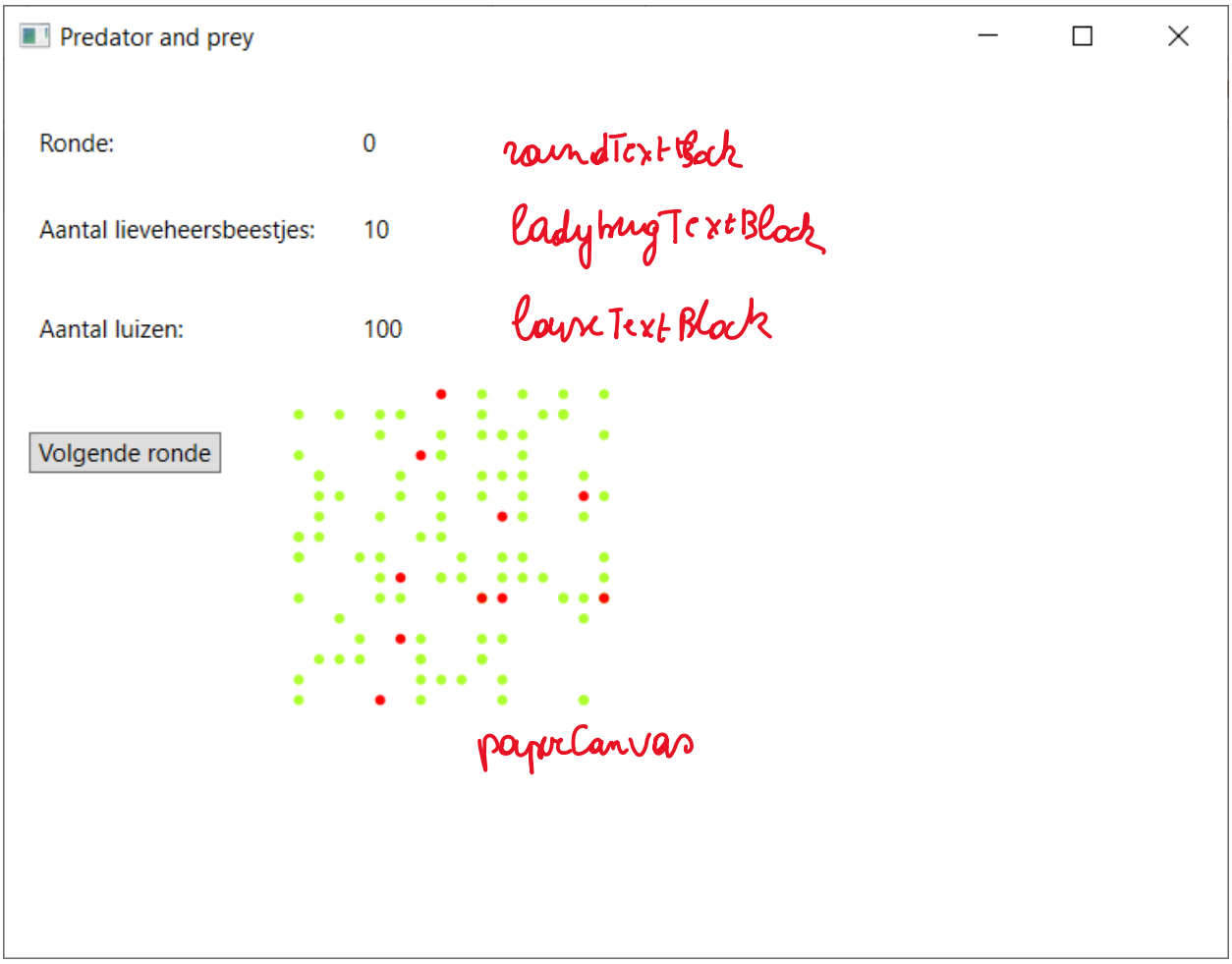
Oefening hoofdstuk 22 Interfaces

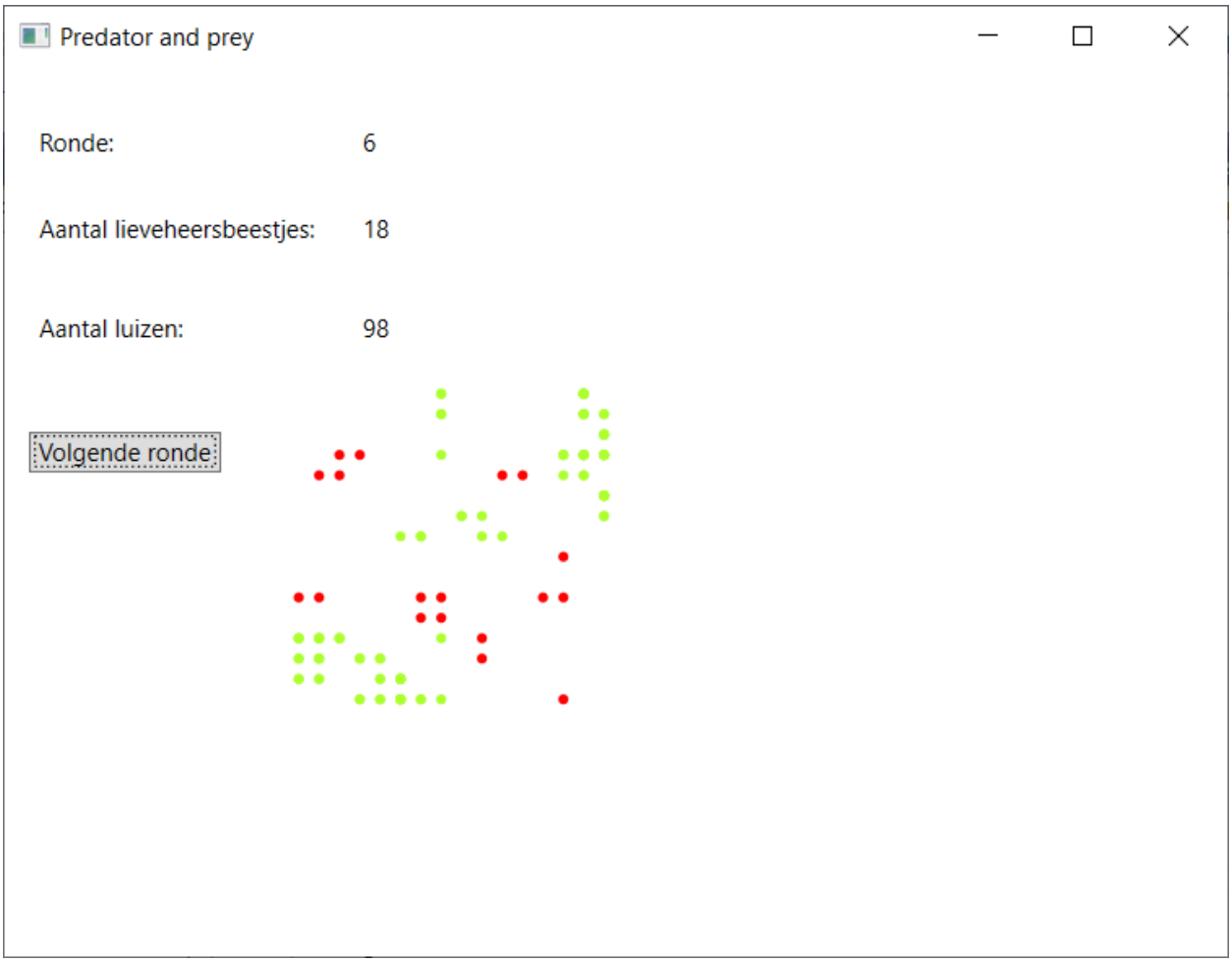
Prey Predator

Prey predator

We maken een simulatie van een dierenwereld.

In deze simulatie hebben we een wereld vol met luizen en lieveheersbeestjes (Luizen zijn het lievelingseten van lieveheersbeestjes). De simulatie start met een aantal lieveheersbeestjes en luizen. Bij elke klik op de knop gaat er een ronde voorbij. Elke ronde bewegen de dieren (willekeurig) rond en proberen zich voort te planten. Roofdieren (lieveheersbeestjes) proberen tevens prooi (luizen) te eten.





We willen de code zo abstract en uitbreidbaar mogelijk maken. Daarom zullen we met interfaces en abstracte klassen werken.

Interfaces

We gaan de dieren in de dierenwereld weergeven op een Canvas.

Daarom maken we als eerste een interface **IDisplayable**. Instanties van klassen die IDisplayable implementeren zullen weergegeven kunnen worden op een Canvas.

Maak een folder "Contracts" aan waarin we alle interfaces zullen plaatsen. Voeg een interface IDisplayable toe in de folder. project RMK new Folder

De interface voorziet de volgende methoden:

- void DisplayOn(Canvas canvas)
    - Plaatst het object op het canvas.
- void StopDisplaying()

- Verwijdert het object van het canvas.

- void UpdateDisplay()
  - Past de Margin van het object aan.

