

Københavns Universitet
PoP Assignment 3

Victor Vangkilde Jørgensen - kft410

December 16, 2024

Contents

1	Question 1 - Reflections on group work and sources	3
1.a	Who are the members of your group that participated in the assignment.	3
1.b	Compare how your group worked on assignment 3 with the way you worked on assignment 2. Describe similarities and differences.	3
1.c	Give a list of the external sources you used during the assignment and how you used them.	3
2	Question 2 - Simulation of Moths	4
2.a	Describe how your solution relies on the object-oriented programming paradigm.	4
2.b	Describe the type Vec and explain how your solution makes use it. . . .	4
2.c	Using your code, detail what is happening in a simulator step.	5

1 Question 1 - Reflections on group work and sources

1.a Who are the members of your group that participated in the assignment.

Udover mig selv, har jeg udarbejdet opgaven i gruppe med:

- Daniel Friis-Hasché - rcb933
- Aksel Mannstaedt Rasmussen - qfl561

1.b Compare how your group worked on assignment 3 with the way you worked on assignment 2. Describe similarities and differences.

Resten af min gruppe kom senere i gang med afleveringen, da de syntes, at afleveringen var meget svær, derudover havde vi andre afleveringer for i samme tidsramme som denne, vi havde derfor, alle sammen arbejdet på forskellige tidspunkter af afleveringsperioden. Dette var ikke et problem som sådan, andet end at vi alle var på forskellige steder af afleveringen, da vi skulle tale om den. Vi talte om hvordan vi hver havde gjort brug af DIKU-Canvas, men udover det, havde vi bare talt om den basale opsetning af afleveringen. Afleveringens struktur gjorde det i højere grad svære at kommunikere om specifikke opgaver, da selve afleveringen ikke var opdelt i opgaver.

1.c Give a list of the external sources you used during the assignment and how you used them.

Jeg har benyttet Claude 3.5 Sonnet til at genere koden til at beregne næste position:

```
1      member this.nextPos(lightOn: bool, lightPos: Vec option) =
2          let mothSpeed = 5.0
3
4          let targetHeading =
5              match lightOn, lightPos with
6              | true, Some lp -> ang (sub lp this.pos)
7              | _ -> this.heading + GetRandomRange -0.174 0.174 // random
3              heading of -10 to 10 degrees in radians
8
9          this.heading <- targetHeading
10
11         let velocity = (mothSpeed * cos this.heading, mothSpeed * sin
12             this.heading)
13
14         let mutable (x, y) = this.pos
15         this.pos <- add (cyclic 0.0 w x, cyclic 0.0 h y) velocity
```

<https://claude.ai> | Dato: 03/12/2024

Jeg har også benyttet Claude 3.5 Sonnet til at genere noget af koden i denne del:

```
1      let react (state: Moth list * bool) (ev: Event) : (Moth list * bool
2          ) option =
3      let (mothList, lightOn) = state
4      match ev with
5      | Key ' ' -> Some (mothList, not lightOn)
6      | TimerTick ->
7          reactAllMoths mothList lightOn
8          Some (mothList, lightOn)
```

<https://claude.ai> | Dato: 03/12/2024

Alt den generede kode er derefter blevet redigeret og verificeret af mig, og resten af afleveringen er skrevet af mig selv i samarbejde med min gruppe.

2 Question 2 - Simulation of Moths

You are to make a simulator of 5 moths that fly randomly on a cyclic domain until the light is turned on in the middle of the image, at which time the moths will move toward the light.

The solution must include a class with the following signature:

```
type Moth =
new: pos: Vec * hdng: float -> Moth
member heading: float
member pos: Vec
member draw: unit -> PrimitiveTree
```

where `Vec` is a `float * float` pair denoting the moth's position, and `hdng` is its initial direction in radians. The `draw` function produces a DIKU-Canvas `PrimitiveTree` that represents the moth object at position `pos`. The type `Vec` is defined in the `asteroid` library presented at the lecture in Week 10, which contains helpful functionality for vector algebra and other things.

The moths never stop flying. The space, in which the moths fly, is to be cyclic meaning that when the moth flies out of the window on the right-hand side, then it reenters on the left and similarly for the other sides of the window. The light is to be turned on and off by pressing space. In each simulator step, a moth moves a small constant distance in the direction of its heading. When the light is on, a moth's heading is the direction of the light, and otherwise, the heading is updated in each step by adding a random number in the range of `[-10,10]` degrees to it.

2.a Describe how your solution relies on the object-oriented programming paradigm.

Ifølge min forståelse af OOP, har jeg benyttet classes, som `Moth` og `Light`, der er i stand til, at have bestemte funktioner og members dannet til kun det objekt. Members gør det muligt, at kalde på bestemte funktioner, som ellers ikke bliver kørt.

Disse objekter bevares, og opdateres når vi animerer, ved brug af DIKU-Canvas. Eksempelvis kalder jeg på en member `"nextPos,"` som beregner næste position hver frame.

2.b Describe the type `Vec` and explain how your solution makes use it.

Jeg åbner `Vectors` fra `asteroids` namespace't i `asteroids`-folderen, ved at bruge `"open Asteroids.Vectors."`

Herfra kan vi nu bruge type'en `Vec` og andre funktioner fra `Vectors` modulet. Jeg bruger selvfølgelig `Vec` som type af `float * float`, men derudover også `"add"` og `"cyclic"` til at lave skærm-grænserne.

2.c Using your code, detail what is happening in a simulator step.

Der sker rigtig meget på bare et step, men her er en overordnet liste af ting der sker:

- Hvis der trykkes på space, tændes lyset hvis det ikke er tændt, ellers slukkes lyset.
- Hvis lyset er tændt, beregner moth'en en ny retning (*targetHeading*), der peger mod lyset.
- Hvis lyset er slukket, tilføjes en tilfældig "drejning" til moth'ens retning for at simulere tilfældig bevægelse.
- Ud fra retningen beregnes en ny position baseret på moth'ens hastighed.
- Positionen opdateres, og moth'en bevæger sig på en måde, der sikrer, at den bliver inden for skærmens grænser (ved hjælp af *cyclic*).
- Hver moth tegnes ved at lave en cirkel omkring dens nuværende position
- Hvis lyset er tændt, tegnes en stor gul cirkel i midten af skærmen.
- Alle moth'nes tegninger kombineres, og lyset tilføjes, hvis det er tændt.
- Den samlede tegning (alle moths og lyset) opdagteres og vises på skærmen.

Alt dette sker $\frac{60}{1000}$ gange pr. sekundt, og da der efter standard går 1000 ticks for 1 sekundt, svarer dette til 60 gentagelser pr. sekundt.