

Challenge – RoboWorld

| | | | |
|---------------|--------------------------------|-------------------|--------------|
| Titel | Challenge – RoboWorld | | |
| Vak | Programmeren in C# - Gevorderd | | |
| OPO Code | MGP02a | Auteur(s) | Johan Roosen |
| Versie | 1.0 | Review | |
| Aangemaakt op | 22/11/2021 | Laatste wijziging | 22/11/2021 |

1 Challenge

Een “challenge” (of uitdaging) is een oefening die vrijblijvend gemaakt kan worden door de studenten. Het doel is om een extra uitdaging te voorzien voor studenten, die door hun ervaring (eerdere opleidingen of diploma’s) de oefeningen en opdrachten uit de opleiding te eenvoudig vinden.

Naast de aangeleerde kennis in de opleiding, zal de uitdaging zich ook richten op:

- Meer analytisch vermogen van de student om een probleem in een oplossend algoritme te implementeren.
- Meer autonomie van de student om extra onderzoekwerk (over C# en object-oriented programming) te verrichten om tot een oplossing te komen.
- De competentie om op basis van een formeel document een opdracht te begrijpen en zelfstandig uit te voeren als developer.

De challenge maakt geen deel uit van de evaluatie en wordt bijgevolg ook niet gescoord door de docent.

2 RoboWorld

In deze challenge gaan we stap voor stap een toepassing bouwen waarbij een robot zich in een virtuele wereld (=map) bevindt, waar de robot de opdracht heeft alle beschikbare diamanten te verzamelen zonder zich te pletter te lopen of van de map te vallen.

De virtuele wereld is een vierkant bestaande uit 10x10 vakjes. Op sommige vakjes ligt een diamant (o). Er bevinden zich echter ook een willekeurig aantal rotsen (#) in de virtuele wereld. Indien de robot tegen een rots loopt of tegen de grens van de virtuele wereld, eindigt de opdracht en heeft de robot gefaald.

```
+---+---+---+---+---+---+---+---+---+---+
|   |   | # |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+---+---+
|   |   |   | # |   |   |   | # |   | o |   |   |
+---+---+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   |   |   | # |   |   |
+---+---+---+---+---+---+---+---+---+---+
|   |   |   |   |   | > |   |   | S |   | # |   |   |
+---+---+---+---+---+---+---+---+---+---+
|   |   | o | # |   |   |   | # | # |   |   |   |
+---+---+---+---+---+---+---+---+---+---+
|   |   |   | # |   |   |   |   | o |   |   |   |
+---+---+---+---+---+---+---+---+---+---+
|   |   |   | # | # |   |   | # | # |   |   |   |
+---+---+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   | # |   |   |   | # |   |
+---+---+---+---+---+---+---+---+---+---+
|   | S |   |   |   | o | # |   |   | # | # |   |
+---+---+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+---+---+
```

3 Versie 1 – Manual

3.1 Beschrijving

De robot wordt in eerste instantie bestuurd door de gebruiker die de volgende instructies kan sturen (via menu opties) naar de robot:

- Move – de robot verplaatst zich 1 vakje naar voor in de kijkrichting van de robot. De kijkrichting kan zijn:
 - Noord: ^
 - Oost: >
 - Zuid: v
 - West: <
- TurnLeft - de robot blijft ter plaatse, maar draait zich 90° in tegengestelde wijzerzin zodat de kijkrichting verandert.
- TurnRight - de robot blijft ter plaatse, maar draait zich 90° in wijzerzin zodat de kijkrichting verandert.
- PickupDiamond – raapt de diamant op.
- IsDiamondAvailable – test of op het vakje waar de robot staat een diamant ligt (ja/nee).
- IsInFrontOfRock – test of de robot voor een rots staat.
- IsInFrontOfFrontier – test of de robot voor de rand van de virtuele wereld staat.

Bij de start van het programma wordt de map van de virtuele wereld ingeladen (inclusief posities van alle rotsen en diamanten). Vervolgens wordt de robot op het eerste vakje van de bovenste rij geplaatst met als kijkrichting “Oost”. Daar kan dus nooit een rots staan.

Vanaf dat moment laat een menu de gebruiker de nodige instructies uitvoeren. De map kan ook een geheime doorgang hebben, waarbij de robot op een andere plaats op de map springt. Bij het oppakken van de laatste diamant, eindigt de opdracht. De gebruiker kan zijn opdracht ook vroegtijdig stoppen via een menu optie “Stop”.

