

Codeglue ontwikkelt game in XNA Game Studio 2.0

“GAME-ONTWIKKELING VERGT VEEL KENNIS VAN HET PLATFORM”

In Historisch Delfshaven in Rotterdam zijn we op bezoek bij Codeglue. Sinds 2000 ontwikkelen zij arcadegames voor mobieltjes en handheld devices. Momenteel wordt de laatste hand gelegd aan hun nieuwste game ‘Rocket Riot’. Deze game is ontwikkeld in XNA Game Studio 2.0 en zal uitkomen voor de Xbox 360.

Peter de Jong is managing director en heeft samen met Maurice Sibrandi Codeglue opgericht. Maurice is verantwoordelijk voor het technische gedeelte. “Het is begonnen met mobile games, games voor mobiele telefoons”, vertelt Peter. “Tegenwoordig zijn we Nintendo DS en Xbox 360 certified developer. Tot nu toe hebben we acht games ontwikkeld en daar zitten kleine en grote projecten bij. We doen alles zelf, gamedesign, programmeren, animaties, level-design, game story, het testen, et cetera, alleen het componeren van de audio besteden we uit. THQ wordt de uitgever van Rocket Riot. Per development-milestone sturen we hen een versie. Naast het testen dat wijzelf doen, testen zij de game ook uitgebreid. We zijn nu met tien mensen en hebben voornamelijk games gemaakt voor mobiele telefoons. Anderhalf jaar geleden zijn we begonnen ons te richten op andere gameplatformen zoals Xbox Live Arcade. Met Managed DirectX hadden we al ervaring en toen XNA Game Studio uitkwam, zijn we direct met de eerste bèta’s begonnen. XNA is een mooi platform voor ons, omdat je met een relatief klein team crossplatform games kunt maken. Toen we hoorden dat je in een keer een game kunt maken voor zowel Xbox 360 als de pc, gaf dat de doorslag om met XNA te beginnen. Zonder dat we Xbox 360-hardware hadden, zijn we direct begonnen. We zijn ook direct lid geworden van de XNA Creators Club en de dag na de aanmelding hadden we onze game op de Xbox 360 draaien. Dat geeft voor ons het succes weer van XNA.”

Sinds 13 december is XNA Game Studio 2.0 gratis beschikbaar en werkt het met Visual Studio Maurice: “Onze eerste game die we met XNA Game Studio ontwikkelen heet Rocket Riot. Aan deze game werken we nu ongeveer tien maanden en daarvoor hebben nog wat tijd besteed aan een prototype. We zijn nu met de laatste milestones bezig en op dit moment gebruiken we een prototype van XNA versie 2.0. Deze versie van XNA is ook geschikt is om de certificatie van Microsoft te halen. Alle titels voor de Xbox 360 worden namelijk getest door Microsoft. Dit test- en certificatieproces duurt ongeveer twee maanden; als je het in één keer haalt. Het is een streng programma en Microsoft kijkt naar alles. Van hoe je Xbox Live schrijft, hoe je onderdelen van het spel noemt - het zijn bijvoorbeeld ‘achievements’ en geen ‘awards’ - tot aan allerlei zwaar technische aspecten. De porting van onze game van XNA 1.0 naar 2.0 was eigenlijk vrij eenvoudig. Veel onderdelen zijn verbeterd of eenvoudiger. Een probleem dat we hadden met versie 1.0 was eigenlijk de documentatie of beter het ontbreken daarvan. Nu is er veel informatie beschikbaar en kun je snel aan de gang.”

Henry Falconer is lead developer voor Rocket Riot. Hij is afgestudeerd aan het Imperial College Londen. Na een eerdere stageperiode bij Codeglue is hij na zijn studie direct begonnen aan Rocket Riot. “Een van de zaken waar we direct tegenaan liepen is de performance

op de Xbox 360 vergeleken met die van een pc. De architectuur van de Xbox 360 is zo anders dat je er gedegen rekening mee moet houden. Toen we de eerste keer onze game op de Xbox 360 speelden, draaide die langzamer dan op de pc. De Xbox heeft een lagere memory bandwidth dan een pc. Er werd in de game erg veel gelezen en naar geheugen geschreven. Met geheugen gaan we nu anders om. Als een visueel object in het spel bijvoorbeeld moet verdwijnen, dan is het niet slim om dat object uit de array met alle visuele objecten te halen. Iets uit het midden van een array verwijderen kost veel tijd, want alles wat er onder komt, moet worden opgeschoven. Het is beter is om het betreffende object even op hidden te zetten via een property. Zo hebben we de hele game geoptimaliseerd en nu draait het op de Xbox sneller dan op de pc.”