Als tweede voorzien we een interface die een dier in de wereld voorstelt. Stop een interface **IAnimal** in de "Contracts" folder. Een IAnimal heeft dezelfde methoden als een IDisplayable. Voorzie bovendien de volgende zaken in deze interface:

IAnimal : IDisplayable

- Een read-only property "**Position** Position".
  - **Position** is een klasse die je zelf maakt. Deze klasse stelt de x- en y- coördinaat voor waar het dier zich bevindt in zijn leefwereld (een matrix van 16 bij 16 posities). Deze klasse heeft volgende members:
    - Property "X" (int). Mag enkel waarden tussen 0 en 15 (incl.) hebben.
      ↳ als waarde < 0 → 0 als waarde > 15 → 15
    - Property "Y" (int). Mag enkel waarden tussen 0 en 15 (incl.) hebben.
    - Methoden "MoveUp", "MoveDown", "MoveLeft", "MoveRight" die de x- of y-coördinaat met 1 verhogen of verlagen.
    - voorzie een constructor met als parameters x en y coörd
- Een property "bool IsDead".
- Een methode "void Move()"
  - Deze methode zorgt ervoor dat het dier 1 stap in een willekeurige richting beweegt.
- Een methode "IAnimal TryBreed()"
  - Deze methode zorgt ervoor dat het dier zich probeert voort te planten. Als het lukt dan wordt het nieuwgeboren dier teruggegeven. Als het niet lukt, wordt null teruggegeven.

Als derde maken we, in de "Contracts" folder een interface **IPredator** die een roofdier voorstelt. Een IPredator heeft dezelfde properties en methoden als een IAnimal. Bovendien heeft een IPredator nog twee extra methoden:

IPredator : IAnimal

- bool CanEat(IAnimal animal)
  - Geeft aan of het roofdier het dier dat meegegeven wordt als parameter, kan opeten.
- void Hunt(IList<IAnimal> possibleVictims)
  - Laat het roofdier jagen op zijn mogelijke slachtoffers

De laatste interface die we toevoegen aan de "Contracts" folder is **IAnimalWorld**. Voorzie de volgende zaken in deze interface:

- Een read-only property “IList<IAnimal> AllAnimals”
    - Geeft een lijst van alle dieren in de wereld terug.
- Een read-only property “int CurrentRoundNumber”
    - Geeft aan in welke ronde de wereld zich bevindt.
- Een methode “void AddAnimal(IAnimal animal)”
    - Plaatst een nieuw dier op een willekeurige positie in de wereld.
- Een methode “void ProcessRound()”
    - Laat een ronde voorbijgaan in de wereld waarbij de dieren bewegen, zich proberen voort te planten en te jagen.

Abstracte klasse

Voeg een abstracte klasse **Animal** toe aan het project.

Deze klasse implementeert de IAnimal interface. Sommige interface methoden laten we nog abstract maar voor de zaken die voor alle dieren hetzelfde zijn kunnen we een concrete implementatie voorzien.

Voorzie een implementatie voor DisplayOn, UpdateDisplay, StopDisplaying, Position, IsDead en Move.

De TryBreed methode houden we abstract.

↳ instantie eigenschap - canvas

Tips:

- Gebruik een static field van het type Random zodat steeds dezelfde instantie van Random gebruikt wordt:
    - private static Random _randomGenerator = new Random();
- Constructoren
    - Voorzie 2 constructoren:
        - De eerste constructor heeft 2 parameters. Via deze parameters worden de maximumleeftijd (elke ronde wordt het dier 1 ouder) en de kleur(Color) van het dier meegegeven. Deze constructor geeft het dier een willekeurige positie op de 16x16 wereld.

          → _maxAge
        - De tweede constructor heeft dezelfde 2 parameters als de andere constructor plus een derde “Position” parameter. Deze constructor geeft het dier een specifieke positie.
    - Door in de ene constructor de andere constructor aan te roepen kan je duplicatie in de code vermijden.

constructor in abstracte klasse → access modifier niet op protected (niet op public!)

- Het dier wordt getekend op een canvas in de vorm van een ellips. De ellips wordt getekend aan 1 van de punten in de 16x16 matrix van de wereld. Tussen elk punt van de wereld zitten 10 pixels. De ellipsen hebben een diameter van 5 pixels. De kleur van de ellips krijg je binnen via de constructor.
- Move methode
    - Als een dier beweegt wordt het ouder. Houd de leeftijd van het dier bij in een field waar ook overgeërfde klassen aan kunnen.
    - Als het dier ouder wordt dan een maximumleeftijd (die je via de constructor binnen krijgt), dan sterft het.
    - Verander de Position door willekeurig een move methode van de Position klasse aan te roepen (MoveLeft, MoveUp, …)
    - Als het dier bewogen heeft, dan moet het ook hertekend worden op het canvas.

Concrete classes

Louse

In de klasse **Louse** maak je een concrete implementatie van Animal.

