

Race Simulator

Software ontwikkeling doorloopt bijna altijd de volgende stappen: **Aanleiding** <-> **Analyse** <-> **Ontwerp** <-> **Realisatie** <-> **Testing** <-> **Implementatie**. Zoals je kan zien wijzen alle pijlen 2 kanten op. Je zult als ontwikkelaar vaak een stap terug moeten doen vanwege bijvoorbeeld nieuwe wensen of inzichten.

De module C# richt zich voornamelijk op de stappen **realisatie** en **testing**. De andere stappen zijn al voor je uitgevoerd en dienen als houvast voor jouw uitwerking van een race simulator. De informatie uit onderstaande **aanleiding** en **analyse** zal in elke level terug komen.

Aanleiding

Om een competitie te kunnen winnen wordt er steeds vaker gebruik gemaakt van ICT middelen. Een simulator kan bijvoorbeeld voorspellingen doen over een competitie. Met deze voorspellingen kunnen teams de middelen beter en efficiënter inzetten.

Nog niet alle competities hebben een simulator. Dit is duidelijk een gat in de markt. Het doel van dit project is om een simulator voor een competitie te ontwikkelen. De simulator wordt internationaal aangeboden waarbij de koper naast de applicatie ook de sourcecode krijgt.

Domeinanalyse

Begrippen

Doordat de applicatie internationaal aangeboden wordt zijn de gebruikte termen en begrippen in het Engels.

Term	Vertaling	Uitleg
Competition	Competitie	Competitie
Participant	Deelnemers	Deelnemer in de competitie
Equipment	Racemiddel	Middel gebruikt door de deelnemer om te racen, bv schoenen, auto, fiets...
Track	Racebaan	Een racebaan waar de deelnemers tegen elkaar strijden

Term	Vertaling	Uitleg
Section	Sectie	Onderdeel van een racebaan bestaande uit twee stroken

Teams

Het kost veel geld om een raceteam te onderhouden. Daarom worden raceteams ook gebruikt als een vorm van reclame. Elke raceteam heeft daarom een eigen teamkleur passend bij de kleur van de sponsor.

Om andere teams te kunnen verslaan worden verschillende ICT middelen gebruikt. Teams kunnen zich geen fouten of vergissingen permitteren daarom wordt er sterk gestuurd op kwaliteit en betrouwbaarheid.

Teams zijn niet erg open om zo andere teams niet op ideeën te brengen. Teams kopen daarom ook de sourcecode van een applicatie om volledig grip en inzicht te krijgen. Geheimen mogen niet uitlekken.

De competitie

Een competitie bestaat minimaal uit 3 deelnemers en 2 racebanen. Elke deelnemer heeft 1 racemiddel om mee te racen, denk bijvoorbeeld aan schoenen, auto of een fiets.

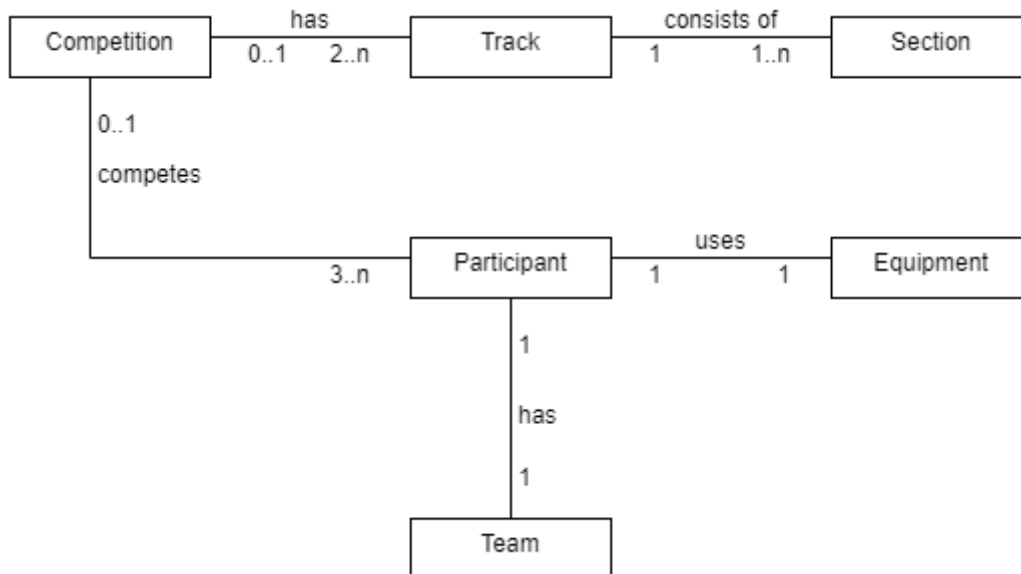
De racebanen hebben een vaste volgorde in de competitie. Alle deelnemers moeten minimaal 2 rondjes racen op een racebaan. Tijdens de race kan het voorkomen dat het gebruikte racemiddel tijdelijk defect is.

Elke racebaan is opgedeeld in meerdere secties waarbij alle secties dezelfde lengte hebben. Tijdens een race kunnen maximaal 2 deelnemers aanwezig zijn op 1 sectie.

Nadat een race is afgerond krijgt elke deelnemer punten. De deelnemers die als eerste is gefinished krijgt de meeste punten. De competitie is afgerond wanneer de deelnemers op alle racebanen hebben geraced en hiervoor punten hebben gekregen.

Domeinmodel

De bovenstaande analyse is samengevat in het onderstaande domeinmodel.



