

Maak je eigen game met Windows Phone 7

VAN SPELIDEE TOT COMMERCIEEL SUCCES

Clemens Schotte

Hoe ontwikkel je een game voor de Windows Phone 7, gebruikmakend van het XNA framework en hoe zorg je dat de marketing van je net ontwikkelde game bijdraagt aan het verwachte succes ervan. In dit artikel gaan we verder in hoe je van een spelidee een succes kan maken met Windows Phone 7.

Om voor Windows Phone 7 software te ontwikkelen heb je de gratis tools nodig van Microsoft. Hierin vindt je onder andere Visual Studio 2010 Express (IDE), XNA Game Studio 4.0 en een Phone 7 Emulator. Wil je daarnaast ook je net ontwikkelde spel in de marktplaats van Windows Phone 7 zetten dan dien je hiervoor een abonnement te hebben bij App Hub van Microsoft. Ben je een DreamSpark student dan is dit abonnement zelfs gratis.

Gameontwikkeling kent verschillende fases die doorlopen worden die anders zijn dan in de traditionele softwareontwikkeling. Het is te vergelijken met het maken van een bioscoopfilm die interactief is. Het begint allemaal met het verhaal, dat wil zeggen: wat wil jij dat de speler ervaart. Dan moet je gaan bedenken hoe je dit kunt gaan realiseren en eventueel met wie. Denk hierbij of je de volgende punten zelf kunt uitvoeren of dat je iemand hiervoor nodig hebt: de grafische weergaven, game logica ontwikkeling, muziek, geluidseffecten en leveldesign.

Welke technologie ga je gebruiken voor je spel, Silverlight of XNA? Windows Phone 7 kan namelijk beide aan. Dit is sterk afhankelijk van de vraag wat voor een spel je hebt bedacht. Ervan uitgaande dat een spel grafisch veeleisend is, is de keuze voor XNA al snel gemaakt. Het scherm van een Windows Phone 7 toestel is 800 bij 480 punten (liggend), dat is best veel voor zo'n klein oppervlak. Tijdens het creëren van je grafische elementen op de computer merk je dat dit op een echt toestel een stuk kleiner uitvalt. Bekijk regelmatig hoe afbeeldingen op een Phone 7 toestel overkomen.

Standaard zal een XNA game niet volledig schermvullend zijn, er blijft een kleine rand over voor de statusbar van je Phone 7. Wil je gebruikmaken van het volledige scherm vergeet dan niet dit aan te zetten door middel van de volgende code: `graphics.IsFullScreen = true;`

Spelconcept

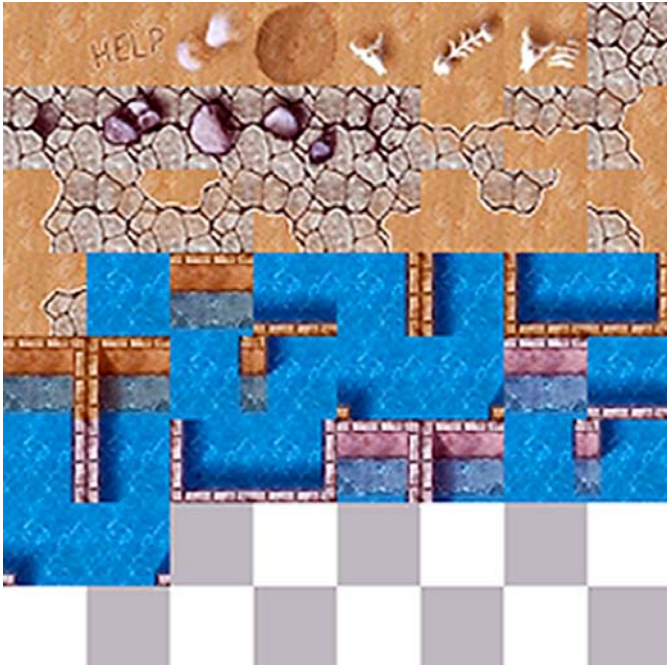
Laten we voor dit artikel een spelconcept bedenken: Een mijnwerker werkt in een mijn om kostbare edelstenen te delven. Hij gebruikt explosies om zich een weg door de mijn te banen. Niet al het gesteente kan worden opgeblazen omdat dit te sterk is. De mijnwerker heeft bovendien te kampen met monsters in de mijn die zijn werk in de weg zitten.

