

Kwaliteit

Gebrek aan kwaliteit betekent **minder punten**.

De code dient goed leesbaar en onderhoudbaar te zijn en voldoet aan de geziene programmeer regels.

Het project moet builden zonder errors en zonder warnings.

Gebruik functies waar het passend is.

Er mag geen code aanwezig zijn die niets met deze opgave te maken heeft (geen copy/paste van andere oefeningen).

Namen van variabelen, functies, enumerations, structs,.. moeten betekenisvol zijn en volgen de afgesproken naamgeving regels.

Vermijd niet-performante code, bijvoorbeeld het steeds opnieuw creëren van textures in elk frame is uit den boze.

Verkiez lokale variabelen boven globale indien mogelijk.

Het is niet toegelaten C++ features te gebruiken die nog niet werden behandeld in de lessen. Bijvoorbeeld: arrays, classes,...

Algemeen

De opgave bestaat uit 2 onderdelen:

1. Een console oefening.
2. Een grafische oefening.

Beide maak je in dezelfde framework solution.

Technische vereisten

Het is belangrijk dat je de opgave uitvoert zoals gevraagd zodat we zien of je bepaalde programmeer vaardigheden verworven hebt. Werkt je oplossing, maar heb je het op een andere manier opgelost, dan krijg je minder punten. Dit betreft bijvoorbeeld: De creatie en het gebruik van de beschreven functies, structs, enumerations,...

Gegeven

- **Een zip file met** een werkend voorbeeld.

Creëer het project

1. Creëer een visual studio project met framework, dus gebruik makend van de bestanden in 00_General/FrameworkFunctions_V0.2.3.zip. En beschreven in 00_General/HowToCreateProject.pdf. De naam van het project is **Prog1DAExx_Qyy_Familienaam_Voornaam**, waarbij je **xx** vervangt door je groepsnummer, **Familienaam_Voornaam** door jouw naam en voornaam, en yy door het Q examennummer zoals het in de header van dit document terug te vinden is.

2. Plaats je volledige naam en groep in de **titel van het window**.

Console

1. Introductie

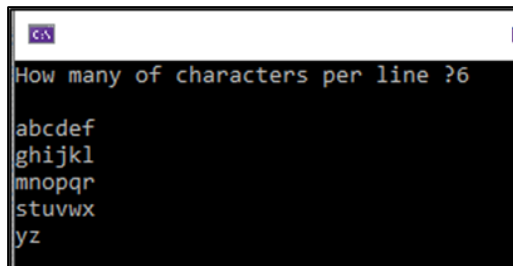
Om in de console vanuit het game.cpp-bestand te kunnen afdrukken, moet je dit opnemen: `#include <iostream>`.

2. Assignment

Maak een PrintLetters functie. De functie wordt aangeroepen vanuit de Start-functie in Game.cpp.

In de functie is er code die resulteert in dit:

- Vraag de gebruiker hoeveel tekens per regel de gebruiker zou willen zien.
- Print alle letters van het alfabet (kleine letters) in het gevraagde aantal tekens per regel.
- Gebruik geen `std::string` voor deze opdracht.
- Druk geen hard gecodeerde tekst af op de console, b.v. `std::cout << "abcdef\n";`
- Gebruik een of meer iteratie lussen.



```
How many of characters per line ?6
abcdef
ghijkl
mnopqr
stuvwx
yz
```

Dit is hoe de console er zou moeten uit zien.

Circle

Beschrijving van het gedrag:

Bij het klikken op het venster begint een willekeurig gekleurde cirkel te groeien, gecentreerd op de aangeklikte positie, van 0 tot een willekeurige maximale straal. Die maximale straal kan elk heel getal in het bereik [50,150] zijn, inclusief 50 en 150. Wanneer de maximale straal is bereikt, krimpt hij tot hij verdwijnt.

Als de cirkel de vensterrand overlapt terwijl het groeit en voordat de maximale radius is bereikt, wordt hij rood en wordt "Ouch!" op de console afgedrukt. Hij begint ook onmiddellijk te krimpen tot hij verdwijnt. Met andere woorden, de cirkel gaat nooit buiten het venster.

Daarna gebeurt er niets meer totdat er weer in het venster wordt geklikt en alles weer opnieuw begint.

Wees voorzichtig dat je geen ellips tekent/vult met een negatieve radius, want dit resulteert in een oneindige lus in de fill ellips functie.

Klikken met de muis terwijl de cirkel zichtbaar is heeft geen effect.

Elke nieuwe cirkel heeft een nieuwe willekeurige kleur en een nieuwe maximale straal.

De groei- en krimpsnelheid van de straal kan worden aangepast met de pijltjestoetsen in het inclusieve interval $[0.2, 5]$, in stappen van 0.4 inclusief de 0.2 en 5. Elke nieuwe snelheid wordt in de console afgedrukt.

Uitvoering:

De cirkel heeft drie states: growing, shrinking, and waiting. (groeien, krimpen, en wachten).

Maak eerst een enumeratie klasse en variabele om deze toestanden weer te geven. Als je niet met enumeraties kunt werken, gebruik dan een alternatief, maar dan verlies je punten.

