Robert-Jan Zandvoort en Arnoud de Jong

Mobile multiplayer location-based gaming

GAMING IS INNOVATIE

Op de International Mobile Gaming Awards, die in februari 2007 plaatsvond, heeft het spel Triangler de overall 'Grand Prix' en de 'Most Innovative Game' awards gewonnen. Het spel is ontwikkeld met behulp van .NET Framework 3.0 en het Compact Framework voor Windows Mobile 5. In dit artikel gaan we in op een aantal technologische aspecten achter het spel.

Je verwacht waarschijnlijk in eerste instantie niet dat TNO zich bezighoudt met het ontwikkelen van games. Toch kunnen games op diverse manieren ingezet worden anders dan alleen voor vermaak. Zo werkt TNO aan 'serious gaming' voor training van respons op crisissituaties en kunnen games erg goed gebruikt worden om nieuwe manieren van leren te ontwikkelen. Gaming is innovatie. Als showcase voor de expertise op dit gebied heeft TNO het spelconcept van Triangler uitgewerkt, gerealiseerd en ingezonden naar de International Mobile Gaming Awards. Op 15 februari heeft het spel uit meer dan 400 inzendingen de hoofdprijs gewonnen.

Spelconcept

Triangler speel je met twee teams van vijf spelers, en elke speler is uitgerust met een mobiele telefoon met locatiebepaling en GPRS-verbinding. Het spel wordt buiten gespeeld op een veld van ongeveer 400 bij 400 meter. Het doel van het spel is om met groepjes van drie teamgenoten op het speelveld gelijkzijdige driehoeken van 150 meter te vormen en in deze driehoeken tegenstanders in te sluiten. Voor elke tegenstander die ingesloten is, worden punten gescoord. Elke combinatie van spelers kan één keer per spel een driehoek vormen. Op de mobiele telefoon zie je als speler waar de andere spelers zich bevinden en kun je verschillende spelcommando's geven, zoals het uitnodigen van teamgenoten om een driehoek te vormen. In afbeelding 1 staat een screenshot van de client-applicatie. In eerste instantie was de jury van de IMGA redelijk sceptisch over het spelconcept. Triangler is namelijk niet de eerste game die locatie gebruikt in het spel en de ervaringen met deze eerdere games waren niet erg positief. Gelukkig hebben we hier een verandering in kunnen aanbrengen, door samen met de jury het spel te spelen. De jury was in een park in Barcelona aanwezig en ons team was in Groningen aan het spelen. Doordat we de beide speelvelden dezelfde grootte hadden gegeven, konden we deze virtueel over elkaar heenleggen, zodat beide teams toch tegen elkaar konden spelen.

Technische opzet

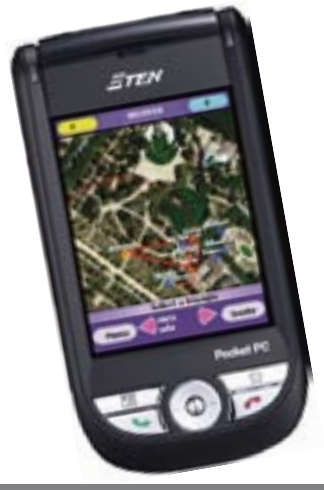
Het spel is ontwikkeld met behulp van het .NET Framework 3.0 aan de gameserver-kant, en het Compact Framework (Windows Mobile 5) aan de mobiele client-kant. De techniek achter Triangler bestaat uit een aantal onderdelen. Een belangrijk onderdeel is de mobiele client, waarmee je als speler bent uitgerust. Hiermee bedien je het spel en zie je waar jouw tegenstanders en medespelers zich bevinden. Een ander onderdeel is de gameserver waar alle mobiele clients mee verbonden zijn. De gameserver zorgt voor het naleven van de spelregels van Triangler en houdt alle 'states' bij, zoals de score, de posities van de spelers, informatie over het speelveld en welke driehoeken zijn gevormd. Om dit te doen ontvangt de gameserver de posities van alle spelers en de verstuurde commando's vanaf de verschillende mobiele clients en verwerkt deze. Ook bevat de server generieke functionaliteit voor geometrische berekeningen, zoals het berekenen van de afstand tussen GPS-posities en het bepalen of een persoon ingesloten zit in een driehoek. Verder is er ook een visualisatie-client voor de pc ontwikkeld om toeschouwers naar het spel te laten kijken. We gaan in rest van het artikel wat dieper in op een aantal van deze onderdelen.