Dit betekent dat we een mijn (ondergrond), monsters, edelstenen, explosies en een mijnwerker nodig hebben in beeld en geluid. Daarnaast moeten we dit in code combineren tot een speelbaar spel. Laten we beginnen met de grafische afbeeldingen. Om het niet meteen te ingewikkeld te maken zal het spel 2D zijn. We gebruiken verschillende grafische lagen over elkaar heen voor de verschillende elementen; Laag 0 ondergrond (de mijn), laag 1 de stenen, laag 2 de monsters, etc. De grafische elementen gaan we met behulp van tegels op het scherm tekenen. Dit is te vergelijken met een mozaïek, maar dan 6 lagen over elkaar heen. Als elke tegel 32 bij 32 pixels is kunnen we 25 bij 15 tegels tekenen op elk scherm (800/32 en 480/32). Om er voor te zorgen dat als die verschillende lagen over elkaar heen liggen we de onderste laag nog kunnen zien maken we gebruik van een Alpha kanaal, dat per pixel de doorzichtigheid kan aangeven. Hierdoor ontstaat de mogelijkheid om te spelen met deze effecten, zoals het toevoegen van een schaduw bij een monster of een steen op de ondergrond. Zorg er voor dat je de afbeeldingen opslaat in het formaat PNG, aangezien hierbij het Alpha kanaal bewaard blijft. Per onderdeel maken we maar 1 afbeelding (texture) waarop we de verschillende tegels plaatsen, dus alle monsters staan op één texture die we opslaan als `monsters.png` en alle stenen op een andere texture die we opslaan als `stones.png`.

Nu dat we de meeste grafische elementen hebben kunnen we ons gaan concentreren op de gameloga. Hiervoor maken we gebruik



FIGUUR 1 - EEN XNA GAME OP DE WINDOWS PHONE 7 EMULATOR



FIGUUR 2 – TEGEL SET VAN DE ONDERGROND (DESERT MAP)

van de XNA Game Studio. Jij bent verantwoordelijk voor alles wat je ziet en hoort in je game. Je tekent elk scherm 30 keer per seconde opnieuw, dit gebeurt in de renderloop. Binnen de renderloop moet je weten in welke fase je game zich bevindt, zodat je de bijbehorende schermen kan tonen. Bevindt de speler zich bijvoorbeeld in het hoofdmenu of is hij de game aan het spelen.

```
protected override void Draw(GameTime gameTime)
{
    switch (gameStatus)
    {
        case GameStatus.Menu:
            DrawMenu();
            break;

        case GameStatus.Start:
            player = new Player();
            GameStart(player.levelId);
            break;

        case GameStatus.Resume:
            GameResume();
            break;

        case GameStatus.Playing:
            DrawPlaying();
            break;
    }

    base.Draw(gameTime);
}
```

Ervan uitgaande dat de gamestatus in de spelstand staat tekenen we de 6 verschillende lagen over elkaar heen zodat het speelveld zichtbaar wordt. We kunnen niet onbeperkt doorgaan met lagen over elkaar heen tekenen, je moet rekeninghouden dat dit alles binnen de tijd van 1 scherm ($1s / 30frames = 0,033s$) moet plaatsvinden. Om elke tegel uit de texture op de juiste positie te krijgen gebruiken we de volgende methode:

```
BlitSprite().
void BlitSprite(int x, int y, Texture2D texture, int index, int pixels)
{
    int width = texture.Width / pixels;
```

```
int xpos = (index % width) * pixels;
int ypos = index > 0 ? (index / width) * pixels : 0;

spriteBatch.Draw(texture, new Vector2(x, y),
    new Rectangle(xpos, ypos, pixels, pixels), Color.White);
}
```

