

# 101 Belysning av en klot

Jakub Tomasz Drewniak

November 2024

# 1 Inledning

Projektet är en visualiseringssapplikation två huvudsakliga funktioner, att visualisera belysning på ett klot, samt spara visualiseringen antigen som en text- eller bildfil. Användaren kommer ha möjligheten att anropa programmet från terminalgränssnittet och med olika argument kommer den välja vilken visualiseringssmetod programmet ska använda.

Projektet använder sig i huvudsak av klasser vilket innebär en annorlunda sätt att ta sig an lösningarna. Då programmet ska fungera mestadels med hjälp av klasser behöver man definiera funktioner i klasserna och få dem olika klasserna att interagera med varandra beroende på visualiseringssalternativet. En annan utmaning i detta projekt är filhanderingen för bildfiler. Detta kommer att kräva hitta den mest optimala lösningen för att skapa antigen en png- eller jpg-fil.

# 2 Användarscenarion

## Ritning

Lisa tycker att sena nätter och tidiga morgonar är dem bästa möjligheterna att uttrycka sin estetiska vilja. Att skugga är inget lätt och den första stege är dem svåraste. Programmet ska vara till hjälp till Lisa som vill ha en formbar mall att utgå ifrån. Efter programmet har startats kommer hon få ett meny i kommandoraden:

Alternativ för visualisering:

- 1: Textritning
- 2: Grafisk ritning
- 3: Avsluta programmet

Lisa matar in "2" och ett fönster öppnas med en klot som har en vit outline, nu behöver hon endast trycka på det tomma klotet som nu visas på skärmen, och där hon har tryckt kommer ljuskällan falla in och måla in klotet med olika belysningar. Då skärmen är inte alltid den bästa att läsa av färg, går det också att spara resultatet som en png-fil för att sedan skrivas ut på papper.

## Skrivare

Olof har precis skapat en skrivare som fungerar som en skrivmaskin. Han har testat den och tycker att den fungerar utmärkt för text men problem förekommer då han försöker infoga bilder i sina artiklar. Olof startar programmet från Ubuntu terminalen genom att skriva:

```
$ python3 klot.py -t
```

Sedan väljer han textvisualiseringssalternativet genom att mata in "1". Sedan anger han klotets radie samt x- och y-koordinater för att placera ljuskällan. Han väljer först en punkt som inte ligger på klotet och programmet ber honom att ange annolunda värden efter felmeddelandet: "Ljuskällan ligger inte på klotet". Efter justeringen, ut kommer en färdig bild av klotet i form av ASCII art. Han bara kopiera denna för att infoga i sin fil till sin artikel om skuggningstekniker. Han har även sparat den som en textfil för senare användning.

### 3 Kodskelett

```
class Klot:  
    """Representerar klotet med en angiven radie"""  
    def __init__(self, radie):  
        """Konstuerar ett klot med en angiven radie"""  
  
    def beräkna_z(self, x, y):  
        """Beräknar värdet av z med givna x- och y-koordinater på klotet.  
        Returnerar Exception i fall punkterna inte befinner sig på klotet.  
        Returnerar värdet z som en float"""  
  
    def in_klot(self, x, y):  
        """Kontrollerar att en given punkt finns i klotet"""  
  
  
class Scen:  
    """Representerar scen som innehåller två objekt, i det här fallet klotet  
    och ljuskällan"""  
  
    def __init__(self, radie, x_0, y_0):  
        """Konstruerar scenen och returnerar Exception i fall ljuskällan inte  
        ligger på klotet"""  
  
    def belysning_styrka(self, x, y, z):  
        """Räknar ut belysningspunkter för angivna koordinater. Tar in z för  
        att beräkna belysningsstyrkan.  
        Returnerar b som en float. b representerar belysningsstyrkan."""  
  
  
class TextVisualisering:  
    """Implementation av visualisering som använder terminalgränssnittet.  
    Sparar filer som textdokument."""  
    def __init__(self, scen):  
        """Konsturerar textvisualiseringen av scenen"""  
  
    def belysning_char(self, b):  
        """Räknar ut tecknet som ska skrivas ut med hjälp av värdet b som  
        representerar belysningsstyrkan. Skriver sedan ut olika tecken  
        beroende på bs värde."""  
  
    def scen_som_sträng(self):  
        """Returnerar textrepresentationen av klotet som en lista av linjer"""  
  
    def utskrivning(self):  
        """Skriver ut scenen i terminalen"""
```

```

def spara_som_text(self):
    """Sparar textrepresentationen av klotet som en textfil"""

class TkInterVisualisering:
    """Implementation av visualiseringen som använder tkinter som
visualiseringslösning. Sparar den som en png-fil """
    canvas_size = int
    radie = int

    def __init__(self):
        """Konstuerar visualiseringssobjekt med angiven scen. Använder värdet
        canvas_size för att initialisera tkinter canvas."""

    def färg(self, b ):
        """Räknar ut färgen för angiven belysningsstryrka använder värdet b
        för att måla in pixlar på klotet med olika nyanser för att
        representera ljuset"""

    def teckna(self, event):
        """Skapar ett klot och målar in den med hjälp av tkinter. Tar
        ljuskällan från en click-event."""

    def spara_som_bild(self):
        """Sparar TkInter canvas som en png fil"""

class App:
    """Representerar fungerande programmet med grafiskt eller text
visualisation"""
    def __init__(self):
        """Skapar klassen för programmet med angiven visualiseringsteknik"""
    def run():
        """Metoden som startar programmet. För textmetoden hämtar den input
        från användaren för att skapa ett klot."""

```

## 4 Minnesmodellen

Beroende på användarens valda alternativ kommer programmet fungera lite annorlunda. Efter textvisualiseringen har anropats skapar den ett objekt av klassen Klot. Detta objekt används sedan för att skapa ett Scen-objekt. Scen-objektet kommer användas av Klassen TextVisualisering och på detta objektet kommer programmet att skapa en ASCII-art versionen av ett klot.

Om grafiska visualiseringen väljs kommer den först skapa ett objekt av klassen Klot. Till skillnad från textvisualiseringen kommer den ta andra steg innan den skapar ett Scen-objekt. Det skapas

respektive, ett Tk-objekt, en Label och Button. Efter dessa har skapats används Klot-objektet för att skapa en scen och denna används sedan av Tk objektet för att ge en utskrivning i ett fönster. I båda fallen används App som en huvudklass för att köra programmet.