Mobiele client

De mobiele client-applicatie moet in essentie vier dingen doen: als eerste moet de locatie van de speler kunnen worden bepaald. Ten tweede moet er een communicatieverbinding zijn met de gameserver, waarmee de locatie van spelers en spelcommando's wordt uitgewisseld. Op basis van de ontvangen events en de acties van de gebruiker moet er een game-state worden beheerd. Als laatste is er een user-interface nodig met daarop een kaart van het speelveld, de spelers en het verloop van het spel. De interface moet de gebruiker in staat stellen een aantal gamefuncties te bedienen en met zijn medespelers te interacteren.

Locatiebepaling

De locatiebepaling van de spelers gebeurt door middel van GPS. Sinds enkele maanden verschijnen er steeds meer mobiele telefoons



Afbeelding 1. De client-applicatie

```
using Microsoft.WindowsMobile.Samples.Location;

private Gps gps;

Public void InitGPS()
{
   gps = new Gps();
   gps.LocationChanged += new  LocationChangedEventHandler(
   gps_LocationChanged);
   gps.Open();
}

protected void gps_LocationChanged(object sender,
LocationChangedEventArgs args)
{
   Position = args.Position;
   if (gps.Opened && Position.LatitudeValid && Position.LongitudeValid)
      {
        // Position.Speed is in knots, convert to km/h
        float speed = Position.Speed * 1.94250;

        double Lat = Position.Latitude;
        double Lon = Position.Longitude;
      }
      ...
}
```

Codevoorbeeld 1.

op de markt met ingebouwde GPS-ontvanger. Dat is gemakkelijk als jij jouw favoriete routeplanner altijd bij de hand wilt hebben of op zoek wilt naar een volgende geocache, maar het wordt vooral leuk als je bedenkt wat je er zelf mee kunt maken. Met de komst van een GPS API in Windows Mobile 5 is het uitlezen van de GPS-positie sterk vereenvoudigd. Hoewel het een native API betreft, laat de SDK met een voorbeeld zien hoe met een paar regels code de positie achterhaald kan worden. Zie codevoorbeeld 1 hoe je de GPS-locatie kunt uitlezen. Voor Triangler konden we dus aan de slag om al snel tegen een aantal praktische problemen op te lopen. Zo werd het al vlug duidelijk dat het verkrijgen van een fix (een goede verbinding met de satellieten) soms wel tien minuten duurt. Daarnaast bleken de toestellen die wij hebben gebruikt niet alle over dezelfde firmware te beschikken, waardoor ze niet allemaal goed geconfigureerd waren voor het gebruik van de ingebouwde GPS-ontvanger.

User-interface

Tijdens het spel kunnen de spelers op een kaart zien waar hun medespelers en tegenstanders zijn. Wanneer er driehoeken worden gevormd, wilden we die op de kaart projecteren, zodat snel duidelijk is wat er zich afspeelt. De basiscontrols van het Compact Framework zijn hiervoor niet toereikend, zodat we wat dieper in de mogelijkheden van het framework moesten duiken. Twee events bleken cruciaal te zijn voor het verkrijgen van meer vrijheid bij de opbouw van het scherm: OnPaint en OnPaintBackground. Het OnPaint-event kan gebruikt worden om zelf op het scherm te tekenen, waarbij je gebruik kan maken van plaatjes, tekst, lijnen, enzovoort. Wat voor ons belangrijk was, was de mogelijkheid transparante plaatjes, ook wel sprites genoemd, te kunnen projecteren op het scherm. Vergelijkbaar met een transparant gif-bestand kan een kleur worden gekozen die niet zichtbaar is op het scherm. Alle spelers kunnen hierdoor goed op de kaart worden gezet, ook als ze dicht bij elkaar in de buurt zijn. Om te voorkomen dat het scherm gaat flikkeren of langzaam wordt opgebouwd, wordt het scherm eerst volledig in geheugen opgebouwd en bij de eerstvolgende aanroep van OnPaint op het scherm weergegeven. Deze techniek wordt ook wel 'double buffering' genoemd. Bij deze aanpak is het van belang om het OnPaintBackground-event te herschrijven en deze niets te laten doen. Hierdoor wordt voorkomen dat delen van de zojuist getekende achtergrond verloren gaan bij het gebruik van controls. In codevoorbeeld 2 wordt de basis van de schermopbouw verder verduidelijkt. Als laatste willen we het belang van geluid in ons spel aanstippen. Tijdens de eerste tests, bleken in de hectiek van het rennen en oriënteren geluiden bijna net zo belangrijk te zijn als visuele aanwijzingen. Je kunt niet continu op het scherm blijven kijken, dus we hebben zoveel mogelijk notificaties in het spel proberen te verwerken: wanneer je uitgenodigd wordt een triangle te vormen, wanneer je zelf ingesloten bent, wanneer een driehoek gelijkzijdig is, enzovoort. Wie meer wil doen met sprites en ze bijvoorbeeld ook wil laten animeren en wie wil zien hoe je geluid kunt afspelen, moet zeker eens gaan rondkijken op MSDN waar een paar heel mooie voorbeeldimplementaties van games met behulp van het Compact Framework te zien zijn.

