

1. INHOUD

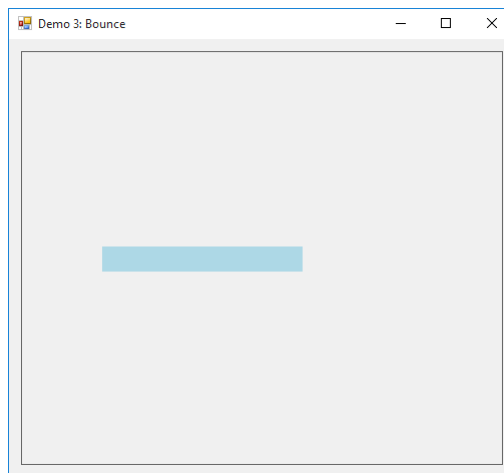
Timers, switch cases, combobox

2. OEFENINGEN

- Demo 1: Bounce
- Demo 2: Hex to decimal converter
- Demo 3: Debug oplossing demo 1
- A: Count to 10
- A: Biljarttafel
- A: To reverse or not to reverse
- E: Random values
- E: Kogelwerpen (2.0)
- E: Klok

2.1 Demo 1: Bounce

Laat een horizontale rechthoek van 25 pixels hoog bij 200 pixels lang horizontaal bewegen. Wanneer de rechthoek de wand raakt van de canvas, kaatst deze terug. De rechthoek beweegt dan in de tegenovergestelde richting voort.



Figuur 1: Voorbeeld demo 3

2.2 Demo 2: Hex to decimal converter

Schrijf een programma waarin het mogelijk is om hexadecimaal geformateerde getallen om te zetten naar gehele getallen. Gebruik een switch case om de mogelijke hexadecimale getallen om te zetten naar de juiste getallen (benader hierbij de input als een string). Merk op dat elk hexadecimaal getal een waarde tussen 0 en 15 (0 tot f) kan bevatten. In het eerste gedeelte van de opgave mag ervan uitgegaan worden dat de gebruiker een correct getal ingeeft. In het 2^{de} deel kunnen ook foutieve inputs ingegeven worden. Foutieve inputs moeten ook correct afgehandeld kunnen worden.

2.3 Demo 3: Debug oplossingen demo's 1 en 2

Ook hier wordt de debugger gebruikt om het verloop van het programma in de voorgaande demo's te illustreren.

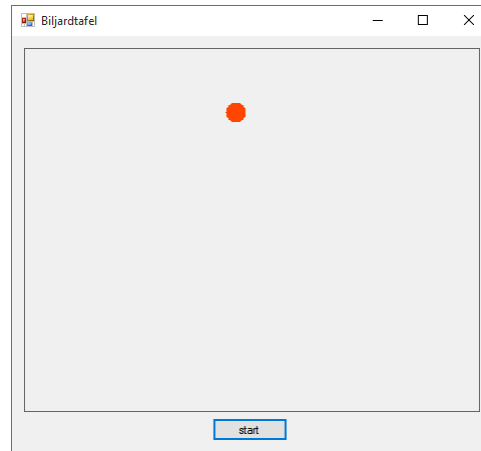
2.4 A: Count to 10

Schrijf een programma die na 10 seconden een messagebox met een boodschap weergeeft. Geef het aftellen weer op het formulier in een label. Gebruik hierbij een timer die je afstelt om eenmaal per seconde een event te genereren.

Hint: Gebruik als teller een variabele die je globaal declareert.

2.5 A: Biljarttafel

Schrijf een programma die een bal op een veld laat rollen. Zorg ervoor dat deze bal bij het botsen tegen een van de wanden terugkaatst. Stel de wanden van het veld gelijk aan de randen van de canvas waarop je tekent. De beweging van deze bal wordt voorgesteld door een X- en Y-component. Samen vormen deze componenten een snelheidsvector. Dankzij een timer kan de beweging van de bal gevisualiseerd worden (zet deze aan, met een interval tussen 10 ms en 250 ms). De X- en Y- beweging worden aangepast in functie van de timerfrequentie, zodat deze bal een al dan niet vloeiende beweging uitvoert. De terugkaatsing kan plaatsvinden door één element van de snelheidsvector van teken te doen wisselen. Als de bal tegen een verticale wand aanbotst, wordt de horizontale richting omgedraaid. In het het geval dat de bal tegen een horizontale wand aanbotst, wordt de verticale richting omgedraaid (tekenwissel). Een voorbeeld van deze applicatie kan teruggevonden worden in figuur 2.