Als we `BlitSprite()` combineren met `DrawMap()` hebben we het complete speelveld.

```
void DrawMap(Texture2D texture, byte[,] items, bool first)
{
    for (int y = 0; y < map.height; y++)
        for (int x = 0; x < map.width; x++)
            if (items[x, y] != 0 || first)

        BlitSprite(x * pixels, y * pixels, texture, items[x, y], pixels);
}
```

Aangezien we voor de Windows Phone 7 ontwikkelden dienen we rekening te houden met verschillende events die kunnen optreden, zoals iemand die gebeld wordt tijdens het spelen van een spel. Je wilt dat je spel daarna weer netjes verder gaat. Hetzelfde geldt als iemand op de Home knop drukt. We dienen dus alle parameters die nodig zijn om het spel verder te kunnen spelen snel te bewaren.

```
private void InitializePhoneServices()
{
    PhoneApplicationService.Current.Activated += Current_
        Activated;
    PhoneApplicationService.Current.Deactivated += Current_
        Deactivated;
    PhoneApplicationService.Current.Closing += Current_
        Closing;
    PhoneApplicationService.Current.Launching += Current_
        Launching;
}
```

We hebben geen rechtstreekse toegang tot het bestandstelsel van Windows Phone 7 en moeten hiervoor gebruikmaken van `IsolatedStorage`. Dit heeft als voordeel dat er geen resten achterblijven als je spel wordt verwijderd, want het bijbehorende `IsolatedStorage` wordt dan ook verwijderd. Zodra we toegang hebben tot `IsolatedStorage` kunnen we gewoon schrijven en lezen van bestanden. Let op dat als je een verbeterde versie van je spel uitbrengt het bijbehorende `IsolatedStorage` bewaard blijft en dus ook moet werken met je nieuwe versie van je spel. Stel je `gamestate.v1` moet uitgebreid worden voor versie 2 van je spel, dan dien je de oude `gamestate` om te zetten naar `gamestate.v2` zonder dat de speler daar iets van merkt. Het komt namelijk regelmatig voor dat ontwikkelaars dit vergeten, waardoor hun spelupdate niet door de certificeringtests van Microsoft komt.

```
using (IsolatedStorageFile isolatedStorageFile =
    IsolatedStorageFile.GetUserStoreForApplication())
{
    // If user choose to save, create a new file
    using (IsolatedStorageFileStream fileStream =
        isolatedStorageFile.CreateFile("gamestate.v1"))
    {
        using (StreamWriter streamWriter = new
            StreamWriter(fileStream))
        {
            // Write date to the file
            streamWriter.WriteLine(player.levelId);
            streamWriter.WriteLine(player.lives);
            ...
        }
    }
}
```

```

...
    streamWriter.Close();
}
}
}

```

Nu wordt het tijd dat wat we tekenen op het scherm ook laten bewegen. Je hebt op de Windows Phone 7 verschillende manieren om aan gebruikersinvoer te komen voor je spel: het scherm aanraken met 1 vinger (muis), meerdere vingers beweging (gesture) of de giroscop sensor (Sensors). De enige gamepad functie die we kunnen gebruiken is de terug-knop, dit is verder niet erg nuttig voor het spelen van het spel.

