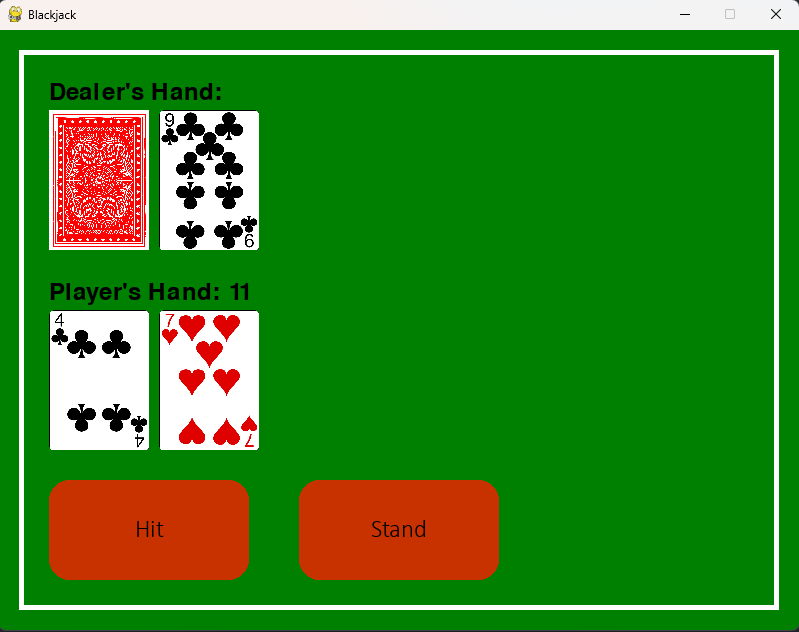
Kristoffer Bille Faaborg

Rybners Tekniske gymnasium

09-05-2025

Blackjack-python



# Titelblad

Uddannelsessted: Rybners Tekniske Gymnasium

Uddannelse: Højere Teknisk Eksamen (HTX)

Fag & Niveau: Programmering B

Emne: Blackjack i Python og Pygame

Vejleder: Carsten Finn Sørensen

Projektperiode: 16/01/2025 kl. 8:15 - 09/05/2025 kl. 12:00

Afleveringsdato: 09/05/2025 kl. 12:00

Sidetal:

Antal anslag: 17969 anslag / 2625 ord

Skrevet af: Kristoffer Bille Faaborg

Underskrift:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Kristoffer Faaborg



Dato for underskrift (dag/måned/år)

\_\_\_\_\_-\_\_\_\_\_-\_\_\_\_\_\_\_\_\_\_



Indholdsfortegnelse

[Titelblad 1](#_Toc197719931)

[Forord og indledning 5](#_Toc197719932)

[Opgaveformulering 5](#_Toc197719933)

[Brainstorm 5](#_Toc197719934)

[Brainstorm af funktioner 6](#_Toc197719935)

[Krav 7](#_Toc197719936)

[Hårde 7](#_Toc197719937)

[Bløde 7](#_Toc197719938)

[Værktøjer 7](#_Toc197719939)

[Python 7](#_Toc197719940)

[Pygame og Pygame widgets 8](#_Toc197719941)

[Random 8](#_Toc197719942)

[Andre værktøjer 8](#_Toc197719943)

[Spilbeskrivelse 8](#_Toc197719944)

[Brugervejledning 9](#_Toc197719945)

[Teknisk beskrivelse af programmet 9](#_Toc197719946)

[Biblioteker 9](#_Toc197719947)

[Bestemmelse af vinder 11](#_Toc197719948)

[Flowchart 13](#_Toc197719949)

[Del test og afprøvning 15](#_Toc197719950)

[Konklusion 16](#_Toc197719951)

[**Bløde** 16](#_Toc197719952)

[Referencer 18](#_Toc197719953)

# Forord og indledning

Programmet er lavet i forbindelse med HTX programmering B eksamensprojekt, hvor der blev valgt et projektoplæg at arbejde ud fra. Denne opgave tager udgangspunkt i følgende projektoplæg:

*Projektoplæg 3: Underholdning*

*Formål: Udvikling af programmer til at underholde flere personer.*

*Beskrivelse: I dette projektoplæg kan man lave et program der kan underholde diverse målgrupper. Man kan vælge at designe noget til en bestemt målgruppe eller designe noget mere bredt der omfavner flere målgrupper. Nogle eksempler indenfor dette projektoplæg kunne være:*

* *Kortspil, terningespil, brætspil*
* *4 på stribe*
* *Roulette*
* *Hangman*
* *Vendespil*
* *Matador*
* *Black Jack*
* *Sænke slagskibe*
* *Yatzy*

# Opgaveformulering

For at bestemme hvilken type spil eller underholdning der vil blive arbejdet med bruges brainstorm til at finde flere ideer til projektet og derefter bestemme sig fir en af dem.

