

**Skak Motor**

Gruppe: Patrick, Christian og Kresten

Dato: 21/2-2021

Fag: Programmering

|  |
| --- |
| **Skak motor**  [**Resumé**](#_sj8u8sfbdmum) **2**  [**Projektformulering**](#_yrp1stopoboe) **2**  [**Programbeskrivelse**](#_ov8ig0fd90c2) **2**  [**Funktionalitet**](#_a71bigx1frds) **3**  [Kravspecifikation](#_bk2vlfmx2vne) 3  [Skak motor](#_lxj3nj8nlriy) 3  [Grafisk brugerflade](#_qqudni3i699p) 4  [**Brugergrænseflader**](#_fwvxuuagkexu) **5**  [Skak Motor](#_djnpt8u90hsr) 5  [Windows forms GUI applikation](#_812od3vzbgq3) 5  [Konsol applikationen](#_3t4mf2j89hk0) 6  [**Rutediagrammer**](#_arfgecdu70xx) **6**  [Brikker](#_31u5nix5irxa) 7  [**Pseudokode**](#_oa6xo86q4no6) **8**  [**Udvalgt kode**](#_njlivt3ind2b) **10**  [Konge i skak](#_oiregu1fk14j) 10  [Dangerzone](#_kfzhru898nau) 11  [**Test af programmet**](#_d8q1oqe68xpv) **12**  [Grafisk brugerflade - Praktisk test](#_vvtiufmfjomv) 12  [Skakmotor - Unit tests](#_uzb6vr2e8qcf) 12  [**Konklusion**](#_ayfag850dw2d) **12**  [**Bilag**](#_lp5haz5lrmuk) **13** |

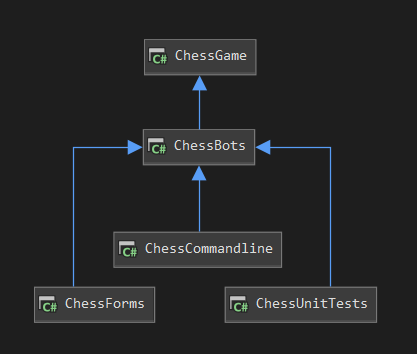
# Resumé

Projektet er et spil skak formidlet som et biblioteks modul, samt andre moduler til blandt andet at fremvise vores spillet grafisk og en skakbot man kan spille imod. Vi har lavet projektet ud fra analyse og design hvor vi blandt andet brugte CRC kort *(Kig bilag 9)*, som vi har haft ændret på efter behov. Vi har også brugt en modelleringsmetode, hvor man markerer udsagnsord og navneord i en beskrivelse af spillet *(Kig bilag 10)*.

# Projektformulering

Kort sagt er vores projekt at lave skak med nogle flere elementer, som kan slukke folks tørst efter at spille skak med hinanden.

# Programbeskrivelse



*Bilag 1: Generel oversigt over programmets moduler.*

Vores program er en skak motor, der kan bruges til nemt at implementere skak til et givent brugergrænseflade eller til et andet projekt, som kræver et skaksystem såsom en skakbot. Man kan implementere flere forskellige game modes, brikker og indstillinger, den er designet så modulært at den overholder alle krav beskrevet i [Kravspecifikation](#2w6wkyoibhd4).

Udover vores skak motor har vi en konsol applikation, som skal forestille at være den mest simple implementation af vores motor. Brikkerne vises gennem konsollen ved brug af symboler fra en bestemt unicode block[[1]](#footnote-0). Man laver træk ved at skiftes til at indtaste den algebraiske eller UCI-notation (UCI er bare feltet man flytter fra samt feltet man flytter til, ideelt til skakcomputere) for det ønskede træk[[[2]](#footnote-1)][[[3]](#footnote-2)]. Kommandoen “c” som ændrer farve.

Yderligere består det overordnede skakprogram af et modul som udgør den grafiske overflade. Bygget op omkring Windows Forms, det gør det mere tilgængeligt og nemmere at spille skak uden at kende til de to implementerede notationsformer. Det gør det nemmere at navigere de funktioner, der er blevet tilføjet over tidens løb.