In de functie OnMouseUpEvent:

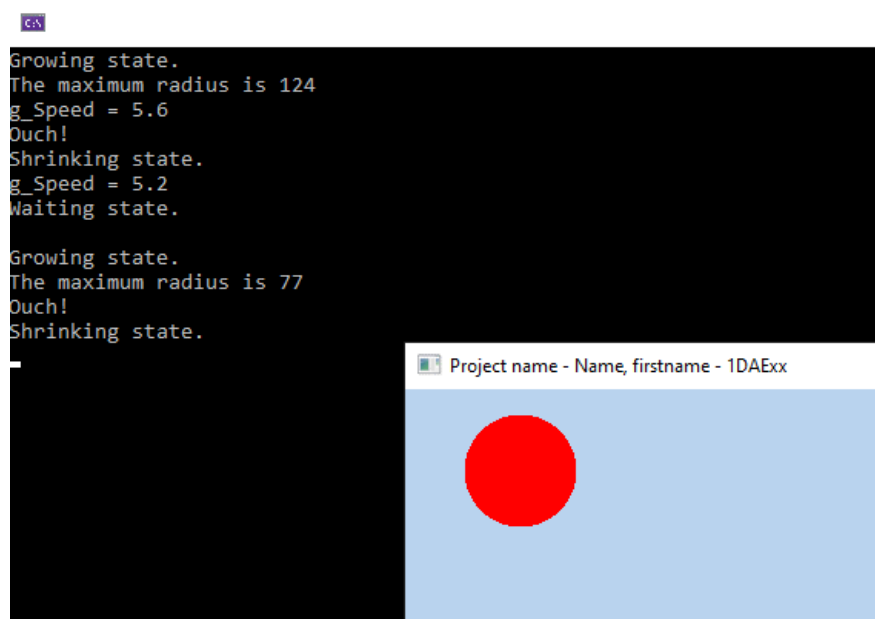
- Als de linkermuisknop in het venster wordt losgelaten terwijl er geen cirkel zichtbaar is, verandert de state in growing en wordt een nieuwe max radius en kleurwaarde gegenereerd. Muisknoppen indrukken terwijl er de cirkel zichtbaar is heeft geen effect. De nieuwe positie wordt opgeslagen. De maximale radiuswaarde en de nieuwe state worden op de console afgedrukt.
- e.x en e.y bevatten de Windows coördinaten van de muis. Vergeet niet de y-coördinaat om te draaien.
-

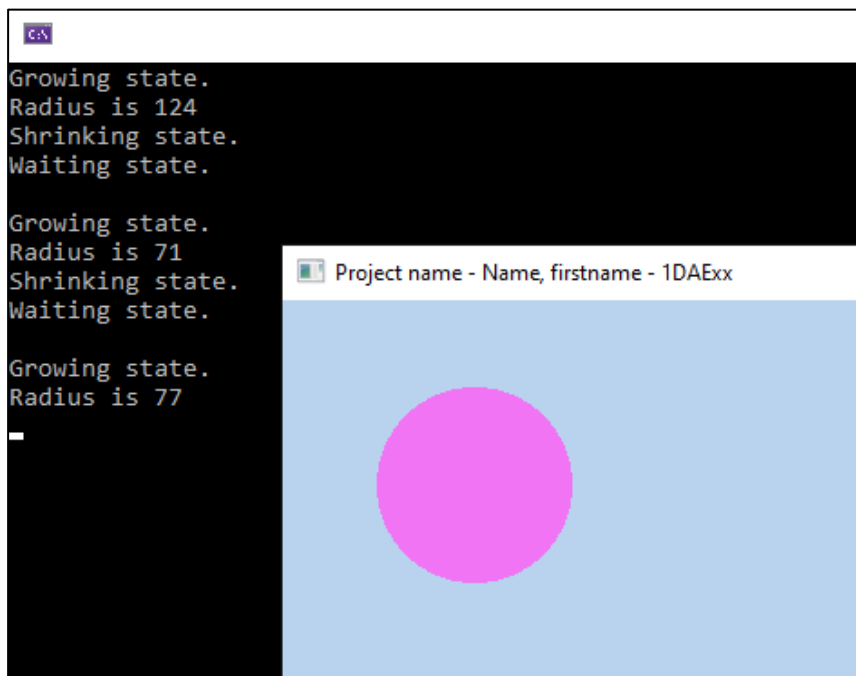
In de functie Update:

- Controleer de state en verander de radiuswaarde dienovereenkomstig.
- Controleer op botsing met de venstergrens en de maximale radiuswaarde en verwerk de state/kleur dienovereenkomstig. Als de cirkel de vensterrand overlapt, wordt "Ouch!" op de console afgedrukt.

In de functie OnKeyUpEvent:

- Controleer de op en neer pijltjestoetsen en verander de snelheid zoals hierboven beschreven.
- Telkens wanneer de toestand verandert, wordt een bericht met de nieuwe toestand naar de console afgedrukt, zoals te zien is op de schermafbeelding.





Indienen

Clean de solution. Sluit Visual Studio. Verwijder de .vs en de debug en/of de x64 folder.

Converteer de oplossingsmap naar een rar/zip bestand met dezelfde naam als de map. Controleer of het zip-bestand nog steeds alle benodigde codebestanden bevat.

Dien deze rar / zip file in onder het leho assignment van dit labo examen.

Zie **zelf** na of dit correct gebeurd is. Verkeerd ingediende examens – bijvoorbeeld een ander project, een project met ontbrekende code files – krijgen 0 punten.