auto auf dem Client, das sich vom Spieler steuern lässt. Er erbt von ClientCar, beinhaltet aber zusätzlich dazu noch allen Code zur Steuerung des Cars.

### ClientOtherCar

Der ClientOtherCar ist ein Auto auf dem Client, das nicht vom eigenen Spieler, sondern von anderen Mitspielern gesteuert wird. Um die Bewegungen einigermassen flüssig aussehen zu lassen, werden Position und Rotation interpoliert ([Interpolation (Wikipedia)](https://de.wikipedia.org/wiki/Interpolation_(Mathematik))), während auf die nächste Nachricht vom Server gewartet wird.

### Connection

Eine Connection ist ein Kommunikationsendpunkt über das Netzwerk. Über eine Connection können Messages geschickt werden und MessageHandlers für das Empfangen und Verarbeiten von Nachrichten festgelegt werden. Im Hintergrund steht dabei ein Java Socket.

### Constants

Constants beinhaltet unerwarteterweise Konstanten, die sich keiner Klasse zuordnen lassen. Beispiele dafür wären die Breite und Höhe des Startfensters oder das Zeitintervall, nachdem der Client den Status seines Autos an den Server übermittelt.

### Countdown

Der Countdown ist ein Objekt, das erstellt wird, wenn der Server das Spiel per Leertaste startet. Er beinhaltet einen Timer, der per «update»-Methode von 3 auf 0 zählt und eine Methode, um den Countdown auf dem Bildschirm anzuzeigen.

### Direction

Repräsentiert einen zweidimensionalen Vektor ([Vektor (Wikipedia)](https://de.wikipedia.org/wiki/Vektor)), der benutzt werden kann, um einen Punkt in eine bestimmte Richtung zu verschieben. Eine Direction kann von einer x-Richtung und y-Richtung gesetzt werden, aber auch von einem Winkel und einer Länge. Die zweite Methode wird verwendet, um das Auto in die Richtung zu bewegen, in die es gedreht ist.

### DrawingUtils

DrawingUtils beinhaltet eine einzige Methode («drawRotated»), deren Zweck es einzig und allein ist, ein Bild gedreht auf einen GraphicsContext zu zeichnen.

### EndScreen

Der EndScreen ist eine Klasse, deren einzige Methode die folgendes auf den Bildschirm zeichnet: die Rangliste, den Text «Ziel erreicht» und eine Information, dass man mit der Leertaste ins Hauptmenü zurückkehren kann. Diese Methode wird aufgerufen, sobald das Auto das Ziel durchquert hat.

### GameLoop

Der GameLoop ist eine abstrakte Klasse, die eine «update»-Methode bereithält. Die Methode wird nach dem Aufrufen der «start»-Methode durchgehend aufgerufen. Der «update»-Methode wird dabei immer die Zeit, die seit dem letzten Frame vergangen ist, in Sekunden mitgegeben.

### GameNetworking

GameNetworking kümmert sich auf dem Client um alles, was mit Netzwerk-Dingen zu tun hat. Die Klasse erstellt den Client, beinhaltet Methoden für das Senden von Nachrichten und verarbeitet Nachrichten, die ankommen.

### GameObject

GameObject ist die Base-Klasse für alle GameObjekt im Spiel, ausser der Strasse (siehe Road). Ein GameObjekt beinhaltet nur einen Point für die Position und zwei abstrakte Methoden für das Zeichen und Updaten.

### GamePainter

Die Aufgabe des GamePainters ist es, alles zu zeichnen, was es zu zeichnen gibt. Dazu muss lediglich die «draw»-Methode mit dem GraphicsContext aufgerufen werden.

### GameTimer

Der GameTimer misst die Zeit zwischen Start und Ende eines Spiels. Er wird gestartet, wenn der Countdown vorüber ist und gestoppt, wenn der Spieler die Ziellinie überquert.

### JamKartApp

JamKartApp ist die Startklasse der Applikation. Hier wird der Navigator erstellt, die Musik gestartet und ClientGame und ServerGame zwischengespeichert, damit man von überall auf sie zugreifen kann.