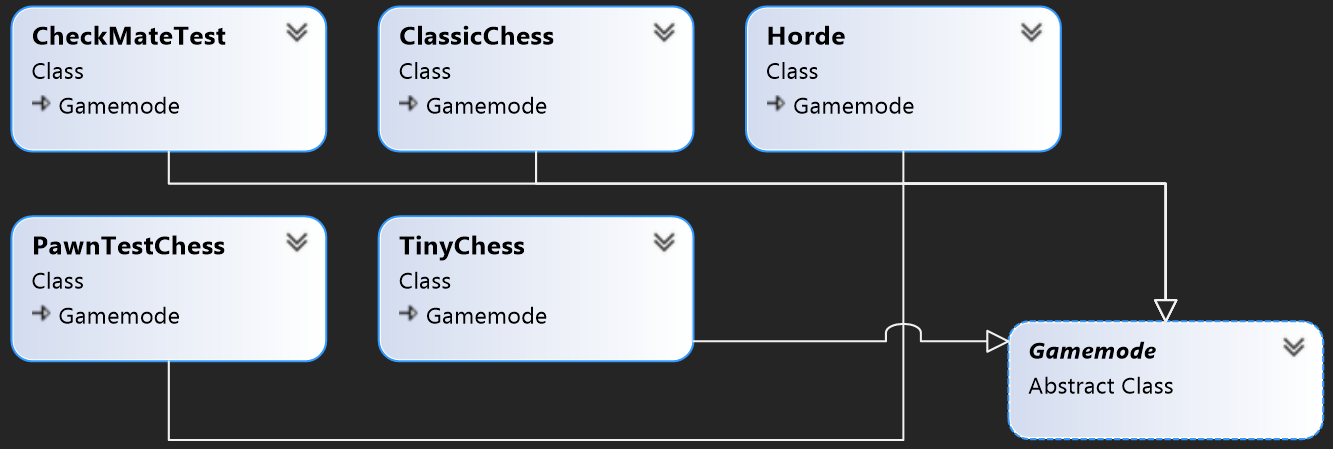
# Funktionalitet

### Kravspecifikation

Beskrevet er de krav vi stillede os.

#### Skak motor

* Generelle skakregler
  + Bevægelsesmønstre
    - Skal støtte alle brikker i skak
      * Mulighed for tilføjelsen af flere
  + Kongen kan blive sat i skak og skakmat
* Special regler
  + At tage en passant
  + At lave rokade
  + Rykke bonden dobbelt i starten
* Spille måde
  + Alene spiller (Mod bot)
    - Man skal kunne spille mod en CPU
  + Lokal multiplayer (split screen på en måde)
    - Man skal kunne spille sammen på en computer
  + Multiplayer over nettet
    - Man skal kunne spille med en anden spiller over nettet



*Bilag 2: Klassediagram over spil tilstande implementeret med Gamemode klassen.*

*Game modes* er et element som er implementeret i spillet. Det giver noget diversitet angående normal skak og en lidt anderledes form for skak. Game modes giver brugerne en anden grund til at spille spillet end normalt skak.

Vi vil gerne give udviklere muligheden til at ændre brikkerne de spiller med eller ændre positionen på brikkerne. Der er:

Horde skak (Hvid har mange bønner)[[4]](#footnote-3), 960 skak (Både hvid og sort har stillet brikkerne tilfældigt, men hvid og sorts brikker har de samme tilfældige positioner).

#### Grafisk brugerflade

Der skal også være noget visuelt at kigge på, når det er man spiller spillet:

Brikkerne skal have et udseende (hvis brikken er på hvid’s side, så er den hvid eller hvis brikken er en løber, så skal det ligne en løber). Til sidst skal der også være et skakbræt, som man kan spille på.

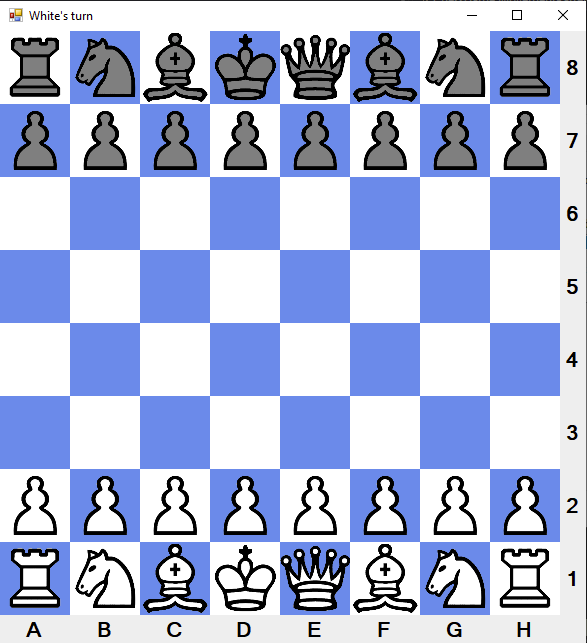
* Lyd
  + Når man flytter eller tager en brik
* Visuelt
  + Brikkerne skal have forskelligt udseende
  + Der skal være et bræt/koordinatsystem
  + Man skal kunne se hvilke træk en brik kan lave når man trykker på den
  + Man skal kunne højre trykke for at kunne markere felter med en farve (Hvilket kan hjælpe hvis man skal forklare noget til en person)
  + Mulighed for at kunne se hvem som er hvem (Give navn)
* Styremåde
  + Med museklik

# Brugergrænseflader

### Skak Motor

Selve skak programmet i sig selv har ikke nogen synlig brugergrænseflade. Den præsenterer en API som en anden programmør kan tage og arbejde med. Men vi har både en forms applikation og en konsol applikation som begge kan benytte vores bibliotek til at spille spillet.

### Windows forms GUI applikation



*Bilag 3: Skærmbillede af spillet.*

Dette modul er lavet med forms og måden man benytter den på er meget mere brugervenlig end konsol applikationen i det at man bare skal trykke på den brik man vil rykke og så trykke der hvor man vil trække den hen. Når man har trykket på den første brik vil der også komme markeringer på andre felter som viser hvilke du kan rykke over på. Hvis man højre trykker på felter så kommer der også en markering som kun ens egen klient kan se hvilket gør at man kan forklare positioner for folk der kigger med og markeringerne kan fjernes igen ved enten at højre trykke på dem eller venstre trykke et vilkårligt sted.

