|  |  |
| --- | --- |
| # | Ticket beschrijving |
| 16 | DeveloperLand is aan het uitbreiden en we zijn dan ook trots dat we binnenkort onze nieuwste attractie, Debug Hero, kunnen openen. De attractie zelf is bijna af, maar daaromheen moet nog een hoop gebeuren. Een van die dingen is het inregelen van de wachrij-indicatoren: Borden in het park die aangeven hoe lang de wachtrij voor een attractie ongeveer is. Gelukkig is Debug Hero niet onze eerste attractie, dus we hoeven niet *from scratch* te beginnen, we kunnen voortborduren op een app die al in gebruik is voor een andere attractie, Bugs of Horror!  **Introductie nieuwe applicatie**  **1.** Voordat we gaan beginnen aan het verbouwen van de applicatie is het belangrijk dat je eerst goed weet wat de huidige applicatie doet. Open het project in Visual Studio, start de applicatie en kijk wat de applicatie doet (hint: Druk op button ‘Update informatie’).  2. De applicatie gebruikt bestanden met een .XML-indeling die automatisch worden gegenereerd door sensoren langs de wachtrij van de attractie. Open het bestand Assets/SensorData/WachtrijSensoren.xml. Je ziet hier bijvoorbeeld:    <Sensor06>False</Sensor06>  3. Wat denk je dat dit betekent? Wat gebeurt er wanneer je de waarde van Sensor06 veranderd naar True? (hint: Druk nog eens op ‘Update informatie’). Probeer eens te bedenken wat de sensorwaarden betekenen en hoe de sensorwaarden invloed hebben op de output van het programma. Wat denk je bijvoorbeeld dat er gebeurt wanneer ik Sensor01 en Sensor10 op True zet, en alle andere op False? Kun je andere waarden dan True of False gebruiken? Wat gebeurt er wanneer je rijen met sensorgegevens verwijdert of toevoegt? Probeer dit uit en kijk wat er gebeurt, zodat je goed begrijpt wat de sensorwaarden precies betekenen.  4. Open nu het bestand ‘AttractieStatus.xml’ in dezelfde map en probeer op dezelfde manier te bepalen wat de gegevens in dit bestand betekenen en beschrijf dan ook de werking van dit bestand.  **5.** Open het project in Visual Studio en open de designer (klik met je rechtermuisknop op ‘MainPage.xaml’ en kies voor ‘View Designer’). Hier zie je de grafische elementen van het bestaande programma. Bekijk nu de code die wordt uitgevoerd, wanneer je op de ‘Update informatie’ knop klikt (dubbelklik op de buttom).  **6.** Visual Studio biedt een hoop tools om je werk makkelijker te maken. Eén daarvan is dat je snel naar andere methoden kunt springen, vanaf de plek waar ze worden aangeroepen. Hierdoor kun je als het ware ‘spoorzoekertje spelen’ door je code. In KnopUpdateInformatie\_Click worden twee methoden aangeroepen (VerwerkWachtrijSensorData en VerwerkAttractieStatusData). Klik met je rechtermuisknop op VerwerkAttractieStatusData, en kies vervolgens voor ‘Goto implementation’ of druk op CTRL+F12.  De eisen qua aanpassing  De volgende zaken moeten minimaal worden aangepast om de app geschikt te maken voor de Debug Hero-attractie:   * Het uiterlijk van de app moet helemaal veranderen; in plaats van een donker horrorthema moet het een felgekleurd superheldenthema krijgen. * De wachtrij van Debug Hero is anders, waardoor de sensordata anders geïnterpreteerd moet worden: Er zijn twaalf sensoren die uitgelezen moeten worden. Bovendien staan de eerste twee sensoren van ‘Debug Hero’ beiden voor 6minuten wachttijd, terwijl de andere sensoren voor 4minuten wachttijd staan. * ‘Debug Hero’ heeft drie karren in plaats van twee. * Het is bij ‘Debug Hero’ mogelijk dat een van de drie karren in onderhoud is, terwijl de attractie wordt gebruikt. Een kar moet dus de status ‘in onderhoud’ kunnen krijgen |