

Kwaliteit

Gebrek aan kwaliteit betekent **minder punten**.

De code dient goed leesbaar en onderhoudbaar te zijn en voldoet aan de geziene programmeer regels. Het is aan jou om de klassen volledig volgens deze regels te maken zoals: const correctness, rule of 5, initialization, naamgeving, formatting, releasing resources,...

Het project moet builden zonder errors en zonder warnings zowel in x86 als x64.

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 regels.

Vermijd niet-performante code, bijvoorbeeld het onnodig steeds opnieuw creëren van objecten in elk frame is uit den boze. Vermijd globale variabelen. Verkijs lokale variabelen boven data members.

Algemeen

Je moet een puzzelspel maken. De puzzel bestaat uit 4 tegels waarmee 1 van 6 dieren kan worden samengesteld. Elke tegel toont een bepaald deel van de dieren en het is mogelijk om door de dieren van een tegel te bladeren. Wanneer alle tegels delen van hetzelfde dier tonen, is de puzzel opgelost. Probeer het gegeven werkend voorbeeld.

Technische vereisten

Het is belangrijk dat je de opgave uitvoert zoals gevraagd zodat we kunnen nagaan 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:

- Definitie van de Tile class,
- Objecten van deze Tile class creëren
- De gecreëerde objecten of hun adressen in een array bijhouden en ze via deze array manipuleren
- Respecteren van de rule of 5
- ...

Gegeven

- De folder **Resources** met daarin de nodige afbeeldingen.
- De folder **CodeFiles** met de files Tile.h en Tile.cpp. De Tile klasse beschrijft een tegel van de puzzel. Alleen de public interface wordt gegeven, jij beslist over het private gedeelte. Je bepaalt ook zelf of een **destructor** nodig is. Deze public interface staat in commentaar, haal de methoden stap voor stap uit commentaar zoals verder beschreven in dit document. Je bepaalt ook zelf welke methoden **const** zijn, dit is niet aangeduid in de gegeven declaraties.
- **EX50.exe** een werkend voorbeeld.

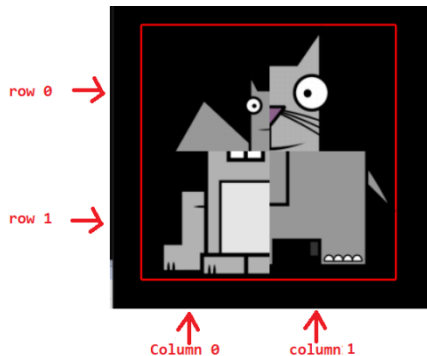
Creëer het project

1. Creëer een project met naam **Prog1DAExxFamilienaamVoornaamEx50**, waarbij je **xx** vervangt door je groepsnummer en FamilienaamVoornaam door jouw naam en voornaam.
2. Doe het nodige zodat je jouw **09_Framework** files gebruikt.

3. Kopieer en voeg de Tile class files **Tile.h** en **Tile.cpp** toe.
4. Kopieer de gegeven **Resources** folder in de folder van dit project.
5. Pas the **titel van het window** aan, het bevat : Ex50, jouw naam, voornaam en groep.

Opgave

1. Creëer en teken 4 Tile objects



Elk Tile-object toont een specifiek deel van de dieren. De Resources folder bevat de afbeeldingen **TilesRC.png** waarbij: **R** de rij en **C** de kolom van de tegel aangeeft, zoals aangeduid in de afbeelding links.

Een voorbeeld: De afbeelding Tiles00.png bevat het linkerboven gedeelte van de 6 dieren.

Tile class

Definieer de Constructor, de parameters zijn:

- Het rechthoekige gebied van het window waarin het Tile-object zijn deel van een dier moet tekenen.
- Het path van de afbeelding dat overeenkomt met dit Tile-object.
- Het aantal dieren dat deze afbeelding bevat.

De constructor kiest willekeurig welke van de dieren dit Tile object zal tekenen. We zullen dit het **current animal** in de verdere beschrijving noemen.

Definieer de Draw method. Het Tile object tekent zijn deel van het current animal in de toegewezen rechthoekige gebied van het window.

