

PIRATES VERSUS NINJAS

SE12 Practicumopdracht week 1

Niveau	★ ★ ☆ ☆ ☆
Versie	2014-05-05 Bas Michielsen
Doelen	Kennismaking met klassen.

Inleiding

Naar aanleiding van de oeverloze internetdiscussies over een gevecht tussen een Europese piraat en een Japanse Ninja ([Pirates versus Ninjas](#)), gaan we nu het allesbeslissende programma maken. De discussie stelt dat in een gevecht tussen



een piraat en een ninja het niet duidelijk is wie van de twee zal winnen. Er is zelfs een [voorwerp](#) in World of Warcraft dat je karakter tijdelijk in een piraat of in een ninja verandert zodat je kan [uitvechten](#) wie er sterker is. In de geschiedenis hebben piraten en ninjas nooit met elkaar gevochten dus er is geen historische basis waarop je deze discussie zou kunnen beëindigen, dus wij laten de computer dit bepalen aan de hand van een spelletje.

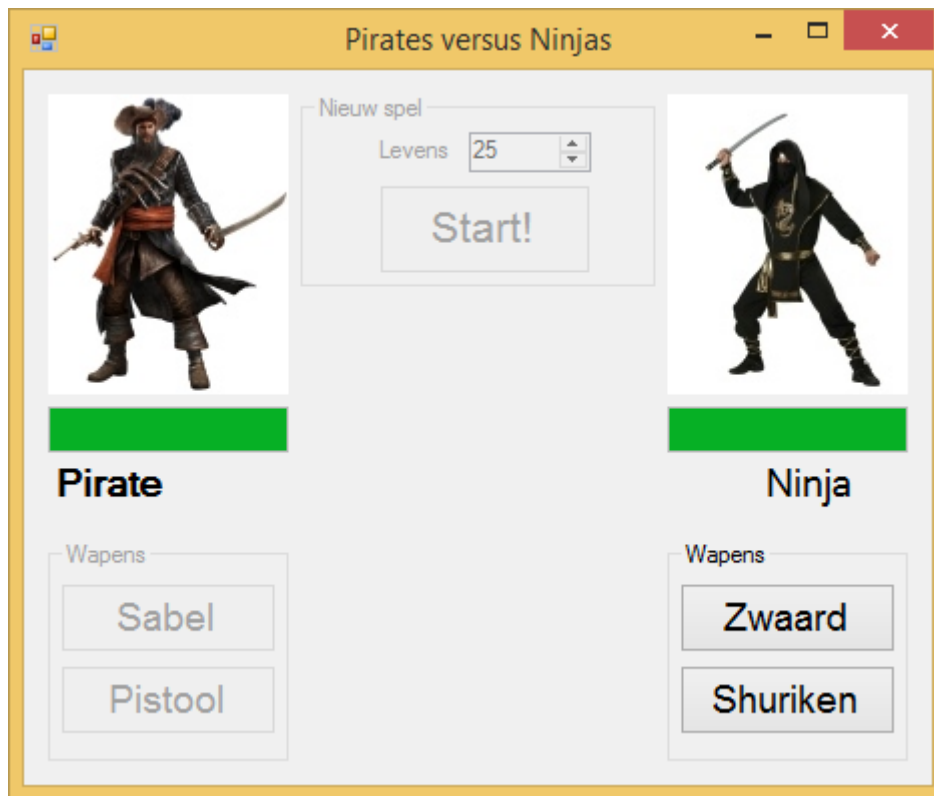


Wie denk jij dat zal winnen? En wie denkt de student naast je dat zal winnen en waarom?

In ons spel heeft elk van de karakters een gewoon wapen en een superwapen. Het superwapen kan 1 keer gebruikt worden en doet dan 9 schade. Als de speler het superwapen nog eens gebruikt doet het 0 schade. Het gewone wapen doet schade gebaseerd op het rollen van een dobbelsteen, dus minimaal 1 en maximaal 6. De piraat heeft een sabel en als superwapen een pistool, de ninja heeft een zwaard en als superwapen shuriken. Wie van de karakters kan het aantal levens van de ander het vlugst op 0 krijgen en de discussie voor eens en altijd beëindigen?

