Beschrijving behaalde requirements c++ eindopdracht

Teun Leenders 2171232

|  |  |
| --- | --- |
| **Behaalde requirement** | **Locatie / beschrijving** |
| De code gebruikt een consistente en gedocumenteerde stijl in layout en naamgeving. | Alle files in de “source” map. |
| De code gebruikt geen overbodige #defines en bevat geen magische constanten. | Alle files in de “source” map. |
| De code bevat geen leaks van memory of andere resources. | Zie destructors in verschillende klassen en het einde van de “main” functie. |
| De code gebruikt op ten minste 1 plek op een zinnige wijze overerving en virtuele functie(s).  Dit wordt aangetoond met een UML diagram van de betreffende klassen. | Zie afbeelding op volgende pagina en “3dGraphicsEindopdracht.vpp”. Overerving vindt plaats door abstracte klasse WorldObject. Deze worden gerealiseerd door “Object3d en ObjectLight. |
| De code bevat een herbruikbaar onderdeel, dat gedocumenteerd is met Doxygen en getest met Catch2 of een andere vorm van geautomatiseerde unit-test. (Dit hoeft dus niet voor alle code.) | Zie map “StringUtilTest”, dit is een Visual Studio testproject waarin de klasse “StringUtil” wordt getest. Deze staat in de map “3dg\_Worldbuilder/source”.  De documentatie hiervan staat in de map “doxs/html/ class\_string\_util.html”. |
| De code gebruikt waar dit mogelijk en zinnig is STL containers en algoritmen. | Voorbeeld: in het bestand “source/WorldFactory.cpp” wordt op regel 53 de functie std::getLine() gebruikt om één regel uit een bestand te lezen. Deze wordt in een std::string gezet en vervolgens wordt deze opgedeeld in een std::vector. |
| De applicatie leest gegevens in vanuit een tekst file naar in-memory data, volgens het factory pattern. | Zie het bestand “source/WorldFactory.cpp”. |
| De documentatie bevat een overzicht (tabel) waarin staat aan welke criteria het code voldoet, en (voor zover zinnig) specifiek welke delen van de code (files en waar nodig regelnumers). | Zie dit bestand. |
| De code gebruikt een zelfgeschreven STL-stijl container of algorithme. | Zie de functie StringUtil::IndexOf() in het bestand “source/StringUtil.cpp” |
| De code gebruikt waar mogelijk en zinnig const, constexpr en override. | In de functie World::FindObjectById(const int& id) (in “source/World.cpp” op regel 100) wordt een const reference naar id meegegeven, om af te dwingen dat id niet wordt aangepast door deze functie.  In het bestand “source/Object3d.h” op regel 20 en 21 worden de keywords override gebruikt om aan te tonen dat deze methode hun abstracte versie vanuit WorldObject overschrijft. |