- Gebruik de kleur “GreenYellow”.
- De maximumleeftijd van een luis is 6 ronden.
- Een luis kan zich voortplanten om de 2 ronden. Om de 2 ronden wordt er een nieuwe luis op de positie van de ouder geplaatst. In deze wereld hoef je niet met 2 te zijn om je te kunnen voortplanten. Een pasgeboren luis (leeftijd = 0) kan zich nog niet voortplanten.

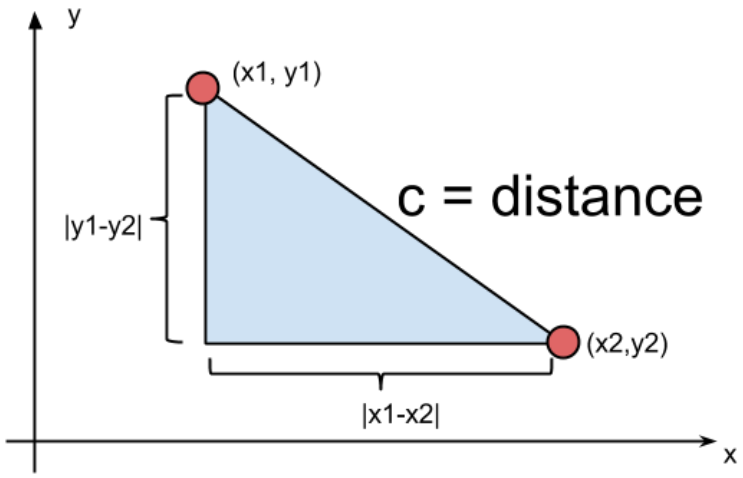
LadyBug

In de klasse **LadyBug** maak je een concrete implementatie van Animal en implementeer je de IPredator interface.

- Gebruik de kleur “Red”.
- De maximumleeftijd van een lieveheersbeestje is 16 ronden.
- Een lieveheersbeestje kan zich voortplanten om de 4 ronden. Na 4 ronden wordt er een nieuw lieveheersbeestje op de positie van de ouder geplaatst. Een pasgeboren lieveheersbeestje (leeftijd = 0) kan zich nog niet voortplanten.
- Een lieveheersbeestje moet jagen om te overleven. Elke ronde gaat het lieveheersbeestje op zoek naar luizen die 3 of minder posities ver zijn. Zijn er luizen in de buurt zijn, dan kan het lieveheersbeestje eten. De luizen die opgegeten worden, sterven uiteraard. Als een lieveheersbeestje 3 ronden lang geen eten vindt, dan sterft het.

- Om de afstand tussen 2 dieren te vinden kan je volgende formule gebruiken:

$$\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$$



AnimalWorld

De klasse AnimalWorld is een implementatie van de IAnimalWorld interface.

Tips:

- Geef een canvas mee als parameter aan de constructor.
- Houd intern (private field) een instantie van IList<IAnimal> bij die alle dieren van de wereld bevat.
- ProcessRound
  - Itereer over alle dieren. Laat elk dier bewegen, zich proberen voort te planten en op jacht te gaan (als het een roofdier is).
  - Dieren die sterven moeten verdwijnen uit de lijst en van het canvas.
  - Dieren die geboren worden moeten toegevoegd worden aan de lijst en op het canvas.
  - Als je over een lijst itereert met een for(each) lus dan mag je geen items toevoegen of verwijderen uit deze lijst terwijl je nog in de lus zit. Hou daarom nog 2 lijstjes bij. Een lijst met nieuwgeboren dieren en een lijst met dieren die gestorven zijn. Nadat de lus is uitgevoerd, kan je de nodige dieren toevoegen en verwijderen.

MainWindow

Bouw een interface zoals in de screenshots bovenaan in deze opgave.

- Een knop. Elke klik op de knop simuleert het voorbijgaan van een ronde.
- Een canvas waarin de dieren weergegeven kunnen worden.
- Enkele labels met wat statistieken (deze worden elke ronde bijgewerkt):
    - Nummer van de ronde
    - Totaal aantal lieveheersbeestjes in de wereld
    - Totaal aantal luizen in de wereld

In de codebehind doe je het volgende:

- Maak bij het uitvoeren van de constructor een instantie aan van AnimalWorld en ken de instantie toe aan een private field "_insectWorld" van het type IAnimalWorld.
    - Voeg 100 luizen toe aan de wereld
    - Voeg 10 lieveheersbeestjes toe aan de wereld
- Voer de methode "ProcessRound" van de AnimalWorld uit bij elke klik op de knop
- Voorzie een private method "DisplayStatistics" die aan de hand van de IAnimalworld instantie de labels met de statistieken gaat updaten.
    - Ook hier kan je met de "is" operator werken om na te gaan hoeveel dieren er van een bepaald type zijn.