### Konsol applikationen



*Bilag 4: Skærmbillede af konsollen.*

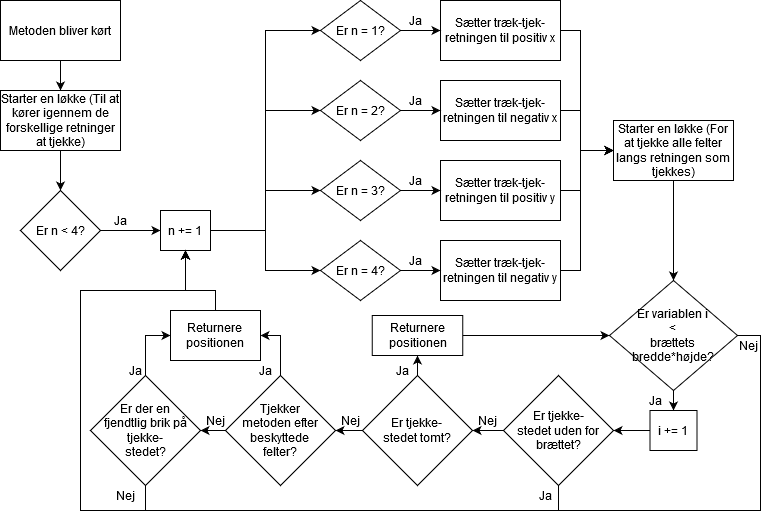
I denne her så spiller man ved at give sit input som algebraisk notation ind i konsollen. Algrabaisk notation er standard måden til at beskrive skaktræk med og de foregår ved at man skriver hvilken brik man vil rykke og hvor man vil trække den hen. Hvert felt har hver sin beskrivelse ved at den vandrette akse har bogstaverne A-H og lodrette har tallene 1-8.

Konsol applikationen var en slags starts forsøg på at lave noget rimelig hurtigt grafisk som gjorde vi kunne bruge vores program.

# Rutediagrammer

*Grafisk repræsentation af en algoritme eller en programdel.*

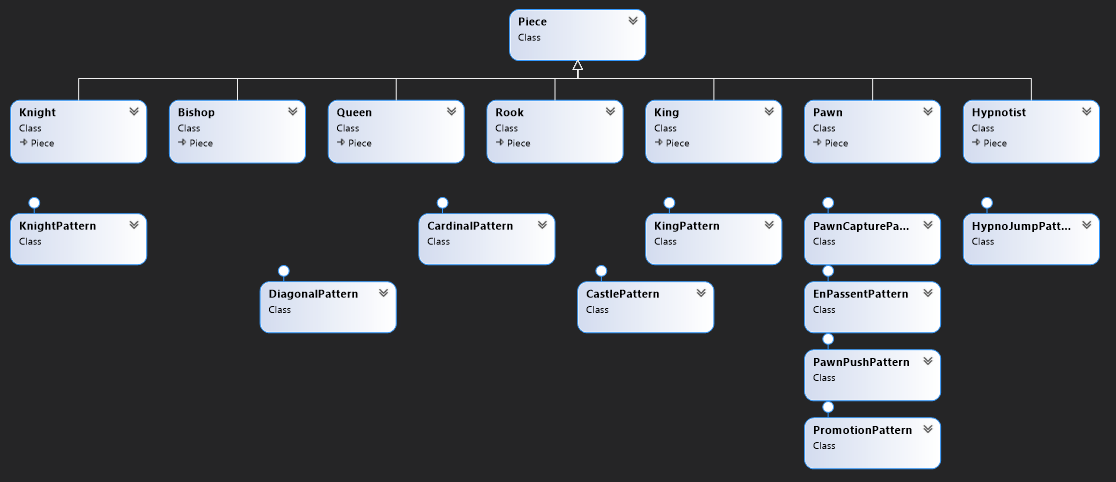
Her vil der blive beskrevet kode dele.



*Bilag 5: Flowchart.*

Det her er et flowchart over *CardinalPattern.cs*, hvilket er en af vores movement patterns som forskellige skakbrætter kan blive tildelt. Den kører gennem alle de forskellige mulige træk for en brik som har mulighederne for at bevæge sig kardinalt, hvilket er dronningen og tårnet. Den består af 2 løkker som henholdsvis bestemmer henholdsvis hvilken retning den skal tjekke og hvor mange felter væk fra brikken den skal tjekke.

#### Brikker



*Bilag 6: Klassediagram over brikker implementeret med Piece-klassen.*

Oversigt over brik typer med de movement patterns de hver i sær implementerer nedenunder, (hypnotist er en eksperimentel brik for at demonstrere man kan implementere andet)

# Pseudokode

**Pseudokode af skaknotation til træk:**

Nedenfor er den algoritme, som står for at oversætte et træk beskrevet i tekst til en *Move* instans. Som nævnt tidligere, støtter den både UCI (hvilket er ideelt at implementere i en skakmotor) samt standard algebraisk notation, som er standarden inden for turnerings verdenen.