Het uitlezen van gebruikersinvoer en het bijwerken van alle spel-elementen (zoals de posities van de speler en de monsters) gebeurt in een aparte updateloop die 'los' staat van de renderloop. Net zoals in de renderloop moeten we weten in welke game fase we ons bevinden. In onderstaande code is te zien hoe we een punt op het scherm dat binnen een driehoek wordt aangeraakt omzetten naar een bepaalde beweging (moveAction) voor de mijnwerker op het scherm.

```

MouseState ms = Mouse.GetState();
if (ms.LeftButton == ButtonState.Pressed)
{
    if (isPointInsideTriangle( new Vector2(640, 320),
                               new Vector2(720, 400),
                               new Vector2(640, 480),
                               new Vector2(ms.X, ms.Y))) moveAction = Keys.Left;
    ...
    ...
}

```

Nu we weten welke kant de mijnwerker op moet, moeten we gaan controleren of dat wel kan; loopt de mijnwerker niet van het speelveld af? Loopt de mijnwerker niet op water (tegels onder 25 zijn grond, daarboven water)? Liggen er geen stenen of een bom in de weg? Dan pas kan de mijnwerker echt gaan bewegen. We geven de mijnwerker een snelheid van 2 pixels per frame, dat komt er op neer dat we iets meer dan een seconde nodig hebben om de mijnwerker op het volgende tegeltje te krijgen, hierna kan de mijnwerker pas weer een andere kant op.

```

switch (moveAction)
{
    case Keys.Down:
        if ((player.y / pixels < map.height - 1) &&
            (map.background[(player.x / pixels), (player.y / pixels) + 1] < 25)
            &&
            (map.stone[(player.x / pixels), (player.y / pixels) + 1] < 5)
            &&
            (map.bomb[(player.x / pixels), (player.y / pixels) + 1] == 0))
        {
            player.id = 1;
            player.ymov = 2;
        }
        break;
    ...
    ...
}

```

Ditzelfde mechanisme kunnen we ook gebruiken voor elk monster. Echter daar bepaalt de computer welke kant de monsters op moeten gaan. Je kunt hiervoor een heel ingewikkeld en slim algoritme verzinnen of de monsters gewoon een willekeurige kant op-

sturen. Ons 'slimme' Artificial intelligence (AI) algoritme ziet er als volgt uit: `int direction = random.Next() % 4;`

Nu bewegen de monsters en de mijnwerker over het veld, maar kunnen ze elkaar nog niet vernietigen. We moeten dus gaan controleren of ze elkaar aanraken.

```

void CheckEnemyHit()
{
    foreach (Enemy enemy in enemys)
    {
        if (enemy.x == player.x && enemy.y == player.y)
        {
            player.energy -= 2;

            if (player.energy < 0)
            {
                player.energy = 0;
            }
        }
    }
}

```

Het oppakken van items door de mijnwerker, om bijvoorbeeld extra energie te krijgen, is door te bepalen of de positie van de mijnwerker op dezelfde tegel valt als waar de power-up staat.

```

// power
if (map.stone[player.x / pixels, player.y / pixels] == 1)
{
    itemSound.Play();

    map.stone[player.x / pixels, player.y / pixels] = 0;

    if (++player.power > 4) player.power = 4;

    player.points += 50;
}

```

Uiteindelijk is je spel klaar, wat nu? Binnen Windows Phone 7 is de enige manier om software te installeren via de Phone 7 marktplaats. Heb je al bedacht wat je spel mag kosten of wordt het gratis? Wat doen andere spellen in jouw categorie? Maak je spel niet

Heb je al bedacht wat je spel mag kosten? Maak het niet te duur, maar zeker niet te goedkoop!

te duur, maar ook zeker niet te goedkoop. Als het te goedkoop is zullen mensen denken dat het niet veel voorstelt. Gratis games doen het altijd goed, dus een betaalde game dient (om succesvol te worden) een probeermode te hebben. Je kunt alleen de eerste 5 niveaus gratis laten spelen.