Opdracht

Voor Pirates versus Ninjas is er al een startproject beschikbaar op de portal. Download en open het startproject. Merk op dat het formulier al helemaal gegeven is, je hoeft dus zelf geen GUI meer te maken.



STAP 1

Je kunt het spel al starten en je kunt dan zien dat de GUI al werkt. Echter als je op de `Button` met de tekst "Start!" klikt vergaat de wereld van piraten en ninjas want dan crasht het spel. Er wordt een foutmelding gegeven omdat de variabele `dobbelsteen` nog geen waarde heeft. Laten we dit even corrigeren!

Ga naar de `EventHandler` van de `Button` Start! en kijk wat er gebeurt. Voeg op de aangegeven plek een stukje programmacode toe om de variabele `dobbelsteen` te initialiseren. Dit heb je bij OIS ook al eens geleerd.

Als je dit gedaan hebt kun je het spel weer starten. Je zult zien dat het nu al een stuk beter werkt! Het spel bepaalt willekeurig wie van de twee karakters mag beginnen, je kunt dat zien omdat 1 set wapens ingeschakeld (`Enabled`) is, en de andere set is uitgeschakeld (`Disabled`). Als je het spel afsluit en weer start kan het dus zijn dat het andere karakter mag beginnen.

STAP 2

Het spel gaat over 2 karakters, een piraat en een ninja, echter hebben we deze karakters nog niet aangemaakt in ons programma. Er staan al wel 2 plaatjes op het scherm maar de computer heeft nog geen weet van het feit dat wij bedoelen dat er twee karakters zijn met een hoeveelheid levens en een set wapens en dat deze karakters in gevecht dienen te gaan.

We gaan nu de twee karakters aanmaken. Hiervoor bestaat er al een klasse `Karakter`. Uiteraard hebben we 2 instanties van deze klasse nodig, 1 die we `piraat` noemen en een andere die we `ninja` noemen, de `Fields` waarin deze instanties bewaard worden zijn al aangemaakt. Het aanmaken van een instantie van de klasse `Karakter` gaat soortgelijk aan het aanmaken van een instantie van de klasse `Random`. Schrijf op de aangegeven plek de programmacode om twee instanties van de klasse `Karakter` aan te maken en deze op te slaan in de `Fields` `piraat` en `ninja`.

Het zal je opvallen dat we bij het aanmaken van een instantie van de klasse `Karakter` een tweetal parameters dienen mee te geven. `Visual Studio` geeft aan dat we een `Random spelDobbelsteen` en een `int karakterLevens` dienen mee te geven. De `spelDobbelsteen` is de variabele `dobbelsteen` die we zojuist geïnitieerd hebben. De `karakterLevens` dienen we uit het `Form` te lezen. Bij het starten van het spel kunnen de spelers het aantal levens bepalen dat de karakters hebben, die opgegeven waarde wordt er bedoeld. Bij OIS heb je geleerd hoe je de waarde van een `NumericUpDown` kunt uitlezen en hoe je deze waarde kunt omzetten naar een `int`. Pas je programmacode zo aan dat de twee gevraagde parameters aan beide instanties meegegeven worden.

STAP 3

Je kunt het spel nu starten, maar dan zie je dat beide karakters, ondanks dat we ze wel aanmaken, nog niet de ingestelde hoeveelheid levens krijgen. Bij beide karakters blijft het levensbalkje nog leeg. Dit komt omdat we in de klasse `Karakter` de parameters van de `Constructor` nog niet verwerkt hebben. Ze worden wel meegegeven naar de instantie van de klasse, maar we doen er nog niets mee.

Ga naar de `Constructor` van de klasse `Karakter`. Het is de bedoeling dat alle drie de `Fields` van deze klasse in de `Constructor` een waarde toegekend krijgen. Geef elk van de drie `Fields` een waarde, 2 van deze waardes zijn natuurlijk de parameters die je bij het aanroepen van de `Constructor` mee krijgt. Start het spel en controleer dat de levensbalkjes van beide karakters nu wel vollopen.

STAP 4

Als je op de wapenknoppen klikt zal je merken dat er nog niets gebeurt. Dit gaan we aanpassen. Het wordt tijd dat er wat gevochten gaat worden. Als de piraat aanvalt dan is de schade natuurlijk voor de ninja, en als de ninja aanvalt is de schade natuurlijk voor de piraat.