Communicatie

Tijdens het spel zijn alle tien spelers voortdurend in beweging om zoveel mogelijk driehoeken te vormen en hiermee de tegenstanders in te sluiten. Voor een goed spelverloop is het belangrijk dat alle spelers continu de laatste details ontvangen van alle andere spelers en andere spelinformatie. De timing is hierbij van belang, omdat een paar seconden vertraging al punten kan kosten. Tijdens het spel hebben alle mobiele clients een verbinding met de gameserver. Deze verbinding gaat over het GPRS-netwerk van de mobiele aanbieder en dat brengt een aantal beperkingen met zich mee. Naast een beperkte snelheid is vooral de beschikbare capaciteit een belangrijke

```
private Bitmap mainImage;

...

private void DrawPlayer(string playerName, float playerPosX, float
playerPosY)
{
   using (Graphics g = Graphics.FromImage(mainImage))
   {
      Font basicFont = new Font(FontFamily.GenericSansSerif, 8,
      FontStyle.Bold);
      Brush brWhite = new SolidBrush(Color.White);

      g.DrawString(playerName, basicFont, brWhite, playerPosX-10,
      player PosY-10);

      Color TransparentColor = Color.FromArgb(254, 0, 254);
      ImageAttributes TransparentAttributes = new ImageAttributes();
      TransparentAttributes.SetColorKey(TransparentColor,
      TransparentColor);

      g.DrawImage(
        spriteBitmap,
        spriteRectangle,
        0, 0, 20, 20,
        GraphicsUnit.Pixel,
      Settings.TransparentAttributes );
}
   ...
private void OnPaint(object sender,
   System.Windows.Forms.PaintEventArgs e)
   {
      // laat de image op het scherm zien
      e.Graphics.DrawImage(mainImage, 0, 0);

   }

protected override void OnPaintBackground(PaintEventArgs e)
   {
      // de background moet niet gepaint worden
   }
```

Codevoorbeeld 2.

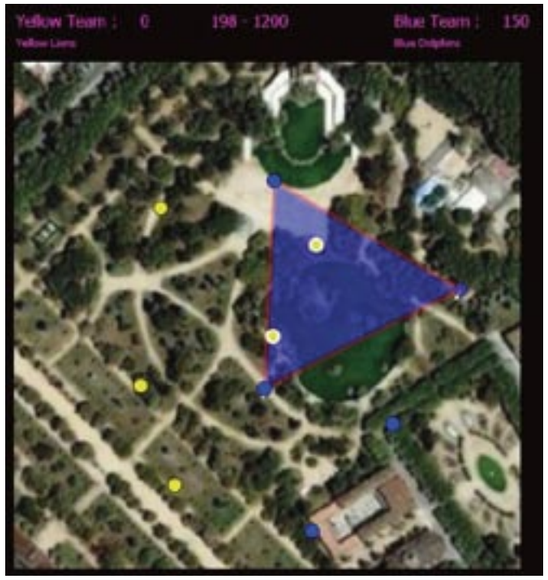
factor om rekening mee te houden. Met een relatief groot aantal mobieltjes op een klein oppervlak, hadden we niet de luxe om veel data over de dataverbinding te sturen. De beschikbare GPRS-capaciteit wordt namelijk gedeeld met alle personen in een mobiele cel. Om zo goed mogelijk met deze beperkingen om te gaan, konden we uitstekend gebruikmaken van de kennis die bij TNO aanwezig is over mobiele datacommunicatie.