De term “Manual” duidt op het feit dat de robot manueel bestuurd wordt door de gebruiker die zicht heeft op de map en dus precies weet waar de obstakels en robot zich bevinden.

3.2 Functionele vereisten

| ID | Fase | Onderdeel | Functionele vereiste |
|---------|------|-----------|--|
| FR-01 | 1 | Opstarten | De toepassing wordt opgestart als Console applicatie, via de “Run” vanuit het Visual Studio project. |
| FR-02.1 | 1 | Opstarten | De vaste map van de virtuele wereld wordt ingelezen uit een string van 100 characters (=10 rijen van 10 vakjes), waarbij een <ul style="list-style-type: none">• “E” staat voor leeg vakje (Empty)• “R” staat voor een rots (Rock)• “D” staat voor diamant (Diamond) |
| FR-04 | 1 | Spelen | De gebruiker heeft een menu om een wereld te initialiseren, de robot instructies uit te voeren of het spel vroegtijdig te beëindigen. <ol style="list-style-type: none">1. Initialiseer map2. Stap3. Draai Links4. Draai Rechts5. Raap diamant op6. Stop |
| FR-05 | 1 | Spelen | Als de robot tegen een rots loopt of uit de wereld valt, krijgt de gebruiker een specifieke foutboodschap en verandert de kleur van de wereld naar rood. |
| FR-06 | 1 | Spelen | De gebruiker heeft steeds zicht op twee tellers: <ul style="list-style-type: none">• Aantal diamanten opgeraapt• Aantal diamanten nog te verzamelen |
| FR-08 | 1 | Wereld | Na elke instructie toont de applicatie de volledige toestand van de wereldmap. |
| FR-09 | 1 | Robot | Als de robot op een diamant staat, wordt een symbool getoond dat aangeeft dat er zowel een diamant als een robot op het vakje staat. |
| FR-03 | 2 | Wereld | De gebruiker kan een wereldmap opmaken en opslaan in een file voor later gebruik. |
| FR-02.2 | 2 | Opstarten | De gebruiker kan een wereldmap inladen uit een file. De keuze wordt gemaakt op basis van een lijst van beschikbare files die in het menu getoond wordt. |
| FR-07 | 2 | Wereld | Een vierde type item kan op de wereld geplaatst worden. <ul style="list-style-type: none">• “S” staat voor geheime doorgang (Secret passage) Zo kunnen er slechts 2 items zijn. Als de robot op een “S” staat, en de Move() instructie wordt uitgevoerd neemt hij de geheime gang en komt er aan de andere kant uit in dezelfde kijkrichting. De robot kan dus niet over een “S” heen stappen. |

4 Versie 2 – AI

Deze vergevorderde versie gaat nog een stap verder. In dit geval is het niet meer de gebruiker die één voor één de instructies geeft op basis van de virtuele wereld die hij/zij ziet, maar ga je een algoritme programmeren (dat gebruik maakt van de mogelijke instructies aan de robot), zodat alle diamanten worden opgeraapt op een **willekeurige** wereld. Bij het inladen van een map, zal de computer volledig autonoom alle diamanten vinden en oprapen.

Verschil is dat de menu opties voor “Stap”, “Draai Links”, “Draai Rechts” niet meer gebruikt worden. Het computerprogramma gaat alles automatisch doen. En worden vervangen door de menu optie “Uitvoeren”.

Bij elke instructie naar de robot moet een nieuwe wereld getekend worden met de actuele situatie. Om de toeschouwer te laten meevolgen, is het nodig om een vertraging in te bouwen van 1 à 2 seconden per stap die de robot zet.

- De eerste versie van het algoritme moet werken voor elke virtuele wereld, maar zonder rotsen en geheime gang.
- Het tweede algoritme moet werken voor elke virtuele wereld met een willekeurig aantal rotsen, maar zonder geheime gang.
- Het derde algoritme moet werken voor elke virtuele wereld met een willekeurig aantal rotsen, en één geheime gang.

5 Opgave en tips

- Probeer de juiste klassen (en interfaces) te definiëren.
- Bouw de Console applicatie die de functionele vereisten implementeert.
- Maak een class diagram om jezelf te helpen.
- Houdt GUI (graphical user interface) en logica gescheiden. Zodat je klassen en logica nadien ook gebruikt kunnen worden in een Window of Web applicatie.