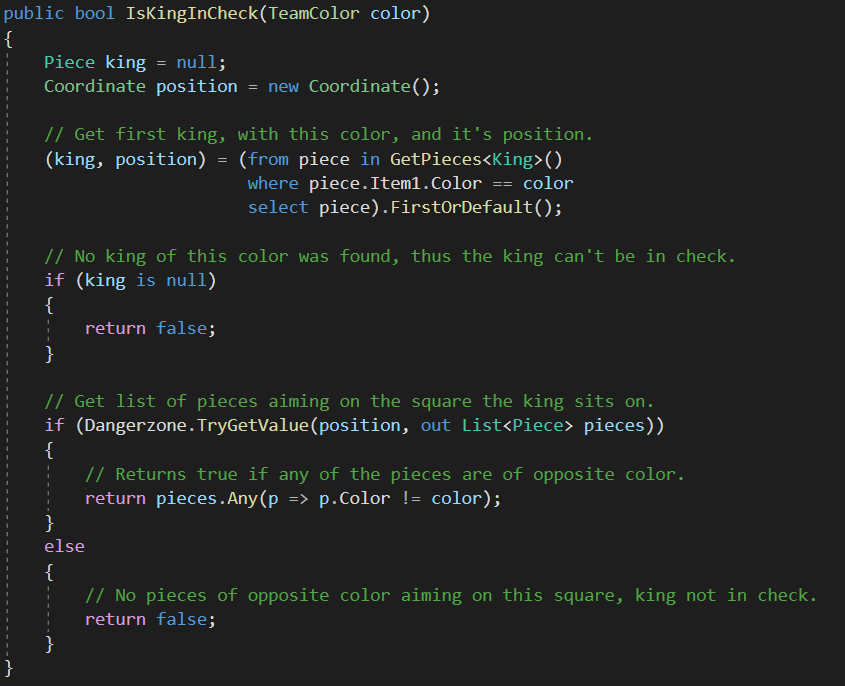
Givet en tekststreng, som indeholder det ønskede træk og notations typen det er beskrevet med, oversætter følgende kode det:

|  |
| --- |
| Hvis notations type = Universal Chess Interface  Hvis længden af notationen er større end eller lig 4  Sætter kilde lig med et nyt koordinat ud fra notationens placering.  Sætter destinationen lig med et nyt koordinat ud fra notationens placering.  Hvis længden af notationen er 5  Sætter promoverings målet til at være en brik aflæst fra notationen  Hvis notations type = standard algebraisk  Kører en løkke som gentagers sig ligeså mange gange som notationens længde  Hvis tegnet er et tal  Sæt sidste nummer indexet lig med længden den er gået igennem løkken  Hvis tegnet er et “x”  Sæt fanger variablen til sand  Hvis sidste nummer indexet er lig 0  Sæt brugerdefineret notation variablen til sand  GOTO: *findtræk*  Sæt destinationen lig med et nyt koordinat ud fra notationens sidste nummer index - 1 og 2 ind i notationen  Hvis notationens længde minus 1 er lig med sidste nummer index og tegnet i notationenen nummer sidste nummer index + 1 er et “=”  Sætter promoverings målet[[5]](#footnote-4) til at være en brik aflæst fra notationen  Sætter brik notationen[[6]](#footnote-5) til ingenting  Hvis sidste nummer index = 1  Sætter brik notationen til ingenting  Ellers  Sætter brik notationen til tegnet i notationen nummer 0  GOTO: *findtræk*  Fejlhåndtering  *findtræk:*  For hver af spillerens træk  Hvis brugerdefineret notation er sandt og trækkes brugerdefinerede notation er lig med notationen  Hvis trækket er lovligt  Returnere trækket  Hvis kilden ikke er ingenting og ikke er trækket  Spring over  Hvis destinationen ikke er ingenting og destinationen ikke er trækket  Spring over  Hvis fanger variablen ikke er ingenting og ikke er trækket  Spring over  Hvis brik notation ikke er ingenting og ikke er trækket  Spring over  Hvis promoverings målet ikke er ingenting og ikke er trækket  Spring over  Hvis trækket ikke er lovligt  Spring over  Returner trækket |