Visualisatie-client

Om ervoor te zorgen dat niet alleen de spelers kunnen genieten van het spel, maar ook de toeschouwers, hebben we een voor de pc een visualisatieapplicatie gemaakt. Deze zorgt er voor dat alle acties die de spelers uitvoeren, zoals het bewegen en het vormen van driehoeken, zichtbaar zijn. In tegenstelling met de mobiele client hadden we op de pc een eenvoudigere manier om met grafische objecten om te gaan, namelijk Windows Presentation Foundation. Het doorzichtig laten zijn van bepaalde objecten, zoals de driehoeken, was hierdoor bijvoorbeeld erg eenvoudig. Ook de gecompliceerdere dingen

```
using System.Windows.Media.Animation

public void AnimateTriangle()
{
   Point point1From = new Point(10, 10);
   Point point1To = new Point(100, 120);
   Point point2From = new Point(100, 10);
   Point point2To = new Point(150, 30);
   Point point3From = new Point(50, 50);
   Point point3To = new Point(30, 80);
   PathGeometry triangleGeometry = new PathGeometry();
   PathFigure triangleFigure = new PathFigure();
   Path trianglePath = new Path();

   // Bouw de driehoek op uit drie lijnsegmenten
   LineSegment ls1 = new LineSegment(point1From, true);
   LineSegment ls2 = new LineSegment(point2From, true);
   LineSegment ls3 = new LineSegment(point3From, true);

   triangleFigure.StartPoint = point3From;
   triangleFigure.Segments.Add(ls1);
   triangleFigure.Segments.Add(ls2);
   triangleFigure.Segments.Add(ls3);
   triangleGeometry.Figures.Add(triangleFigure);

   // Definieer het pad
   trianglePath.Stroke = Brushes.Black;
   trianglePath.StrokeThickness = 3;
   trianglePath.Fill = Brushes.Blue;
   trianglePath.Data = triangleGeometry;

   // Voeg het pad toe aan een Canvas
   canvas1.Children.Add(trianglePath);

   // Maak animaties aan voor de hoekpunten van de driehoek
   PointAnimation pa1 = new PointAnimation(point1To, new Duration(new
   TimeSpan(0, 0, 4)));
   PointAnimation pa2 = new PointAnimation(point2To, new Duration(new
   TimeSpan(0, 0, 4)));
   PointAnimation pa3 = new PointAnimation(point3To, new Duration(new
   TimeSpan(0, 0, 4)));

   // Animeer de lijnsegmenten waaruit de driehoek bestaat
   ls1.BeginAnimation(LineSegment.PointProperty, pa1);
   ls2.BeginAnimation(LineSegment.PointProperty, pa2);
   ls3.BeginAnimation(LineSegment.PointProperty, pa3);
   triangleFigure.BeginAnimation(PathFigure.StartPointProperty, pa3);
}
```

Codevoorbeeld 3.



Afbeelding 2. Een plaatje van het speloverzicht

zoals het animeren van polygonen is met enig internetspeurwerk gelukt. Een uitwerking hiervan kun je zien in codevoorbeeld 3. We gaan ervan uit dat je een WPF-applicatie hebt gemaakt met hierin een Canvas-object genaamd 'canvas1'. Om een driehoek weer te geven, wordt gebruik gemaakt van een 'PathGeometry'-object waarin zich drie lijnstukken bevatten die samen de driehoek voorstellen. De eindpunten van de lijnstukken kunnen vervolgens geanimeerd worden. In afbeelding 2 wordt een plaatje getoond van het speloverzicht dat een toeschouwer heeft.

Spelregels

Voor de implementatie van de gamelogica en spelregels wordt gebruikgemaakt van de state machine-workflow van Windows Workflow Foundation. Voor ons was dit project een mooie gelegenheid om deze technologie in de praktijk te kunnen uitproberen. WF biedt een aantal eigenschappen dat goed in het spel te gebruiken is. Zo worden de spelregels op een begrijpbare manier visueel inzichtelijk gemaakt en kunnen deze eenvoudig aangepast worden. Er is op dit moment bijvoorbeeld de regel dat een combinatie van spelers maar één keer per spel een driehoek mag maken. Wil je deze regel aanpassen, dan is dit eenvoudig te doen, en wordt dit ook duidelijk zichtbaar in de grafische weergave van de state machine in Visual Studio. Als we de state machine zonder Workflow Foundation hadden geïmplementeerd, zou je de code in moeten duiken om te achterhalen wat er precies gebeurt.

Innovatieve applicaties

Nu mobiele telefoons vaker uitgerust worden met een GPS-ontvanger en een dataconnectie steeds goedkoper wordt, krijgen we steeds meer applicaties te zien die hiervan op innovatieve wijze gebruik gaan maken. We zijn erg benieuwd wat de toekomst ons hierin zal brengen en dagen jullie allen uit om creatieve ideeën te verzinnen. Wat betekent locatie voor jouw mobiele applicatie?

Robert-Jan Zandvoort (robert-jan.zandvoort@tno.nl) en **Arnoud de Jong** (arnoud.dejong@tno.nl) zijn werkzaam bij TNO Informatie- en Communicatietechnologie.

Referenties

Triangler-informatie: http://www.tno.nl/triangler
IMGA-website: http://www.imgawards.com/
Videoverslag van gamer.tv: http://imgawards.tmp01.haisoft.net/tv_feb2007/