Tip 1: bouw een probeermode in je spel.

```

LicenseInformation license = new LicenseInformation();
isTrial = license.IsTrial();

```

Leuk zo'n probeermode, maar hoe gaan spelers dan mijn spel kopen? Hiervoor dienen drie zaken in balans te zijn:

1. Het spel is niet te duur, zeg 99 cent. Dat is goedkoper dan een bakje koffie en geeft je langer plezier.



FIGUUR 3 - VOORTGANG VAN HET AANTAL DOWNLOADS

2. De speler kan makkelijk van de probeermode naar de volledige versie van je spel gaan.
3. In je probeermode heb je de speler genoeg weten te boeien om te gaan kopen en niet zo veel weggegeven dat hij het niet meer hoeft te kopen. Het mooie van Windows Phone 7 is dat je maar één spel hoeft te bouwen, dus geen aparte probeer versie.

Tip 2: Zorg dat een speler makkelijk vanuit je probeermode de volledige versie kan kopen.

```
MarketplaceDetailTask marketplace = new MarketplaceDetailTask();
marketplace.Show();
```

Om je spel op de marktplaats verder onder de aandacht te brengen, dien je het spel van een pakkende maar bondige introtekst te voorzien. Mensen lezen niet graag. Een opvallend icoon en scherm-afdrukken vanuit verschillende tijdstippen in je game moeten genoeg interesse wekken om in ieder geval de probeermode van je spel te downloaden.

Tip 3: Zorg ervoor dat spelers vanuit je spel direct jouw spel kunnen beoordelen in de marktplaats.




LOOPT DIE MIJNWERKER NIET OP WATER, OF VAN HET VELD AF?

Meer positieve reacties, uitgedrukt in sterretjes op de marktplaats, zullen andere spelers sneller doen overhalen.

Nu staat je spel in de Windows Phone 7 marktplaats. Je hebt een probeerversie, is niet te duur en je hebt pakkende afbeeldingen en teksten. Hoe weet je dan dat het goed gaat met de verkoop of juist slecht? Moet je toch een andere prijs voeren, etc. Gelukkig kun je op de App Hub de voortgang van je spel volgen.

Tip 4: Maak naast reclame in de Phone 7 marktplaats ook reclame voor je spel op andere media, zoals facebook en twitter.

Of je nu een zelfstandig ontwikkelaar bent of een gamestudio, iedereen kan met het juiste idee een succesvol spel ontwikkelen voor Windows Phone 7. Je hebt alleen de juiste ingrediënten nodig en de passie om het tot een succes te maken. Laat je niet belemmeren door je eigen fantasie. Veel plezier in het bedenken en ontwikkelen van je nieuwe spel.

Wil je het spel CaveRace, wat we in dit artikel hebben gebruikt als voorbeeld, spelen? Probeer CaveRace voor Windows Phone 7 gratis uit op de volgende link: <http://bit.ly/bWPRvS> 



.....
Clemens Schotte, is consultant bij Microsoft. Hij is te bereiken op clemens.schotte@microsoft.com

Livescribe schrijft en luistert mee

SMARTPEN MET CAMERA REGISTREERT TEKST EN AUDIO

Robert de Ruiter

Tijdens klantgesprekken maak je aantekeningen. Je werkt die zo snel mogelijk uit om later misverstanden te voorkomen. Maar onder tijdsdruk blijft die uitwerking liggen. Dan blijkt het een paar dagen later moeilijk terug te halen wat de klant precies zei over... Een nieuw gadget van Livescribe, dat sinds januari in een Nederlandse versie wordt geleverd, is voor dit soort situaties een handig hulpmiddel.

Het is een smartpen, die met behulp van een ingebouwde infrarood camera de aantekeningen en tegelijkertijd ook het geluid opneemt. De audio-opname kan worden teruggespeeld door met de pen op een bepaalde plek in de aantekeningen te tippen.

De opname geeft het gesprek dan weer vanaf het moment, waarop de betreffende aantekening is gemaakt.

De kwaliteit van het geluid is uiteraard afhankelijk van de setting waarin het is opgenomen. Voor een een-op-een gesprek of een gesprek met enkele mensen rond een tafel voldoet de pen uitstekend. Voor opnames in een grotere ruimte, bijvoorbeeld een congres- of collegezaal, wordt een set oordopjes met ingebouwde microfoontjes meegeleverd. Hierdoor kan dan ook stereogeluid worden opgenomen.