# 

# Udvalgt kode

### Konge i skak

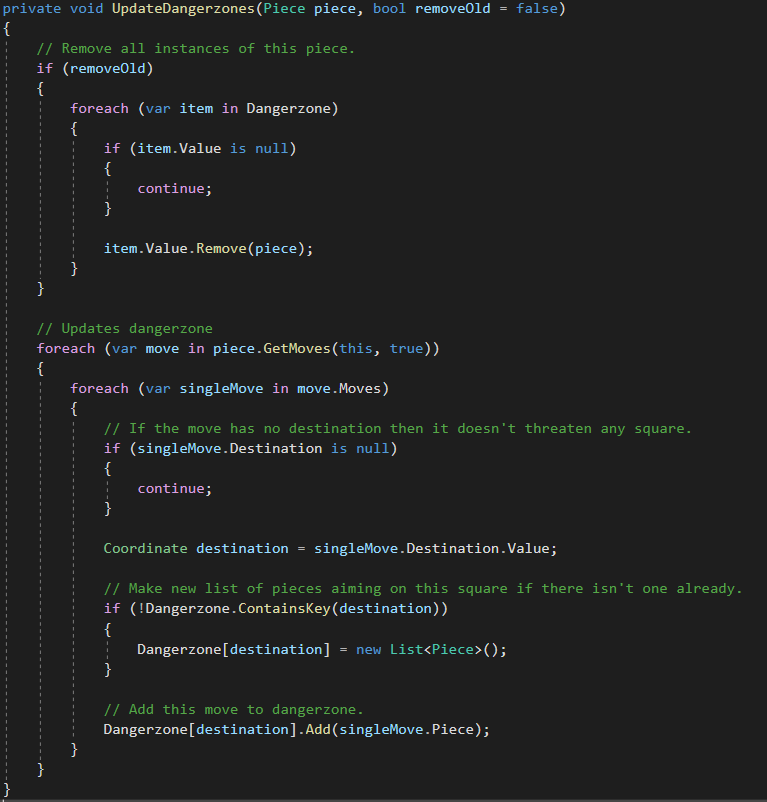


*Bilag 7: Skærmbillede af funktionen IsKingInCheck* *fra Chessboard.cs.*

Koden står for at tjekke om kongen er sat i skak. Kort sagt, fungerer det ved at tjekke om kongen står på et intersection felt[[7]](#footnote-6), hvis den gør er den i skak.

*Dangerzone[[8]](#footnote-7)* er samling af felter og de brikker som sigter på dem, den bliver opdateret hver gang der bliver lavet et træk.

### Dangerzone



*Bilag 8: Skærmbillede af UpdateDangerzone funktion* *fra* *Chessboard.cs.*

Givet en brik opdaterer koden *Dangerzone* samlingens referencer til denne brik efter alle de træk den har.

# 

# Test af programmet

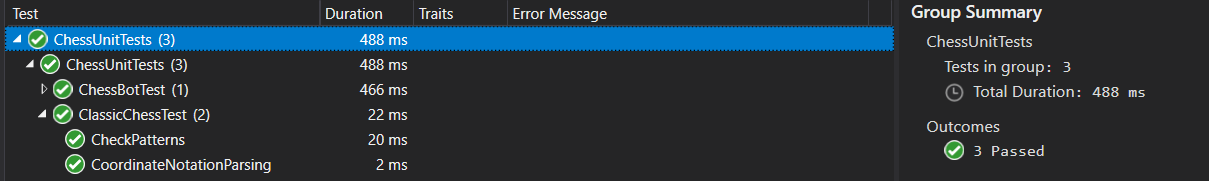
### Grafisk brugerflade - Praktisk test

|  |  |  |
| --- | --- | --- |
| Input | Forventet output | Faktisk output |
| Der klikkes på cirklen (radio knappen) | Cirklen bliver markeret | Cirklen bliver markeret. |
| Der klikkes på navnet (det ved siden af radioknappen) | Cirklen bliver markeret | Cirklen bliver markeret |
| Der klikkes på tekstfeltet | Tekstfeltet bliver markeret og man kan skrive | Tekstfeltet bliver markeret og man kan skrive |
| Der klikkes på et felt med en brik på | Brikken bliver markeret og man får nogle felter man kan rykke hen | Brikken bliver markeret og den viser nogle grønne prikker hvor de felter man kan rykke hen. |
| Der højreklikkes på et felt | Der kommer farve på feltet | Feltet bliver markeret med farve |
| Der venstreklikkes på et tomt felt | Der sker ingenting | Man fjerner markeringerne |

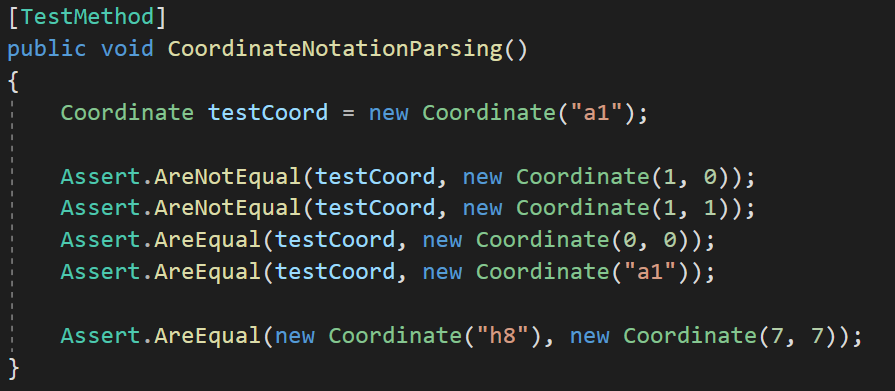
### Skakmotor - Unit tests

For at teste implementationen af nogle af de forskellige funktioner der er i vores skakmotor har vi lavet nogle unit tests.

Der er kode til at teste *parsing* af træk (implementationen er beskrevet i pseudokode afsnittet)



*Bilag 11: Unit test kørsel*



*Bilag 12: Test kode*

Tjekker om koordinat notation parser korrekt.