Programmeer eerst de `Button` Sabel van de piraat. De hoeveelheid schade wordt bepaald door de methode `ValAan` aan te roepen op de instantie van `Karakter` die opgeslagen is in de variabele `piraat`. Deze methode retourneert een integer, die je vervolgens als parameter meegeeft aan de methode `OntvangSchade` van de instantie van `Karakter` die opgeslagen is in de variabele `ninja`. Kortom de schade wordt opgevraagd bij de piraat en gegeven aan de ninja. Daarna dien je de spelstatus te verversen door de methode `VerversSpelstatus` aan te roepen, geef wederom de schade mee als parameter. Daarna wisselt de beurt, roep de methode `WisselBeurt` aan.

```
private void btnPirateWapen_Click(object sender, EventArgs e)
{
}
```

Doe daarna hetzelfde voor de `Button` Zwaard van de ninja, uiteraard wordt dan de schade opgevraagd bij de ninja en gegeven aan de piraat.

```
private void btnNinjaWapen_Click(object sender, EventArgs e)
{
}
```

STAP 5

Programmeer de methode `WisselBeurt`, je kunt controleren wie er nu aan de beurt is door te kijken welke van de wapensets ingeschakeld is (`Enabled`). Als het nu de beurt van de piraat is wordt het de beurt van ninja en andersom.

```
private void WisselBeurt()
{
}
```

STAP 6

Als je het spel nu gaat spelen merk je dat er altijd precies 0 schade gedaan wordt. Dat is natuurlijk niet zo leuk want zo wint er niemand. We kunnen dit aanpassen door in de klasse `Karakter` de methodes `ValAan` en `OntvangSchade` te implementeren. Programmeer deze methodes. De methode `ValAan` retourneert de hoeveelheid

schade die het karakter doet bij een aanval met zijn gewone wapen. De methode `OntvangSchade` verwacht een parameter `schade`, dit is de te ontvangen schade gebaseerd op de aanval van het andere karakter.

```
public int ValAan()
{
}

public void OntvangSchade(int schade)
{
}
```

STAP 7

De superwapens van de twee karakters werken nog niet, zo doen nu altijd 0 schade en dat is natuurlijk niet correct. Het is de bedoeling dat elk karakter zijn superwapen 1 keer kan gebruiken en het wapen doet dan altijd 9 schade. Als de speler daarna nog eens op de `Button` voor het superwapen klikt dient het 0 schade te doen. Programmeer de twee `EventHandlers` voor het klikken op de Buttons van de twee superwapens.

```
private void btnPirateSuperwapen_Click(object sender, EventArgs e)
{
}

private void btnNinjaSuperwapen_Click(object sender, EventArgs e)
{
}
```

Implementeer ook de methode `GebruikSuperwapen` in de klasse `Karakter`. Gebruik het Field `superwapenGebruikt` om bij te houden of dit karakter zijn superwapen al gebruikt heeft.

```
public int GebruikSuperwapen()
{
}
```

STAP 8

Het spel is bijna compleet, de piraat en de ninja gaan in gevecht en elk van hen kan zijn superwapen 1 maal gebruiken. Echter nog niemand kan winnen. Als de levens van een karakter op zijn gebeurt er niets. Implementeer de methodes `PirateGewonnen` en `NinjaGewonnen`. Je dient het spel te stoppen door de methode `StopSpel` aan te



roepen. Daarnaast is het leuk als er een bericht in beeld komt dat de winst van het winnende karakter vereeuwigd. Gebruik hiertoe een `MessageBox`. Speel het spel een aantal keren om te testen of alle functionaliteit correct werkt.



Uitbreidingen

NIVEAU

★★★★☆

★★★★☆

OMSCHRIJVING UITBREIDING

Pas het spel aan zodat het superwapen onbeperkt gebruikt kan worden. Het heeft nu echter 1/3 kans om 9 schade te doen en 2/3 kans om 0 schade te doen.

Bedenk zelf nog een ander wapen voor zowel de piraat als voor de ninja. Zorg er wel voor dat het wapen niet onevenredig sterk wordt en de spelers alleen dit wapen zullen gebruiken.