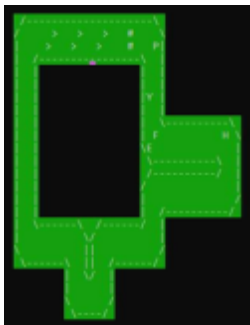
Inrichten solution

Taakgroep: Project Basics

Context

In deze periode ontwikkel je een game geschreven in C# (spreek uit siesjarp). Het doel is dat je C# en een aantal concepten leert zoals het werken met threads.

Eerst zet je het project op waarbij elk project een eigen doel heeft dat tijdens de ontwikkeling ook niet meer verandert. Door deze strategie kun je de game eerst bouwen als console applicatie en voeg je later ook een grafische weergave toe.



Tekst versie



Grafische versie

In de instructie wordt steeds het voorbeeld gegeven van een racebaan, maar je krijgt zelf de vrije hand in de vorm: de game zou even goed een zwemwedstrijd, een paardenrace of hardloop wedstrijd kunnen zijn. De minimale eisen voor een race zijn:

- De racetrack heeft een start en een finish waarbij de deelnemers meerdere rondjes moeten doen.
- Maximaal 2 deelnemers naast elkaar tijdens de race.
- De competitie krijgt minimaal 2 verschillende racetracks.

Taak: Solution

Met deze opdracht maak je een eerste begin met het opzetten van het project in de ontwikkelomgeving. Als ontwikkelomgeving maak je gebruik van Visual Studio. De uitleg in deze en volgende opdrachten is steeds gebaseerd op Visual Studio.

In C# bestaat een applicatie altijd uit een 'solution', een oplossing. Een solution bevat één of meerdere projecten. Elk project in een solution heeft zijn eigen verantwoordelijk.

In deze opdracht maak je de solution aan met een console applicatie project.

Aanpak

Met deze opdracht maak je solution voor je game zodat je daarin later projecten kunt toevoegen.

- Start Visual Studio op.
- Kies: 'Create new project'.
- Kies als project template: Console App (.NET Core)
- Vul de velden met passend waarden. Project Name en Solution Name mogen verschillen. Ook de standaard locatie mag je wijzigen naar eigen voorkeur.
- Rechts in de Solution Explorer zie je de solution met daarin je console project. In dit project is een klasse `Program.cs` aanwezig. Start het programma met 'F5'.

Ondersteunende informatie

Microsoft heeft een [korte introductie](#) op C# projecten, waarin ook testen worden besproken.

Taak: Projecten opzetten

Met deze opdracht zet je de noodzakelijke projecten op. Elk project heeft zijn eigen verantwoordelijkheid. Bij de scheiding van verantwoordelijkheden hanteer je in dit project het **MVC pattern**.

Het 'console project' is de 'view' in het MVC patroon. Er moeten nog twee projecten aangemaakt worden in de solution.

Eerst voeg je de twee projecten toe en vervolgens geef je het 'console project' een referentie naar de toegevoegde projecten om van de toegevoegde projecten gebruik te kunnen maken.

Aanpak

Standaard heeft Visual Studio rechts een 'Solution Explorer'. Dit is een weergave van alle projecten in de solution en alle onderliggende bestanden binnen de projecten.

In de solution voeg je voor de controller en het model een project toe. **Deze projecten zijn van het type Class Library**. Vervolgens geef je de view project, het console project, referenties naar de andere projecten.

- Klik op **Solution** in de **Solution explorer**
- Ga via het contextmenu (klik met rechter muisknop) naar **Add > New Project...**
- Zoek op **Class Library**
- Selecteer **Class Library (.NET Core)**. Doorloop de stappen waarbij je het project de naam 'Model' geeft.
- Repeteer voorgaande stappen, met dit verschil: geef als naam 'Controller'.
- Geef het *console project* **referenties** naar de toegevoegde projecten. Klik op het console project.
- Ga naar de Reference Manager: **context menu > Add > Reference**. Selecteer de toegevoegde projecten en klik op 'OK'.

Ondersteunende informatie

Meer informatie over het **MVC pattern** kan je [hier](#) vinden.

Naast een referentie naar een ander project is het mogelijk om ook [andere bronnen te refereren](#).

Taak: Test project opzetten

Omdat een solution kan bestaan uit meerdere projecten is het gebruikelijk om ook meerdere test projecten op te nemen in een solution.

In deze opdracht maak je een test project om de het controller project te kunnen testen.

Aanpak

Eerst voeg je een test project toe en dit project geef je een referentie naar het controller project.

- Maak een nieuwe branch
- Klik op **Solution** in de **Solution explorer**
- Ga via het contextmenu (klik met rechter muisknop) naar **Add > New Project...**
- Zoek op **NUnit Test**
- Selecteer **NUnit Test Project (.NET Core)**. Doorloop de stappen waarbij je het project de naam 'ControllerTest' geeft.
- Geef het project een referentie naar het controller project.

Ondersteunende informatie

Deze opdracht is alleen gebaseerd op kennis uit de voorgaande opdrachten.

Taak: GIT repository

Met deze opdracht zet je een GIT repository op om je source code te bewaren.

Je maakt een repository en pusht je code naar de repository.

Aanpak

Je maakt een repository bij een bekende vendor en pusht daar je code naar toe.

- Kies een vendor waar je je repository wilt plaatsen: Github.com, Bitbucket.com, gitlab.windesheim.nl, Gitlab.com.
- Log in bij de vendor, of maak eerst een account.
- Het is verstandig om SSH keys te gebruiken. Dit kan niet bij gitlab.windesheim.nl.
- Push je code naar de repository.
- controleer of je onderstaande commands kent, en anders moet je ze uitzoeken:

```
git init
git add .
git commit -m '[string]'
git push
git checkout -b
git checkout [branch]
git status
git pull
git merge
```

Ondersteunende informatie

Een heel toegankelijk boek is online verkrijgbaar over [git](#).