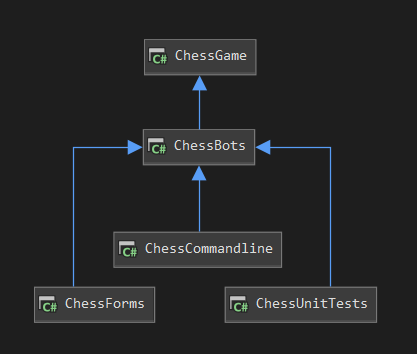
# Konklusion

Vi har opnået vores projektformulering og har fået lavet os et dejligt stort skakbibliotek med mange muligheder for at bruge forskellige funktioner og viderebygge eventuelle ikke udviklede funktioner man kunne have brug for i fremtiden.

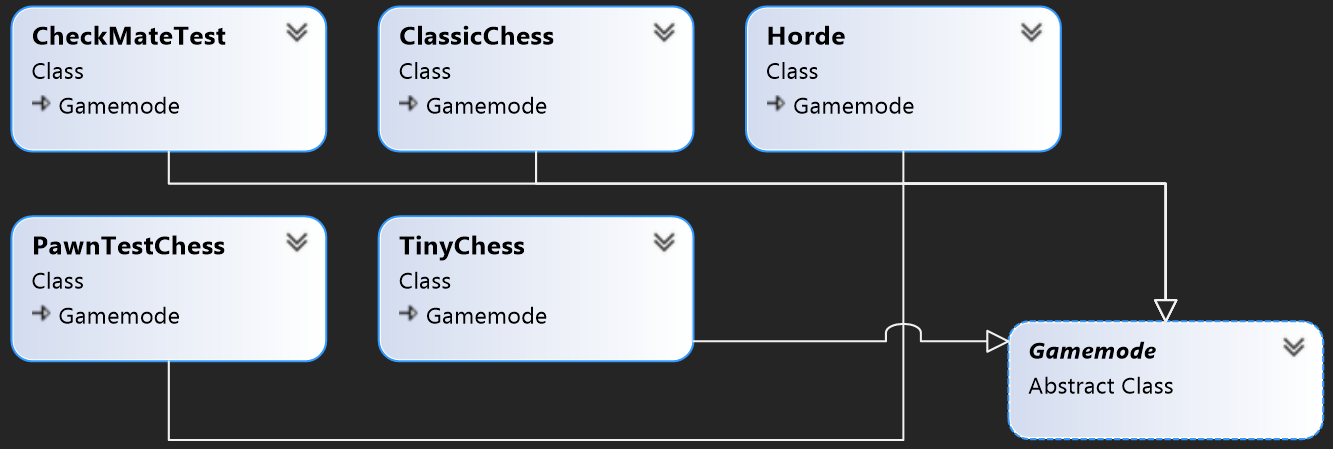
# 

# Bilag

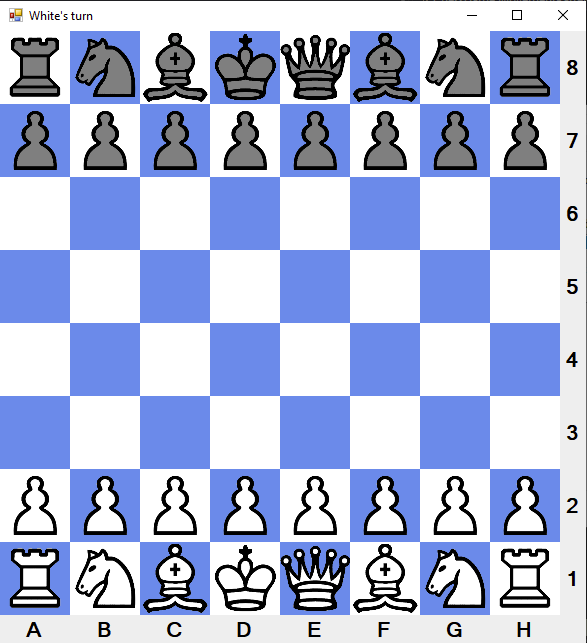
***Bilag 1: Generel oversigt over programmets moduler.***