Figuur 2: Voorbeeld biljardtafel

2.6 A: To reverse or not to reverse

Schrijf een programma waarin je een string gaat manipuleren. Dit programma voorziet 3 opties:

- Print de string in dezelfde volgorde af in een label.
- Print de string in omgekeerde volgorde af (in dezelfde label).
- Verander alle karakters in de string door random karakters en print daarna de array af in dezelfde label.

Om de keuze tussen de 3 opties te kunnen maken wordt er op de het formulier een combobox (naam = cmbBox) gezet. Om de opties in de combobox te plaatsen kan men onderstaand stukje code gebruiken.

Codefragment 1: Combobox vullen met 3 elementen

```
1 cmbBox.Items.Add("Print in volgorde af");  
2 cmbBox.Items.Add("Print in omgekeerde volgorde af");  
3 cmbBox.Items.Add("Print randomwaarden af");
```

Deze code wordt bij het opstarten van het programma uitgevoerd. Die code wordt dus in de functie “public MainWindow()” geplaatst. Bovenstaande code wordt dus omgebouwd tot onderstaand codefragment.

Codefragment 2: Combobox vullen met 3 elementen

```
1 public MainWindow()  
2 {  
3     InitializeComponent();  
4     // add additional code here to launch at startup of our application  
5     cmbBox.Items.Add("Print in volgorde af");  
6     cmbBox.Items.Add("Print in omgekeerde volgorde af");  
7     cmbBox.Items.Add("Print randomwaarden af");  
8 }
```

Om te weten welk element uit de combobox geselecteerd is geweest, wordt de eigenschap “SelectedIndex” opgevraagd. Deze eigenschap is een getal tussen -1 en $n - 1$ met n het aantal opties. Indien men -1 terugkrijgt, is geen enkele optie geselecteerd. In de andere gevallen is een optie aangeduid. Dit kan gebruikt worden binnen een switch case.

Codefragment 3: Geselecteerde index van een combobox opvragen

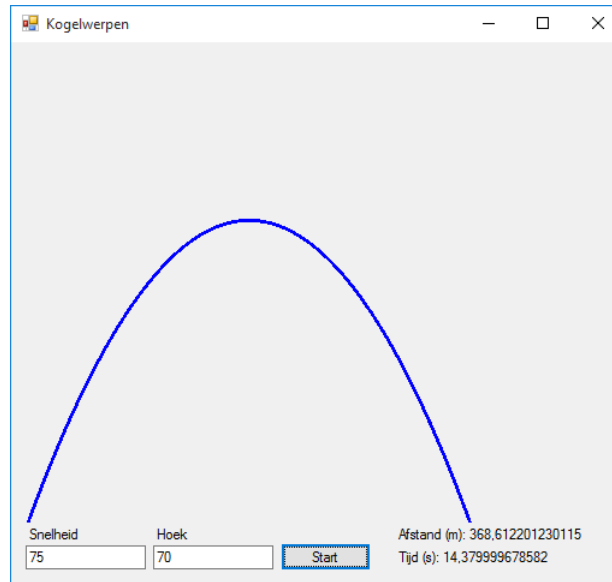
```
1  switch(cmbBox.SelectedIndex)
2  {
3      case 0:
4          //-----
5          break;
6      case 1:
7          //-----
8          break;
9      case 2:
10         //-----
11         break;
12     default: // (-1 and others)
13         //-----
14 }
```

2.7 E: Kogelwerpen (2.0)

Herneem de vorige opgave van het kogelwerpen. Hierbij wordt de evolutie van de beweging visueel weergegeven a.d.h.v. een timer. De kromme die de massa's volgen wordt beschreven als:

$$\begin{cases} x(t) = v \cdot \cos(\alpha) \cdot t \\ y(t) = v \cdot \sin(\alpha) \cdot t - \frac{1}{2} \cdot g \cdot t^2 \end{cases} \quad (1)$$

Plot de baan die deze kogel aflegt. De beginsnelheid en de afschiethoek worden ingegeven worden door de gebruiker. Let hierbij op dat het mogelijk is dat de plot niet op het scherm past.