## Brainstorm

Først blev der lavet en brainstorm over de forskellige spil og underholdning der kan laves ud fra projektoplægget.

Et billede, der indeholder tegning, skitse, Stregtegning, clipart

Indhold genereret af kunstig intelligens kan være forkert.

Figur : Brainstorm over spil og underholdning

Figur 1 viser brainstormen der blev lavet til at bestemme hvilket spil eller underholdning opgaven vil gå ud fra. Der blev valgt blackjack som projektet vil blive udarbejdet fra.

## Brainstorm af funktioner

Efter blackjack blev valgt som projektet skal der findes funktioner til spillet. For at bestemme funktionerne blev der lavet endnu en brainstorm hvor ideer til funktioner udarbejdes.

Et billede, der indeholder skitse, Stregtegning, clipart, illustration/afbildning

Indhold genereret af kunstig intelligens kan være forkert.

Figur : Brainstorm over funktioner til spillet

Figur 2 viser brainstormen over funktionerne spillet kan have. Ud fra brainstormen findes de funktioner som er nyttigt til projektet og hvilke der er overflødige.

## Krav

Ud fra de valgte funktioner fra brainstormen blev der bestemt hvilke af kravene som er essentielle for projektet og hvilke der kun er til forbedring. De bliver også kaldt hårde og bløde krav, hvor de hårde krav skal til bedste evne blive fuldendt hvor de bløde krav kun er ekstra.

### Hårde

* Trække kort ("Hit") og stoppe ("Stand") i henhold til reglerne
* Dealeren trækker kort indtil hånden har en værdi på mindst 17
* Få en vindererklæring baseret på spillets regler
* Se spillerens og dealerens kort som tekst
* Skjule det ene af dealerens kort indtil "Stand" eller "Bust"
* Se spillerens pointsum løbende
* Spilleren eller dealeren "buster" hvis de overstiger 21 point
* Se om spilleren, dealeren eller ingen har vundet (uafgjort)

### Bløde

* Kortgrafik og enkel brugergrænseflade
* Mulighed for "Double Down"
* Pengesystem til at byde med
* Mulighed for "Insurance"
* Animationer for korttræk
* Flere spillere
* Computerstyrede spillere med strategi

## Værktøjer

Til at udvikle Blackjack-spillet blev der anvendt følgende programmeringsværktøjer og biblioteker

## Python

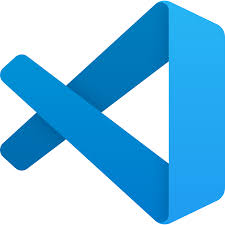
Python er kodesproget som vil blive brugt i projektet til at udvikle det. Python er velegnet til mindre spilprojekter med det enkelte og ligefremme sprog. Som stadig har alle de nødvendige funktioner til at udvikle projektet. Python har også mange biblioteker som gør det nemmere at kode da man kan få et bibliotek til at hjælpe.

### Python (programming language) - WikipediaPygame og Pygame widgets

Figur : Billede af Python

De vigtigste biblioteker der bliver brugt i projektet er Pygame og Pygame widgets. Pygame bruges til at lave GUI’en som er alt det grafiske som brugen ser og interagerer med. Pygame widgets er en tilføjelse til Pygame hvor den hjælper med at lave mere komplekse ting som knapper.

### Random

Random er også et python bibliotek som bliver brugt i projektet. Det bruges til at gør kortne tilfældige så det ikke er koden selv der skal bestemme næste kort.

Figur : Billede af Visual Studio Code

Random kører dog også selv på et på et program og er derfor ikke rigtig tilfældig, i dette tilfælde betyder det intet men man skal være opmærksom på at man kan med de rigtige værktøjer forudse random biblioteket.

## Andre værktøjer

Koden blev skrevet i Visual Studio Code som hjælper med at finde fejl i koden og kan køre koden. Det er en fordel da programmet vil blive kort i flere filer som gør hver deres ting, for at kunne holde styr på koden nemmere.

Figur : Billede af Github Copilot

I Visual Studio Code er der det som heder extensions. Det er der hvor man downloader kode sproget Python så man kan køre koden og Python værktøjer som tjekker koden for fejl eller problemer. En anden vigtig extensions er Github copilot, som ve hjælp af AI kan hjælpe med at forbedre og skrive kode.

