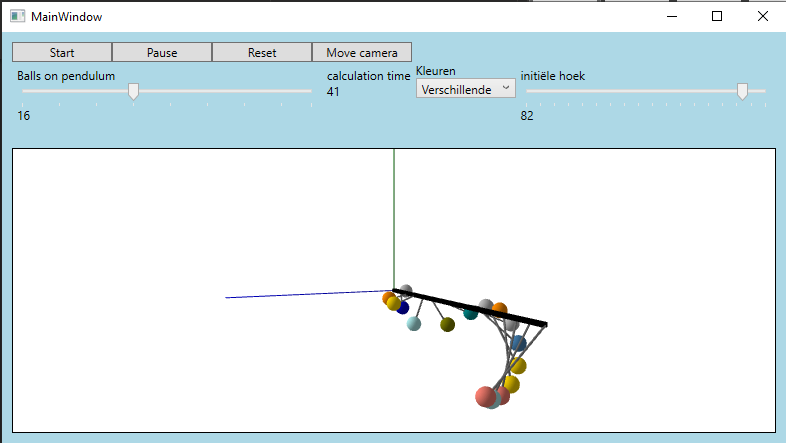
Pendulum Wave Pieter De Block

# Korte inleiding

De tweede opdracht van aplied programming was het maken van een Pendulum Wave in WPF vorm. Met verschillende instelbare parameters.



# GUI

De initiële hoek geeft de startpositie van de bollen aan ik raad aan om deze op > 45° te zetten. De knop “Move camera” is helaas niet in gebruik.

# Methoden van MainViewModel.cs

**InitDispatchTimer()**

Deze methode maakt een instantie aan van een dispatcher timer die 60 keer per seconde mijn de help methode oproept.

**Help**()

Deze asyncrone methode gaat er voor zorgen dat ik nog aanpassingen kan doen via mijn gui als er berekeningen en rendering bezig is.

**dispatcherTimer\_Tick()**

Deze methode berekend voor elke slinger die aanwezig is de hoek van de slinger. Daarna update ik de slingers en de bollen.

**InitSupport()**

Maakt een maakt een bar afhankelelijk van de hoeveelheid slingers die er aanwezig zijn.

**CreateSlingerArray()**

Maakt een array van instanties van de classe Slinger aan. Daarna vult het programma elke slinger aan met de juiste variabelen per Slinger. Nadien initialiseert het programma de support, slingers en ballen.

**InitialiseSlinger()**

Is een methode die maar 1 keer zal gebruikt worden wegens vertraging. Hier worden de beams gecreëerd , dit word niet gedaan in de Update methodes omdat er al beams aangemaakt zijn die gewoon moeten transformed worden naar de juiste locatie. In deze methoden worden dus de beams aangemaakt en onmiddellijk op de juiste hoek ten opzichte van de oorsprong geplaatst. Deze worden toegevoegd aan een Model3D group.

**InitialseBalls()**

Idem initilise slinger

**GetColour()**

De gebruiker heeft de keuze om 1 kleur of meerdere kleuren te hebben tijdens het runnen van het programma. Als de gebruiker kiest voor verschillende kleuren op de ballen dan word er een random kleur gekozen van de \_colorBrushList deze lijst heeft wel maar 14 verschillende kleuren in de aanbieding.

**UpdateSlinger & UpdateBollen**

Hierbij worden geen nieuwe ballen of staven aangemaakt. Hijbij worden deze terug gezet op het referentie punt en juist getransformeerd om daarna terug op de juiste plaats gezet te worden.