Peter: “De game-engine voor Rocket Riot is door ons zelf ontwikkeld. Daar gaat ontzettend veel tijd in zitten. Ik schat dat ongeveer 50% van alle tijd is gegaan in het bouwen van de game-engine. Het ontwerp van Rocket Riot is zo specifiek dat er geen engine voor bestaat. Het is geen standaard 3D-game waarvoor we een kant-en-klare engine kunnen kopen.” Maurice: “Naast de games schrijven we ook zelf veel tools. Voor de content-pipeline van XNA hebben we zelf importers geschreven. Een van de dingen die de content-pipeline namelijk niet doet is compressie. Je kunt custom importers en custom processors schrijven voor de XNA Framework Content Pipeline. Met de importers die we zelf hebben geschreven, zijn we van 200MB naar 50MB geslonken. Ook hebben we een traditionele WinForms-applicatie geschreven voor de level-editor. De levels worden weggeschreven in een xml-file en in de game geïmporteerd.”



Afbeelding 1. V.l.n.r. Maurice Sibrandi, Henry Falconer en Peter de Jong, (Foto: Hans Oostrum fotografie)



Afbeelding 2. Rocket Riot

XNA Game Studio 2.0 biedt nu ook networking-support in het XNA Framework. “Network latency is het grootste probleem bij online games”, zegt Henry. “De key in network-gaming is om te definiëren welke machine verantwoordelijk is voor welk onderdeel. Mijn machine bijvoorbeeld is altijd verantwoordelijk voor mijn player’s karakter en bepaalt dus exclusief waar het karaktertje zich bevindt in het speelveld. Per game-onderdeel heb je dus één autoriteit, anders krijg je problemen.” Maurice vult het relaas van zijn collega aan. “Timing is een van de zaken waar je bij game-development altijd mee te maken hebt. Er is een groot verschil tussen applicatie- en game-development. Een game is een real-time systeem, er worden veel zware berekeningen uitgevoerd en dat mag het verloop van het spel niet beïnvloeden. Bij kantoorapplicaties is de user-interface het grootste gedeelte van de tijd aan het wachten op input van de gebruiker. De informatie die op het scherm getoond, wordt is vaak statisch. Wanneer de gebruiker eindelijk een opdracht geeft, gaat de applicatie een berekening uitvoe-

ren en als het resultaat na een paar seconden komt, vindt niemand dat vreemd. Bij games is het precies omgekeerd, een game is altijd busy, het scherm wordt constant geüpdate, er is een voortdurende stroom van input van de gebruiker en continu worden zware berekeningen verwerkt met op de achtergrond ook nog muziek en geluidseffecten. Je loopt constant tegen de grenzen aan van het device of de console en dat vergt veel kennis van het platform. Het is in wezen een gesloten systeem en je moet real-time alles bijhouden, de score, positie van de spelers, de AI-update.”

Henry: “De CLR op de Xbox is een stuk eenvoudiger dan die van de pc-versie. De garbage collector bijvoorbeeld is erg simpel, die kent geen generations. De garbage collector doet alles ineens wanneer het een collection moet doen. Een truc is om dit zo min mogelijk te laten plaatsvinden in de game. Ik werk zo nauwelijks met dynamic calculations, en gebruik pools en hergebruik objecten. Een ander groot verschil met de pc is de processor van de Xbox 360, dat is namelijk een Xenon-processor gebaseerd op de IBM PowerPC. Deze chip heeft drie onafhankelijke cores en elke core heeft twee symmetrische hardware-threads. Om het maximale uit de Xbox te halen moet je het werk opdelen in verschillende packages en verdelen over de drie cores. In tegenstelling tot normale multithreading op .NET gebruiken we een eigen coördinator. Als er bijvoorbeeld bij een explosie verschillende brokstukken uit elkaar moeten spatten, definiëren we dit als een functie. Die wordt op de lijst met acties gezet en zodra een thread klaar is met een voorgaande actie pakt deze van de lijst met acties de eerste volgende nieuwe opdracht en gaat die uitvoeren. Dit blijkt uitstekend te werken voor de game en is bovendien schaalbaar. Op een platform met meer cores hoeft je niet te veranderen en krijg je automatisch meer performance. Op een pc is deze aanpak niet echt nodig, maar het werkt prima of je de game nu draait op een pc met slechts één of met vier cores. Rocket Riot werkt in ieder geval heel dynamisch.”