Inhoud projectplan

Context – Een beschrijving van de fysieke, bedrijfsmatige en/of maatschappelijke omgeving waarin het projectresultaat zal worden gebruikt. De context heeft invloed op zowel eisen aan het projectresultaat als ontwerpkeuzen. Bij dit project maakt het bijvoorbeeld veel verschil of het bestuurde voertuig in de binnenstad van Rotterdam of op een afgesloten race-circuit moet rijden.

**Onze simulatie en ons circuit zijn uitsluitend visueel gesimuleerd. In de simulatie bevindt zich een track waar de auto op moet rijden. Ook buiten onze track rijdt de auto door, in een oneindige (gesimuleerde) omgeving. Ons projectresultaat zetten wij op papier, slaan wij op in onze MakeAIWork directory en maken dit inzichtelijk voor docenten middels Git. Tevens zullen wij op 7 oktober ons resultaat presenteren aan docenten en toekomstig werkgevers.**

Bewustzijn van de context zorgt ervoor dat geen requirements worden vergeten en dat het ontwerp een mate van degelijkheid vertoont die past bij de situatie. Een besturing van een modeltrein en een echte trein kunnen qua functionele eisen veel op elkaar lijken, maar qua betrouwbaarheid zijn er verschillen die tot uitdrukking komen in het ontwerp, bijvoorbeeld wat betreft “layered safety”.

Requirements – Eisen aan het eindresultaat, puntsgewijs, nauwkeurig (liefst kwantitatief) geformuleerd en met MoSCow aanduidingen.

1. **Het projectplan**
2. **Broncode**
3. **Testrapport – met daarin**

* **Welke scanningsmethode is het meest geschikt en waarom?**
* **Welke aansturingsmethode werkt het beste en waarom?**
* **Hoe is veiligheid gegarandeerd in alle gevallen?**
* **Welke aansturingsmethode is het veiligst en waarom?**
* **Is de opzet van de codebase modulair? Hoe aan te tonen?**

1. **Projectevaluatie, tweeledig:**

* **Meta-functionele overwegingen**
* **Individuele ervaringen**

Testspecs – Operationele beschrijving van de (eveneens bij voorkeur kwantitatieve) tests die nodig zijn om vast te stellen of aan de requirements wordt voldaan.

**Beide scanningmethoden (LIDAR/SONAR) zullen worden getest in beide aansturingmethoden (hardcoded/neural network). In de tests zullen we data verzamelen**

Ontwerp – Een beschrijving van de te volgen oplossingsstrategieën met voldoende details om programmering door een weldenkende programmeur (in tegenstelling tot “coder”) mogelijk te maken, maar niet meer dan dat. Diagrammen, lijstjes en berekeningen kunnen deel uitmaken van een ontwerp. Het ontwerp zelf is echter de *boodschap*, *niet de vorm* waarin deze is gegoten.

Een goede manier van ontwerpen is “Literate Programming” (zie Google), waarbij documentatie zoveel mogelijk voorafgaand aan programmering in de broncode-bestanden wordt geplaatst. Dit heeft als voordeel dat documentatie altijd bij de hand is en dus wordt bijgewerkt.

**Het ontwerp in python waarbij voor iedere regel broncode staat vermeld wat er moet gebeuren. Diagrammen, lijstjes en berekeningen mogen erbij als dit verduidelijkt.**

Het is vaak handig je programma al bij het ontwerpen in compartimenten te verdelen die redelijk los van elkaar staan (niet te veel onderlinge informatie-uitwisseling hebben) en dus ook los te testen zijn. Zulke compartimenten heten modules. Als je modules gebruikt, specificeer ze dan, inclusief de informatie-uitwisseling, in het ontwerp. Dit eerste project is beperkt van omvang en het gebruik van modules is niet per se noodzakelijk. Pas ze alleen toe als dit in jouw aanpak voordelen heeft.

Het automatisch genereren van diagrammen uit rudimentaire stukken broncode zoals class declarations en function definitions is te prefereren boven het met behulp van een grafisch een tool samenstellen van prachtige diagrammen die vervolgens bij de eerste codewijziging verouderd zijn en niet, of ten koste van veel uren, worden bijgewerkt.

Planning – Een initiële, onderweg bij te stellen beschrijving van volgorde en doorlooptijd en benodigde mensuren voor de activiteiten die onderdeel zijn van het project. Als je modulair (met modules) werkt, laat deze dan terugkeren in je planning (en in wat grotere “real life” projecten ook in de taakverdeling).

Het projectplan, inclusief de planning, staat onder versiebeheer.