***Bilag 2: Klassediagram over spil tilstande implementeret med Gamemode klassen.***



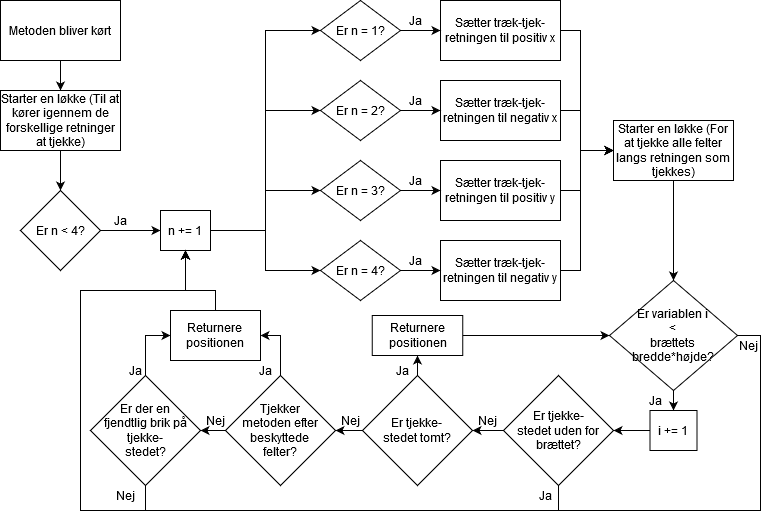
***Bilag 3: Skærmbillede af spillet.***



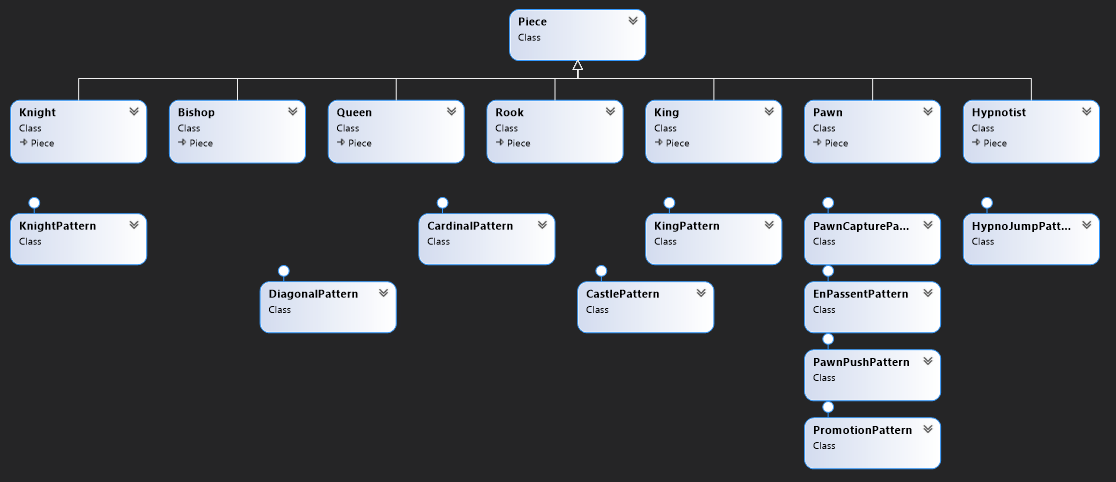
***Bilag 4: Skærmbillede af konsollen.***