Game class

- Creëer 4 Tile-objecten en wijs aan elk Tile-object een rechthoekig gebied met een breedte en hoogte van 128 pixels toe. De tiles zijn gerangschikt in een 2x2 rooster. Bewaar de Tile objecten of hun adressen - je bepaalt dit zelf - in een fixed size array.
- Draw de Tile objecten.

2. Een activated Tile object tekent extra informatie

Wanneer de muis zich in het rechthoekige gebied van een Tile-object bevindt, wordt dit Tile-object geactiveerd, zoniet is het gedeactiveerd.

In het geval van een geactiveerd Tile-object worden er een aantal extra's getekend:

- Boven de afbeelding van het dier gedeelte, wordt een semi-transparante grijsachtige rechthoek getekend.
- En daarboven 2 rode rechthoekige hitregions, één linksonder en één rechtsonder. De breedte en hoogte van een hitregion is $1/4^{\text{de}}$ van de breedte / hoogte van het rechthoekige gebied van het Tile-object.

Tile class

Definieer de CheckActivation method, wanneer de gegeven positie zich binnen het rechthoekige gebied van het Tile-object bevindt, wordt het Tile-object geactiveerd. Wanneer het zich daarbuiten bevindt, wordt het gedeactiveerd.

Breid de **Draw** method uit met de extra tekeningen in geval het Tile object geactiveerd is.

Game class

Voer de nodige stappen uit zodat een Tile object wordt geactiveerd / gedeactiveerd wanneer de muis zich binnen / buiten zijn gebied bevindt.

3. Het vorige/volgende dier tonen bij het klikken op een hitregion

Tile class

Definieer de CheckHit method. Wanneer de gegeven positie zich binnen de linker/rechter hitregion van het Tile-object bevindt, wordt het vorige/volgende dier het current animal en wordt *true* gereturned. Dit is een circulaire operatie:

- Wanneer het eerste dier het *current animal* is, wordt bij het klikken op de linker hitregion het laatste dier het *current animal*.
- Wanneer het laatste dier het *current animal* is, wordt klikken op het rechter hitregion het eerste dier het *current animal*.

Wanneer de gegeven positie niet binnen die hitregions valt, wordt *false* gereturned.

Game class

Voer de nodige stappen uit om de speler door de dieren onderdelen te laten bladeren als de linkermuisknop ingedrukt wordt boven één van de hitregions.

4. Duid aan wanneer de puzzel opgelost is

Tile class

Definieer de GetCurrentAnimal method die het nummer (vb: 0 to 5) van het current animal teruggeeft.

Definieer de Deactivate method, die het Tile object deactiveert.

Game class

Voer de nodige stappen uit om te weten of een puzzel is opgelost, dit is wanneer alle tiles een deel van hetzelfde dier tonen.

Teken een rand rond de puzzel. De rand is **rood** als de puzzel **niet is opgelost** en **groen** als deze wel is **opgelost**.

Teken ook de **naam van het dier** wanneer de puzzel opgelost is. De namen staan in de afbeelding **Names.png**. De volgorde van de namen (van boven naar beneden) is hetzelfde als in de TilesRC.png-afbeeldingen (van links naar rechts).

Schrijf performante code, controleer niet in elk frame of de puzzel opgelost is, maar alleen als het echt nodig is.

Wanneer de puzzel opgelost is, zijn de tiles niet langer geactiveerd en kunnen ze ook niet meer geactiveerd worden, dus de hitregions en grijsachtige rechthoek worden niet langer getekend en het klikken op een tile mag ook niet langer een effect hebben.

5. Schud de tiles als de r-key losgelaten wordt

Tile class

Definieer de Randomize method die random een *current animal* kiest.

Game class

Wanneer de r-toets omhoog gaat, call je deze Randomize functie van de Tile-objecten.

Zorg ervoor dat de puzzel hierna niet toevallig in opgeloste toestand komt. Dus de tiles mogen niet toevallig hetzelfde dier tonen na deze operatie.

Hierna kan men opnieuw starten met het oplossen van de puzzel door op de tiles te klikken.

6. Bij de start mag de puzzel nog niet opgelost zijn

Nadat de Tile-objecten gecreëerd zijn, is het mogelijk dat ze toevallig allen hetzelfde dier tonen. Vermijd dit door - op dezelfde manier als in de vorige stap - na de creatie de tiles te schudden.

Indienen