Pling

Team: Nexicat
Opdrachtgever: Stichting Sterrenkind
Opdracht: Breid de functionaliteit van de Sterrewereld uit met een spel voor chronisch zieke kinderen van 7 tot 19 jaar

Oplossing: Pling

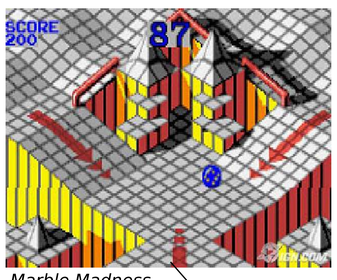
Pling is een competatief online spel voor meerdere spelers, los gebaseerd op het knikkerspel. Door het behendig bewegen van het *batje*, vinden spelers de snelste route die de meeste punten oplevert.

Doelen

Multiplayer & Interactie
Aantrekkelijk
Uitdagend
En natuurlijk leuk!

Het Idee

Het idee voor het spel is ontstaan toen we aan het brainstormen waren over welke spelletjes we vroeger op school speelden. Al snel was daar knikkeren en werd het spel een combinatie van Marble Madness, Mario en Pong!



Marble Madness

Mario

Pong

Pling in actie

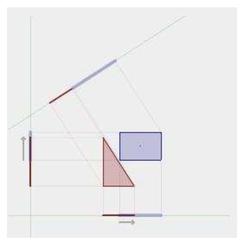
Aanpak

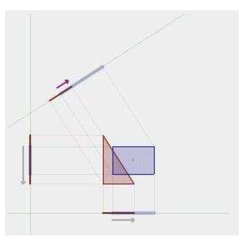
Het maken van een spel houdt in dat je niet kan steunen op de traditionele manieren van software maken die je tijdens Software Engineering leert. Een spel is een creatief programma dat moet groeien buiten de grenzen van het orginele ontwerp. Het ontwikkelmodel hierbij is dan ook een evolutionair model, wat er in feite op neerkomt dat we steeds nieuwere en betere versies. Aangezien een spel nooit 'af' is, wordt er gestopt wanneer de tijd op is.

Pling in actie

Collision Detection

Een van de grootste uitdagingen in bijna alle spelletjes, is het detecteren van aanraking tussen verschillende objecten in het spel. In het algemeen zijn er twee manieren om dit te doen: Detecteren van overlap: nadat object objecten hebben bewogen en het berekenen van aanraking voordat het gebeurt. Om snelheidsredenen is in Pling het eerste principe toegepast.
Er zijn hiervoor legio principes, waarvan het "Separating Axis Theorem" een van de meest universele is. In het kort, als er geen as te vinden is waarop twee (convexe) objecten niet overlappen, dan overlappen de twee objecten.

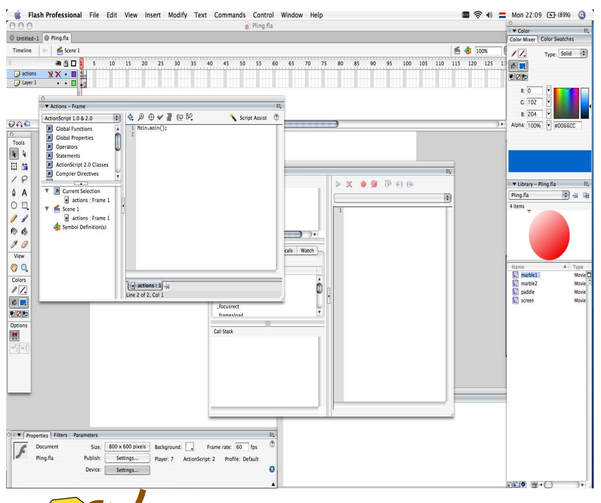




Hoewel dit principe goed bruikbaar zou zijn, is het niet het meest snelle algoritme (snelheid vs genericiteit). Daarom is er in Flade (zie kader over libraries) gekozen om een ander principe toe te passen wat specifieker is voor de vormen die gebruikt worden.

Flash 8

http://www.macromedia.com



$699

Vim + Mtasc

http://www.vim.org
http://www.mtasc.org

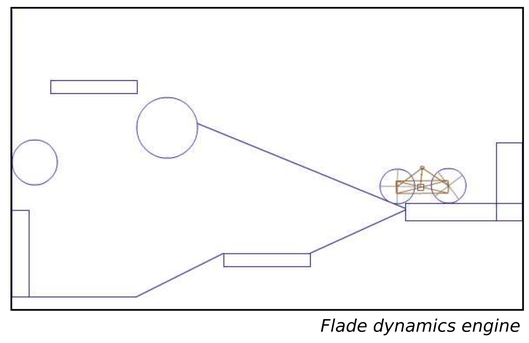
$0

Keuze voor Mtasc

Flash 8 is het officiele programma van Macromedia om Flash applicaties in te bouwen. Tijdens het programmeren kwamen we er echter achter dat de ontwikkelomgeving traag en belemmerend was. Wat we eigenlijk wilden was onze favoriete editor (Vim) in een UNIX omgeving met een command line compiler die flash applicaties genereert. Mtasc is een open source project die precies dat leverde waar we naar zochten.

Software Engineering vs Flash : 0 - 1

Het grootste probleem wat bij Pling naar voren kwam was het lopende conflict tussen nette software engineering principes en de ActionScript en Flash omgeving. Hoewel ActionScript als taal in principe voldoende in huis heeft om netjes in te kunnen ontwikkelen, blijkt dit in de praktijk toch tegen te vallen.
ActionScript is bijvoorbeeld niet type-safe en mist zoiets simpels als null pointer exceptions of segfaults (wat ineens erg nuttige tools blijken te zijn...)
Verder is ActionScript sinds versie 2.0 volledige object georienteerd, maar de Flash API is dat niet. Met name op het gebied van event handling zijn er lelijke hacks nodig om te zorgen dat events bij het goede object aankomen.
Tot slot is de ingebouwde compiler en debugger niet berekend op nette ontwerpen met veel kleine klassen: Het compilen van Pling duurt enkele minuten in de Flash compiler. Het compilen met behulp van MTASC is aanzienlijk sneller, maar ontbeert een debugger. Debuggen via debug output en backtraces wordt echter aanzienlijk bemoeilijkt door het eerstgenoemde punt.


Flade dynamics engine

Libraries & Flash

Om het heruitvinden van het wiel te voorkomen en tijd te besparen is het uiteraard de moeite waard om libraries te gebruiken die in de applicatie passen. Na zorgvuldig afwegen zijn er in Pling 2 gebruikt: de Flash Dynamics Engine (Flade) en EnFlash, een interface framework. Beide libraries zijn volledige object georienteerd en passen goed binnen onze software engineering principes. Hoewel deze libraries een grote hulp zijn in het programmeren van het spel, is het duidelijk dat deze (en eigenlijk alle Flash libraries) nog onvolwassen en weinig generiek zijn. Een aantal gewenste functies zijn niet aanwezig en moeten er bij geprogrammeerd worden.