## Spilbeskrivelse

Spillet, der er udviklet, er en digital version af kortspillet Blackjack. Spillet er til én spiller mod computeren, som fungerer som dealer. Spillet starter med, at både spiller og dealer får to kort. Spilleren kan vælge mellem at "hit" (trække et nyt kort) eller "stand" (stå og lade dealeren spille). Dealeren trækker kort indtil summen er 17 eller mere. Målet er at komme så tæt på 21 som muligt uden at overskride det. Der bliver vist tekstinformation undervejs, så spilleren kan se både sine egne og dealerens kort, dog med én af dealerens kort skjult, indtil spilleren står eller buster.

# Brugervejledning

Når programmet startes åbnes først en menu, hvor spilleren har muligheden for at trykke ”play” eller ”quit”, for enten at starte spillet eller lukke spillet ned. Når spilleren trykker ”play” vil de komme til en ny side spillefladen, hvor 4 kort vil være fremme. To af dem er spillerens og de andre to er dealerens, hvor den ene af dealens er skjult. Spilleren har derefter muligheden for at trykke ”hit” eller ”stand” for enten at få et mere kort eller stoppe og lade dealeren spille, man kan også bruge tastaturet med knapperne ”s” og ”h”. Hvis Spilleren trykker ”hit” og point kommer over 21 har spilleren ”busted” og har tabt. Hvis spilleren derimod trykker stand vil dealeren spille indtil deres point er 17 eller ovre. Hvis dealeren ”buster” vinder spilleren eller hvis spillerens point er over dealerens vinder spilleren også. Hvis dealeren har flere point end spilleren vinder dealeren. Og hvis spilleren og dealeren har samme point vil det være en ”tie” eller uafgjort. Point vises løbende. Når spillet er slut, vises resultatet med en besked som fx "Winner", "Loser" eller "Tie". Når runden er ovre vil ”hit” og ”stand” knapperne fjernes og erstattes med en restart til at starte ny runde.

# Teknisk beskrivelse af programmet

## Biblioteker

Pygame biblioteket bliver brugt til at lave spillefladen og alt grafisk.

Et billede, der indeholder tekst, skærmbillede, software, Font/skrifttype

Indhold genereret af kunstig intelligens kan være forkert.

Figur : Billede af kode eksempel med Pygame

Figur 6 viser hvordan man importere et bibliotek som Pygame, efter man har downloadede ellers vil der komme en error. ”Pygame.init()” bruges til at starte pygame op, uden vil Pygame ikke virke. ”screen = pygame.display.set\_mode((SCREEN\_WIDTH, SCREEN\_HEIGHT))” bruges til at lave et vindue hvor grafikken kan være i, den har så navnet screen, som kommer i brug senere.

Et billede, der indeholder tekst, skærmbillede

Indhold genereret af kunstig intelligens kan være forkert.

Figur : kodeeksempel af Pygame widgets

Pygame widgets er et ekstra bibliotek til Pygame som kan lave mere komplekse ting som knapper, slidere osv. Figur 7 viser hvordan man laver en knap i Pygame widgets. Her er den en del af en funktion da den skal kunne kaldes fra en anden fil, og har brug for at vide hvilket vindue den skal side på og i dette tilfælde er det en den af screen, som vi så i figur 6. Det meste af koden er til at bestemme udsende og placering af knappen. ”onClick” bruges til at bestemme hvad der sker når knappen trykkes. Her bruges den til at kalde en funktion som genstarter spillet. Og til sidst bruges ”return {'buttonRestart': restart\_button}” til at sende en variable over til den anden fil, for eller vil man ikke kunne bruge ”hide” og ”show” funktionen som ses lige ovre. Dette er ikke nødvendig hvism na skriver i samme fil hele tiden.

## Bestemmelse af vinder

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype

Indhold genereret af kunstig intelligens kan være forkert.

Figur : kode af bestemmelse af vinder

Figur 8 viser en del af koden som bestemmer hvem der har vundet. Her bruges ”if winner:” til at bestemme om der er data i variablen, hvis der ikke var vil den gå videre fordi så har spilleren ikke færdiggjort sit træk endnu. Hvis der så er data i variablen, tjekker den hvad, og afhængigt hvad der er i bestemmer det så hvem der er vinderen.

Et billede, der indeholder tekst, skærmbillede

Indhold genereret af kunstig intelligens kan være forkert.

Figur 9: Kode til beregning score

Figur 9 viser hvordan at koden beregner point for spilleren og dealeren, det er også derfor den er i en funktion så den kan bruges til både spiller og dealer.

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype

Indhold genereret af kunstig intelligens kan være forkert.