### KeyEventHandler

Der KeyEventHandler ist ein JavaFX-EventHandler für Tastatur-Events. Dabei werden die fünf Booleans «isForwardPressed», «isBackwardPressed», «isLeftPressed», «isRightPressed» und «isSpacePressed» je nachdem, ob die jeweiligen Tasten gedrückt werden, gesetzt. Diese Variablen werden dann beispielsweise bei der Steuerung des ClientMyCars verwendet.

### Launcher

Der Launcher tut genau das, was sein Name sagt: Er beinhaltet die Main-Methode, die die JamKartApp über die JavaFX-Methode «launch» startet. Dies ist notwendig, um die Applikation als JAR-Datei zu exportieren.

### Lobby

Die Lobby beinhaltet wie EndScreen nur eine Methode, diese zeichnet jedoch alle anderen Mitspieler mit ihren Namen und einen Hinweis für den Ersteller dies Spiels, dass dieses mit der Leertaste gestartet werden kann.

### Map

Die Map beinhaltet ganz einfach eine Liste mit allen GameObjects und die Road. Dazu kommt allerdings noch eine Methode, die eine Liste aller GameObjekte zurückgibt, die von einer bestimmten Klasse erben.

### MapLoader

Der MapLoader lädt alle GameObjekte (ausser dem ClientMyCar) und die Road. Dabei werden der Road alle Punkte und RoadPathMarkers hinzugefügt.

### MathUtils

MathUtils beinhaltet Hilfsmethoden, die mit Mathematik zu tun haben, wie beispielsweise Sinus, Cosinus, eine Methode, die die Breite eines Textes zurückgibt oder eine, die die Wurzel einer Zahl zieht.

### Message

Repräsentiert eine Nachricht, die über das Netzwerk gesendet/empfangen werden kann. Im Hintergrund steht dabei eine HashMap, in die von aussen Daten eingespeist werden können. Eine Mesage, die über das Netzwerk versendet wird, muss einen Typ haben, der im Konstruktor mitgegeben werden kann. Beim Senden wird die HashMap zu einem String umgewandelt und beim Empfangen dann wieder zurück zu einer HashMap.

### MessageHandler

Ein Interface, das nur eine «handle»-Methode beinhaltet, der eine Message übergeben wird. Der MessageHandler muss bei einer Connection gesetzt werden, wenn man Nachrichten behandeln möchte, sodass dieser die «handle»-Methode dann aufrufen kann.

### MessageType

Das Enum MessageTyp ist der Typ einer Message, der dem Empfänger der Nachricht signalisiert, um was für eine Art Nachricht es sich handelt und wie sie verarbeitet werden sollte.

### Navigator

Der Navigator wechselt auf der Stage die verschiedenen Scenes aus. Die Scenes müssen allerdings zuerst mit einem SceneType registriert werden, damit man «navigateTo» mit dem neuen SceneTyp aufrufen und somit zu der angegebenen Szene wechseln kann.

### NetErrorMessages

NetErrorMessages beinhaltet String-Konstanten, die Nachrichten beinhalten, die dem Benutzer angezeigt werden, wenn ein Netzwerk-Fehler auftritt und dieser zur SceneBackToStart umgeleitet wird.

### OilPuddle

Repräsentiert ein Ölpfützen-Gameobjekt, bei dem der Spieler die Kontrolle über das Auto verliert, sollte dieses die Ölpfützen berühren.

### Point

Ein Point stellt einen Punkt im zweidimensionalen Raum dar. Ein Punkt wird dabei durch eine x-Koordinate und eine y-Koordinate definiert.

### Road

Die Road ist die Strasse, auf der die Autos fahren. Die Autos verlieren dabei an Geschwindigkeit, sollten sie die Strasse verlassen. Die Strasse beinhaltet eine «draw»-Methode, mit der sie auf den Bildschirm gezeichnet werden kann und eine «isInside»-Methode, mit der geprüft wird, ob sich ein Punkt auf der Strasse befindet. Die Road beinhaltet aber auch eine Liste mit RoadPathMarkern.

