

Richtlijnen voor de finale inzending van het practicum Computer Graphics

Bart Sas
Daniel van den Akker

15 mei 2013

Dit document bevat de richtlijnen voor de finale inzending van het practicum van het vak Computer Graphics. Daarnaast worden er ook nog enkele fouten besproken die we tijdens het jaar meerdere keren zijn tegengekomen en een negatieve invloed op je punten kunnen hebben hoewel ze eenvoudig kunnen worden vermeden.

Richtlijnen voor een geldige inzending

In deze sectie worden de vereisten besproken waaraan je inzending moet voldoen. Inzendingen die niet aan deze vereisten voldoen kunnen worden beschouwd als niet ingeleverd.

Je inzending moet ten laatste op *27 mei 2013 om 23:59u* via *Blackboard* ingediend worden. Na dit tijdstip verdwijnt de ‘dropbox’ op Blackboard en wordt je inzending niet meer aanvaard. Lever *één* Engine in die alle functionaliteit ondersteunt en niet een aparte Engine voor iedere opgave. Je kan je Grafische Engine meerdere keren inleveren. Als je na het inleveren van je Engine dus nog fouten ontdekt dan kun je, zolang de ‘dropbox’ beschikbaar is, alsnog een verbeterde versie insturen. **Let op:** wij kijken *enkel* de versie na die het laatst werd ingestuurd!

Je inzending moet bestaan uit *één* enkel *.tar.gz* bestand. Andere formaten zoals *.zip* of *.rar* worden *niet* aanvaard. Dit *.tar.gz* bestand moet als naam **sXXXXXXX.tar.gz** (waarbij XXXXXXX de laatste 7 cijfers van je rolnummer zijn) hebben en onder andere aan de volgende vereisten voldoen:

- Het *.tar.gz* bestand moet *alle* source bestanden bevatten die je Engine nodig heeft om te werken. Dit betekent dat je niet alleen de code van je Engine zélf maar ook alle overige code die standaard niet in de standaard C++ library zit (zoals bijvoorbeeld *ini_configuration*) in je *.tar.gz* bestand moet meeleveren.
- Voeg geen niet-compilerende code aan je *.tar.gz* bestand toe. Je kan beter een inzending insturen waarvan de code wel compileert maar die niet alle functionaliteit implementeert dan een inzending die niet compileert. Je code moet compileren met g++ 4.6 onder Linux. Als referentie-platform worden de linux-distributie en g++ compiler gebruikt die in de PC klassen is voorzien.
- Het *.tar.gz* bestand mag geen afbeeldingen of gecompileerde code bevatten. Zorg er dus voor dat er geen *.o*, *.bmp*, *.png*, ... bestanden of executables in de archive aanwezig zijn. De enige uitzondering hierop zijn textures die gebruikt worden voor texture mapping.
- Op Blackboard hebben we voor de meeste opgaves een aantal voorbeeld *.ini* bestanden beschikbaar gemaakt waarmee je je Engine kon testen. Zorg ervoor dat deze bestanden *niet* in het *.tar.gz* bestand aanwezig zijn.

- Voeg voor de opdrachten waarvoor geen .ini bestandsformaat werd opgegeven en voor eventuele extra functionaliteit wél voldoende .ini bestanden toe zodat we kunnen zien dat de verschillende aspecten hiervan goed werken; je moet uiteraard niet overdrijven. Je geeft de .ini bestanden best ook een naam die beschrijft wat het .ini bestand demonstreert, bijvoorbeeld: `stochastisch_lsysteem.ini` of `schaduw_met_puntbron.ini`.
- Als je extra functionaliteit hebt geïmplementeerd geef je best in een **README** kort weer welke extra functionaliteit je hebt voorzien, welke .ini bestanden deze functionaliteit laten zien en indien nodig een woordje uitleg over hoe deze functionaliteit kan worden aangesproken.

Je .tar.gz bestand moet ook één Makefile bevatten waarmee we je Engine eenvoudig kunnen compileren. Dit bestand moet aan de volgende voorwaarden voldoen:

- De Makefile moet in de root van het .tar.gz bestand staan.
- Met het `make all` commando moet je engine te compileren zijn. De gegenereerde executable moet `engine` heten en moet in dezelfde map gegenereerd worden als de Makefile.
- De Makefile moet gebruik maken van de `$(CXX)` omgevingsvariabelen. Hardcode het pad of naam van de compiler dus niet in de Makefile aangezien deze kan verschillen van platform tot platform.
- Je kunt je hiervoor baseren op de Makefile die je kan vinden op Blackboard.

Je Grafische Engine moet aangesproken kunnen worden via de interface van `engine.cc`. Concreet betekent dit dat de functie `generate_image` geïmplementeerd moet zijn. Hierbij moeten de volgende regels in acht genomen worden:

- Wanneer je engine wordt opgeroepen met een .ini bestand als parameter moet er een .bmp bestand worden gegenereerd met dezelfde naam.
- Wanneer een bepaalde image niet gegenereerd kan worden omdat de benodigde functionaliteit niet geïmplementeerd is moet er geen afbeelding gegenereerd worden. Je kan dit doen door `generate_image` een lege afbeelding te laten returnen. Dit doe je d.m.v. `return img::Image();`

Veel voorkomende fouten

Tijdens het jaar zijn we meerdere inzendingen tegengekomen slechts gedeeltelijk of zelfs helemaal niet werkten door kleine foutjes die eenvoudig vermeden hadden kunnen worden. Vermits wij je Engine enkel kunnen evalueren op basis van wat er werkt en niet op basis van ‘wat er zou moeten werken’ raden we iedereen *ten sterkste* aan om te controleren of je inzending in orde is alvorens je de opdracht indient op Blackboard. Om je hierbij te helpen worden hieronder een aantal belangrijke ‘valkuilen’ opgesomd:

- Zorg ervoor dat je Engine niet ‘overijverig’ is. Je Engine moet, bij de verplichte opdrachten, *enkel* doen wat er volgens de opgave verwacht wordt en niets meer. Als het type van een .ini bestand gelijk is aan ‘ZBuffering’ dan mogen belichting of texture mapping bijvoorbeeld niet worden toegepast.
- Controleer dat je .tar.gz bestand in orde is vóór je deze instuurt. Dit betekent dat je .tar.gz alle bestanden moet bevatten die nodig zijn om de code te compileren en dat je Makefile ook effectief op basis van deze bestanden een werkende executable kan genereren. Tijdens het jaar werden er vele engines ingestuurd die niet konden worden gecompileerd door foutjes in de Makefile. Kopieer, om te controleren dat dit niet het geval is, het .tar.gz bestand naar een *lege* directory, pak hem uit, compileer de code en run je Engine.

- Je inzending wordt geëvalueerd met de g++ compiler op een Linux systeem. Als je je Engine hebt ontwikkeld op een ander besturingssysteem of met een andere compiler, controleer dan of je Engine ook werkt met de g++ compiler op Linux.