Figur : Finde billeder af kort

Figur 10 viser koden som erstatter kort teksten med billedet. Der bruges Pygame funktionen “pygame.image.load(image\_path)” først til at finde billedet og sætte det i en variabel. Derefter bruge ”pygame.transform.scale(placeholder\_image, (desired\_width, desired\_height))” til at lave den endelige kortbillede som senere kan blive kaldt, da dette kun laver en liste med alle kort billeder.

## FlowchartEt billede, der indeholder diagram, tekst, skitse, Plan Indhold genereret af kunstig intelligens kan være forkert.

Figur 11: flowchart af Menu.py

Figure 11 viser flowcharten for menu.py som er menuen, her ser den om hvor vidst spilleren har trykket på knapperne, men først bygger den GUI fladen.

Et billede, der indeholder tekst, diagram, skitse, Teknisk tegning

Indhold genereret af kunstig intelligens kan være forkert.

Figur : Flowchart af programmet

Figur 12 viser en simpel version af flowchartet til programmet. En vigtig detalje er nr. 2 kasse fra toppen hvor den funktion/handling der sker kommer fra en anden fil, board.py. der bruges mange steder i programmet men i den simple version er det kun starten der er væsentlig.

# Del test og afprøvning

Gennem hele projektet har der lavet del test til at se om koden virker. Og under vejs har der været flere udfordringer.

Et billede, der indeholder tekst, skærmbillede, Rektangel, grøn

Indhold genereret af kunstig intelligens kan være forkert.

Figur : problem med knapper

Den størte udfordring har med pygame widgets at gøre, da selv efter menu GUI fladen blev lukket med ”pygame.QUIT()” forsatte knapperne med at eksitere. Man kan se på figur 13 hvordan ”play og ”quit” knapperne stadig eksitere selv efter menuen er lukket, og er nu problematisk da det fylder meget af skærmen.

Det blev løst ved at Copilot hjalp med at finde funktionen ”button.hide()” med den vil den fjerne/usynliggøre knapperne så de ikke længere kan bruges. Der kom så ydreligere et problem når man skriver koden i flere filer for så vil variablen/navnet på knappen ikke genkendes i de andre filer, og da knapperne bliver lavet i board.py og bruges i blackjack.py giver det problemer når man ikke kan fjerne dem.

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype

Indhold genereret af kunstig intelligens kan være forkert.

Figur : kode af hide og show knapper fra andre filer

Det problem blev løst ved at sende variablen/navnet på knappen som en ”return” igennem funktionen, som blev vist i figur 7. Figur 14 viser hvordan den return bruges til at ”hide” eller ”show” knapperne.

# Konklusion

Spillet der blev lavet er fuldt funktionel og er en god prototype til at vise hvad blackjack er. Ud fra kravene der blev sat kan man vurdere hvor vidt om det blev fuldelt, både hårde og bløde krav.

**Hårde**

* Trække kort ("Hit") og stoppe ("Stand") i henhold til reglerne
* Dealeren trækker kort indtil hånden har en værdi på mindst 17
* Få en vindererklæring baseret på spillets regler
* Se spillerens og dealerens kort som tekst
* Skjule det ene af dealerens kort indtil "Stand" eller "Bust"
* Se spillerens pointsum løbende
* Spilleren eller dealeren "buster" hvis de overstiger 21 point
* Se om spilleren, dealeren eller ingen har vundet (uafgjort)

### **Bløde**

* Kortgrafik og enkel brugergrænseflade
* Mulighed for "Double Down"
* Pengesystem til at byde med
* Mulighed for "Insurance"
* Animationer for korttræk
* Flere spillere
* Computerstyrede spillere med strategi

Af de hårde krav blev alle opfyldt hvor at alle nødvendige funktioner som blev stillet krav til er lavet og gør at spillet kan alt nødvendigt. Af de bløde krav blev en enkelt opfyldt. Kortgrafik og enkel brugergrænseflade blev opfyldt ved at ændre kort teksten til billeder af kort, det samme med dealerens skjulte som er bagsiden af et spillekort. På samme tid visualiserer spillefladen en almindeligt, dog simplificeret blackjack bord med den grønne baggrund og hvide stribe hele vejen rundt om bordet. Resten af de bløde krav blev ikke lavet, med de ekstra funktioner som ”double down” og ”insurance” som gør spillet mere sjovt og avanceret.

# Referencer

Kort billeder: <https://code.google.com/archive/p/vector-playing-cards/downloads>

bagside billede af kort: <https://opengameart.org/content/colorful-poker-card-back>

kode findes på: <https://github.com/krisserdk/Eksamens-projekt.git>