### RoadPathMarker

Ein RoadPathMarker ist ein Kreis auf der Strecke, der benutzt wird, um herauszufinden, ob der Spieler eine Runde legal gefahren hat oder welchen Rang er momentan einnimmt (siehe RoadPathTracker).

### RoadPathTracker

Der RoadPathTracker hat die Aufgabe, das Fahren des Autos entlang der Strecke zu überwachen. Er zählt dabei die Anzahl an RoadPathMarkers, die der Spieler berührt hat. Wenn alle RoadPathMarkers berührt wurden, wurde eine Runde beendet und der Rundenzähler wird erhöht. Weiterhin wird die Anzahl RoadPathMarkers mithilfe der Methode «setProgress» auf dem Spieler gesetzt, sodass dieser abschätzen kann, wie weit vorne er liegt und welchen Rang er einnimmt.

### SceneBackToStart

Die SceneBackToStart wird angezeigt, wenn es einen Netzwerkfehler gegeben hat und das Spiel unterbrochen werden muss. Dabei wird eine Fehlermeldung dick angezeigt und einen Knopf, der zum Hauptmenü zurückführt.

### SceneCreateGame

In der SceneCreateGame kann ein Spiel erstellt oder zum Hauptmenü zurückgekehrt werden. Beim Erstellen muss ein Name und ein Port angegeben werden. Beim Klick auf «Spiel erstellen» wird dann ein ServerGame und ein ClientGame erstellt.

### SceneGame

SceneGame ist die Szene, auf die das gesamte ClientGame seinen Inhalt zeichnet. Ausserdem wird hier der KeyEventHandler erstellt, der vom ClientGame verwendet wird.

### SceneJoinGame

In der SceneJoinGame kann einem Spiel beigetreten oder zum Hauptmenü zurückgekehrt werden. Beim Beitreten muss ein Name, eine Ziel-IP-Adresse und ein Zielport angegeben werden. Beim Klick auf «Spiel beitreten» wird dann ein ClientGame erstellt.

### SceneStart

Diee SceneStart ist die Szene, die angezeigt wird, wenn das Spiel startet. Hier gibt es zwei Knöpfe («Spiel erstellen» und «Spiel beitreten»), die auf die SceneCreateGame und die SceneJoinGame verweisen.

### SceneType

Das Enum SceneType ist im Navigator der Schlüssel, der verwendet wird, um auf eine bestimmte Scene zu navigieren. Dabei hat jede Scene ihren eigenen Schlüssel, damit sie identifiziert werden kann.

### Server

Der Server ist eine Klasse, die Verbindungen von Clients akzeptiert und Messages verarbeitet. Dabei wird für jeden Client, der verbindet, ein Connection-Objekt erstellt. Für jede Connection wird dann ein MessageHandler gesetzt, der die Nachricht einfach an den ServerMessageHandler weiterleitet.

### ServerCar

Ein ServerCar repräsentiert das Auto eines Clients. Dieser hat diejenigen Eigenschaften, die der Client an den Server schickt, wie Position, Drehung, Fortschritt auf der Strecke etc.

### ServerGame

Das ServerGame ist der Startpunkt eines Spiels auf dem Server. Hier wird der Server gestartet, neue Spieler empfangen, Nachrichten gesendet und Nachrichten empfangen.

### ServerMessageHandler

Der ServerMessageHandler ist wie der MessageHandler ein Interface, dass nur eine «handle»-Methode beinhaltet. Allerdings wird hier nicht nur die Message mitgegeben, sondern auch die Connection, von der die Nachricht kommt.

### Tree

Ein Tree ist ein dekoratives GameObject und wird überall auf der Map verteilt.

### Validator

Der Validator beinhaltet Methoden für das Überprüfen der Gültigkeit von Benutzereingaben. Dazu gehören Namen, Ports und IP-Adressen.