Figuur 3: Voorbeeld kogelwerpen

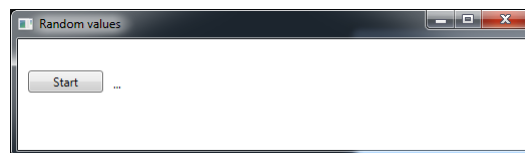
2.8 E: Random values

Schrijf een programma dat op regelmatige tijdstippen een nieuw getal genereert en het aan een label op het formulier toevoegt. Dit label moet ook van doorzichtigheid veranderen en het is pas wanneer de doorzichtigheid op 0 staat dat er een nieuwe waarde toegevoegd mag worden. De doorzichtigheid verandert als volgt:

- als de doorzichtigheid op 0 (of lager) staat, gaat de doorzichtigheid toenemen totdat die 1 bereikt (double),
- als de doorzichtigheid op 1 (of groter) staat, gaat die terug naar 0 aftellen.

Gebruik een variabele die je op 1 of -1 instelt, naargelang dat de doorzichtigheid toe- of afneemt. Gebruik een waarde die kleiner is dan 0.1 om de increment of decrement te realiseren. Hieronder worden een aantal afbeeldingen weergegeven van het resultaat.

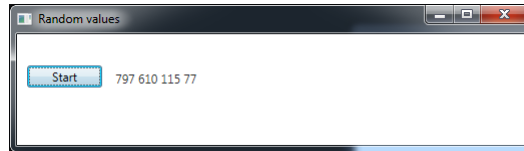
Hint: Gebruik de “Opacity” eigenschap van de label om de doorzichtigheid (tussen 0 en 1) aan te passen.



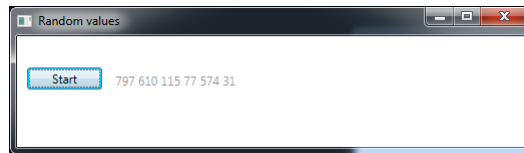
Figuur 4: Start programma.



Figuur 5: Random values na de 1^{ste} iteratie.



Figuur 6: Random values na de 5^{de} iteratie.



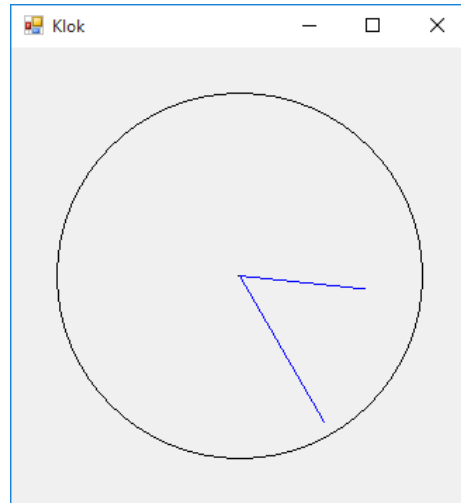
Figuur 7: Random values na 7^{de} iteratie. Merk de doorzichtigheid van de label op.

2.9 E: Klok

Programmeer een analoge klok waarin enkel de seconden en minuten weergegeven worden. Gebruik een timer die je om de seconde laat afgaan (wijzers bewegen). Gebruik een canvas dat je op 300 bij 300 pixels instelt. Het middelpunt van de klok plaats je in het midden van de canvas. De wijzers worden weergegeven a.d.h.v. lijnen die van dat middelpunt vertrekken. De wijzers hebben een lengte van 120 pixels.

Hint: De eindpunten van de wijzers t.o.v. het middelpunt M kunnen gevonden worden via formule 2.

$$P = \begin{cases} P_x = M_x + 120 \cdot \cos(2 \cdot \pi \cdot \frac{t}{60}) \\ P_y = M_y + 120 \cdot \sin(2 \cdot \pi \cdot \frac{t}{60}) \end{cases} \quad (2)$$



Figuur 8: Voorbeeld klok