De Livescribe smartpen is makkelijk in gebruik voor iedereen die overweg kan met pen en papier. De pen is ideaal voor professionals die veel aantekeningen maken. Alle vluchtig geschreven aantekeningen, telefoonnummers of e-mailadressen worden opgeslagen. Ook het notuleren tijdens vergaderingen en meetings of het opnemen van interviews is gemakkelijk met behulp van deze smartpen. Het geheim van de smartpen schuilt in 'Dot Paper' en een Dot Positioning System (DPS): een technologie die ervoor zorgt dat gewone notities omgezet worden in live documenten.

De microdots die geprint zijn op het papier zorgen ervoor dat de ingebouwde infrarood camera alles ziet en opslaat wat de smartpen schrijft. Naast alle notitieblokken en schriften van microdot papier is het ook mogelijk om zelf microdot papier te printen. De kleurenprinter moet dan wel een resolutie van 600dpi of meer hebben.

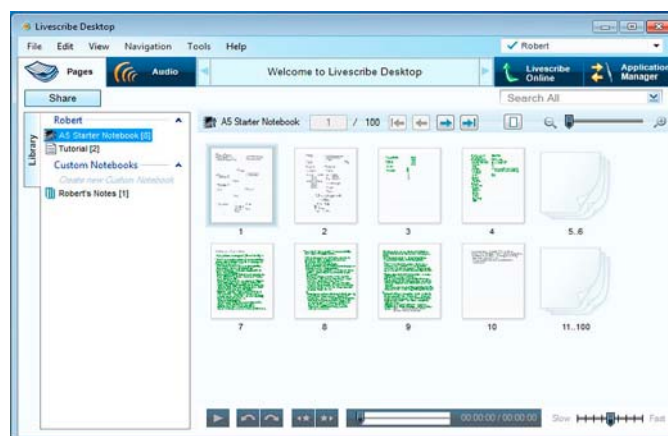
In de pen zijn nog een aantal grappige extra's ingebouwd. Zo kan de pen vertalen van engels naar enkele andere talen. De woordenschat van deze vertaler is echter uiterst beperkt en hooguit bruikbaar voor toeristische doeleinden. Verder kan de pen met behulp van een in het blocknote afgedrukte rekenmachine rekenen en geeft hij tijd, datum, batterijspanning en geheugencapaciteit weer. De software, die na aanschaf van de pen van de website van livescribe kan worden gedownload, werkt uiterst intuïtief een eenvoud-

dig. Het bevat overigens geen tekstherkenning; de geschreven aantekeningen worden wel in beeld gebracht, maar kunnen niet in digitale tekst worden omgezet.

Handig hulpmiddel

In de praktijk blijkt de smartpen inderdaad een uiterst handig hulpmiddel. De batterij van de pen wordt opgeladen via een usb-aansluiting en kan dan urenlang notities en geluid opnemen. De pen wordt geleverd met 2, 4 of 8 GB geheugen. De 2 GB versie, die wordt verkocht voor € 149,-, kan ongeveer 50 uur gegevens opslaan. Deze kunnen worden ingeladen in laptop of pc om ze eventueel langer te bewaren. Het geheugen van de pen kan selectief (of geheel) worden geschoond. De pen wordt geleverd met een A5 notitieblok met 'Dot Paper' en een extra schrijfstift. Er is – tegen acceptabele prijzen – een range van papier en accessoires verkrijgbaar bij internetshops en enkele grote elektronica ketens.

Meer informatie op de website van distributeur Navstar: www.navstar.nl.



DE GEGEVENS VAN DE SMARTPEN WORDEN OP PC OF LAPTOP INGELEZEN ZODRA DE PEN OP HET DOCKINGSTATION WORDT GELEGD.