

Bestudeer de video van de volgende les geheel:

Overerving en veelvormigheid

Experimenteer zelfstandig met de daarin behandelde voorbeeld-programma's. Een link naar de broncode daarvan staat onder de video. In de volgende les kun je hierover vragen stellen.

Opgaven bij de huidige les:

In te leveren uiterlijk 2 dagen vóór de volgende les.

Het programma heeft een voorgeschreven opbouw. Verbeteringen hierop zijn toegestaan in overleg met de docent. Begin je bron-bestand weer met requirements, testspecs en design.

Declareer de volgende klassen:

class Vec3D: Deze class neemt de plaats in van de struct uit de vorige huiswerkopgave. De functies uit die opgave, behalve *main*, zijn nu methods van deze class. De functie die eerst *vec3D* heette, wordt nu de constructor en gaat dus *Vec3D* heten.

class Ray: Deze class stelt een "lichtstraal" voor. Hij heeft twee fields van het type *Vec3D*, namelijk de steunvector *support* en de richtingvector *direction*. Daarnaast heeft hij een constructor met de volgende signature:

```
Ray (float xSup, float ySup, float zSup, float xDir, float yDir, float zDir)
```

De parameters van de constructor zijn de componenten van respectievelijk *support* en *direction*.

class Sphere: Deze class stelt een bol voor. Hij heeft fields *Vec3D center* en *float radius*. De constructor van deze class heeft de signature:

```
Sphere (float x, float y, float z, float radius)
```

Daarnaast heeft hij een method die de afstand van z'n *center* tot een bepaalde *Ray* berekent, met signature:

```
float distFromRay (Ray const &ray) const
```

Ook heeft hij een method die bepaalt of een ray de bol raakt, met het prototype:

```
bool hit (Ray const &ray) const
```

Tenslotte heeft hij een method die bepaald waar een bepaalde ray de bol raakt, met het prototype:

```
Vec3D hitPoint (Ray const &ray) const
```

Naast deze classes met hun methods is er de main functie, met het gebruikelijke prototype. In deze main functie worden een `stl::vector` gemaakt met daarin volgende bollen:

```
Sphere (-0.4, 0.23, -1, 0.4), Sphere (0.4, 0.4, -1.2, 0.3) en  
Sphere (0.7, -0.15, -1.5, 0.2)
```

Ook worden een `stl::vector` gemaakt met daarin de volgende "lichtstralen":

```
Ray (-far, 0.23, -1, far, 0, 0), Ray (0.4, -far, -1.2, 0, far, 0) en Ray (0.7,  
-0.15, -far, 0, 0, far)
```

Vervolgens worden alle punten waar een straal een bol raakt afgedrukt in een console window. Elke straal weerkaatst niet, hij eindigt als hij voor het eerst een bol raakt.

Deze opgave is opnieuw een stap richting de eindopgave. Referenties naar de wiskunde die nodig is voor de methods *distFromRay*, *hit* en *hitPoint* staan in het document *les_inhoud.pdf* van deze les. Hoe het zit met steunvectoren en richtingvectoren wordt behandeld in de video uit de les *Vergelijkingen, matrices en vectoren* van de module *Toegepaste wiskunde* en ook in het vak Lineaire Algebra. Je kunt de video echter zelfstandig bestuderen.

De beoordeling van deze huiswerkopgave maakt deel uit van je eindcijfer. Wat beoordeeld wordt is niet alleen een al dan niet werkend programma, maar vooral of je zowel in grote lijnen als in details begrepen hebt wat je aan het doen bent, inclusief de wiskunde, en de gemaakte keuzen kunt motiveren.

Programma en verstandig gedoseerd commentaar weer in het engels. Zet verwijzingen naar gebruikte on-line informatie-bronnen ook als commentaar in je programma. Dit is een goede gewoonte, zo worden je collega's geen spoorzoekers.

--/--