

Egdom, Ronald van

Studentnummer: 567906

Instelling: Hoge School Arnhem & Nijmegen; Informatie Communicatie Academie.  
Klas: I1E.  
Vak: SPD (Structured Program Development)  
Docent: Wim Zoet  
Datum: 13-10-2015, Plaats: Arnhem  
Versie: 0.1

Casus hardloopwedstrijd v2

Analyse

# Inleiding

Bij deze Casus moet een programma geschreven worden dat een hardloopwedstrijd tussen robots simuleert. De robots worden hierbij weergeven als een driehoek.

Bij start gaan de robots dobbelen met een aantal dobbelstenen om het aantal pixels dat de robots mogen voortbrengen te bepalen. Wanneer alle dobbelstenen evenveel ogen hebben krijgt de robot een extra beurt. Dus als de robot 2x 5 gooit mag de robot nog een keer gooien, als de robot dan 2 en 6 gooit mag de robot totaal 18 pixel vooruit. Als alle robots gefinished zijn komt er een scoreboard die het aantal beurten van de robots laat zien.

# Antwoorden vragen Casus

Vraag 1: “Het programma bestaat uit 3 grote stappen. Welke zijn dat?”

Antwoord 1: “De belangrijkste stappen binnen het programma zijn “Het selecteren van de robots en opties”, “Het laten racen van de robots” en “Het tonen van de scores”

Vraag 2: “Wat is de minimumwaarde van de slider “Aantal Robots”?”

Antwoord 2: “De minimumwaarde is 1, want je wilt minstens 1 robot laten lopen”

Vraag 3: “Wat is de maximumwaarde van de slider “Aantal Robots”?”

Antwoord 3: “Het maximaal aantal robots wordt door de volgende formule bepaald: “((Schermbreedte-50)-25(marge))/((25(minimumrobotbreedte+25(marge))” Dat zal dus bij een schermbreedte van 800, een minimumrobotbreedte van 25 en een marge van 25 totaal 14 robots zijn.

Vraag 4: “Zelfde vraag voor de slider marge”

Antwoord 4: “

Vraag 5:

Antwoord 5: “

Vraag 6:

Antwoord 6: “

Vraag 7:

Antwoord 7: “

Vraag 8:

Antwoord 8: “

Vraag 9:

Antwoord 9: “

# Functionele eisen

Het aantal robots is configureerbaar.

De breedte en dus de hoogte van de robot wordt berekend.

De afstand tussen de robots en de rand van het speelveld en de robots is configureerbaar.

De af te leggen afstand is configureerbaar.

Het aantal te gebruiken dobbelstenen is configureerbaar.

De wedstrijd moet in ieder geval in een rechte lijn kunnen plaatsvinden.

Zowel de startstreep en de finishlijn zijn zichtbaar.

De uitslag mag niet door de startstreep en finishlijn heen geprojecteerd worden.

**Extra:**

Een 400 meter uitslag waarbij de robots een rondje moeten lopen

Snelste tijd per afstand bewaren en aanpassen.

# Events, Methoden en Variabelen

Hieronder is een weergave van de events waarop het programma zal reageren en de methoden en variabelen die het gebruikt.

|  |  |  |
| --- | --- | --- |
| Module | Event/Methode/Variabele | Toelichting |
| Instellingen | int wMargeBreedte | Horizontale marge van het window binnen het scherm |
|  | Int wMargeHoogte | Verticale marge van het window binnen het scherm |
| Tekenen | int marge | De marge die wordt gebruikt om de afstand tussen de robots te berekenen |
|  | int minimumRobotBreedte | De minimale breedte van een robot |
| Scores | int[] hoogstescores | het bijhouden van de 10 hoogte scores. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Events | void Setup() | In de set-up wordt de wedstrijd geinitialiseerd en aangeroepen. Hierbinnen worden de andere functies aangeroepen. |
|  | void MouseClicked() | Wanneer de speler ergens metde muis op klikt moet er bepaald worden waar het scherm zich eigen vervolgens naar toe brengt. Als er geklikt wordt op het hoofdmenu icoon moet er naar het hoofdmenu verplaatst worden. |
| Tekenen/Rekenen | void BeweegRobots() | Deze methode moet de positie van de robots bepalen. |
|  | void RolDobbelsteen() | Met deze methode kunnen de robots een positieverandering aanvragen doormiddel van een rol van een dobbelsteen |
|  | void TekenRobots | Met deze methode worden de robots getekend |
|  |  |  |