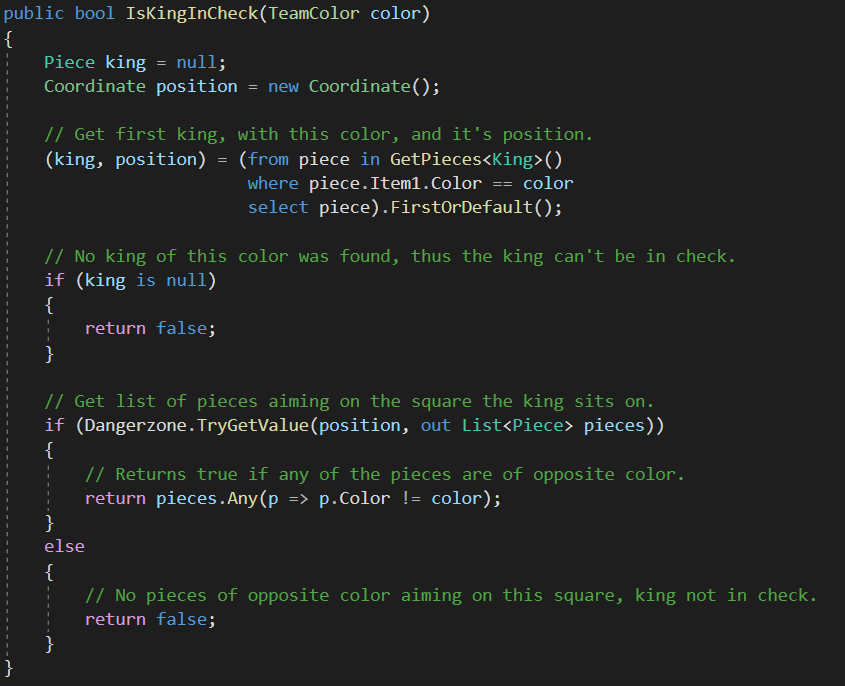
***Bilag 5: Flowchart.***

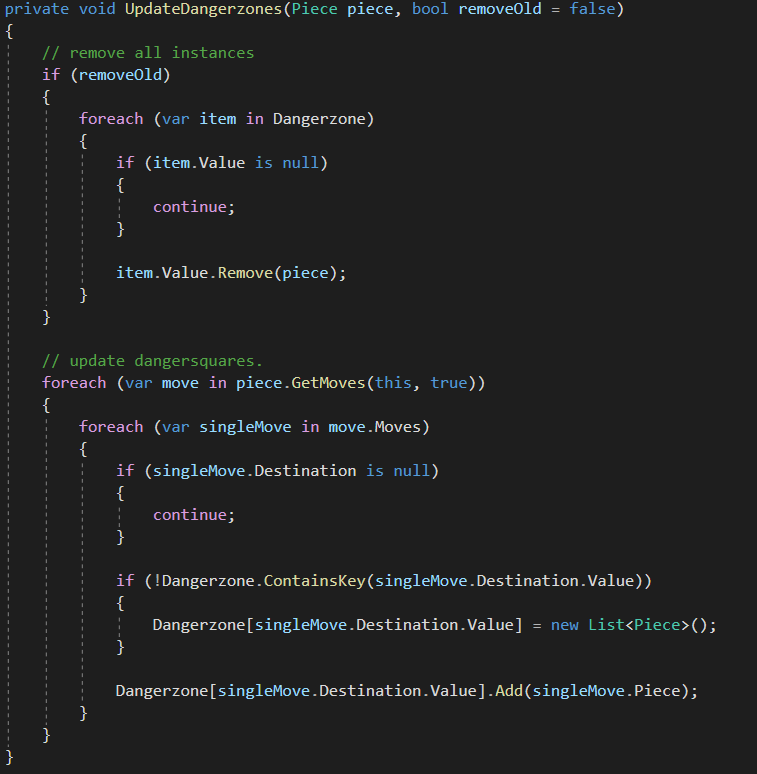


***Bilag 6: Klassediagram over brikker implementeret med Piece-klassen.***



***Bilag 7: Skærmbillede af funktionen IsKingInCheck.***



***Bilag 8: Skærmbillede af UpdateDangerzone funktion.***

***Bilag 9: CRC kort***

|  |  |
| --- | --- |
| **Class name:** | **Collaborators:** |
| Spil Type (Game mode) | bræt |
| **Responsibilities:** |  |
| Finde næste spillers tur |  |
| Indstille start position af bræt |  |
| Indstille urets start tid |  |

|  |  |
| --- | --- |
| **Class name:** | **Collaborators:** |
| Parti | Spiller (en liste over spillere) |
| **Responsibilities:** | Bræt () |
| Spil Type (Game mode) |  |
| Tur |  |
| Timer liste (En til hver spiller) (har en timer) |  |

|  |  |
| --- | --- |
| **Class name:** | **Collaborators:** |
| Bræt |  |
| **Responsibilities:** |  |
| Liste over brikker |  |
| Flytte brikkerne |  |
| Brætstørrelse |  |
| Udregn materiale værdi for alle brikker |  |

|  |  |
| --- | --- |
| **Class name:** | **Collaborators:** |
| Timer |  |
| **Responsibilities:** |  |
| Holde styr på en spillers tid |  |

|  |  |
| --- | --- |
| **Class name:** | **Collaborators:** |
| Brik | Bevægelsesmønster (hver brik har et specifikt bevægelsesmønster indstillet) |
| **Responsibilities:** |  |
| Position |  |
| Har flyttet sig? (Bool) |  |

|  |  |
| --- | --- |
| **Class name:** | **Collaborators:** |
| Bevægelsesmønster | Bræt |
| **Responsibilities:** | Træk/move |
| Udregne felter den kan flytte til |  |

|  |  |
| --- | --- |
| **Class name:** | **Collaborators:** |
| Bonde | Brik |
| **Responsibilities:** |  |
| Bevægelsesmønstre |  |
| Pointværdi (1) |  |

|  |  |
| --- | --- |
| **Class name:** | **Collaborators:** |
| Hest | Brik |
| **Responsibilities:** |  |
| Bevægelsesmønstre |  |
| Pointværdi (3) |  |

|  |  |
| --- | --- |
| **Class name:** | **Collaborators:** |
| Løber | Brik |
| **Responsibilities:** |  |
| Bevægelsesmønstre |  |
| Pointværdi (3) |  |

|  |  |
| --- | --- |
| **Class name:** | **Collaborators:** |
| Tårn | Brik |
| **Responsibilities:** |  |
| Bevægelsesmønstre |  |
| Pointværdi (5) |  |

|  |  |
| --- | --- |
| **Class name:** | **Collaborators:** |
| Dronning | Brik |
| **Responsibilities:** |  |
| Bevægelsesmønstre (Tårnets og løberens) |  |
| Pointværdi (9) |  |

|  |  |
| --- | --- |
| **Class name:** | **Collaborators:** |
| Konge | Brik |
| **Responsibilities:** |  |
| Bevægelsesmønstre |  |
| Pointværdi (uendelig, tabes brikken tabes spillet) |  |

|  |  |
| --- | --- |
| **Class name:** | **Collaborators:** |
| Træk/move | Koordinat |
| **Responsibilities:** | Point system |
| Brikker involveret i trækket (ved rokade er det tårn og konge) |  |
| Algebraisk notation gennem getstring |  |
| Om den tager en brik eller ej |  |

|  |  |
| --- | --- |
| **Struct name:** | **Collaborators:** |
| Koordinat (Ligesom vector2 i unity, beskriver positionen) |  |
| **Responsibilities:** |  |
| Beskrive en position på brættet |  |

***Bilag 10: Problemidentifikation***

|  |
| --- |
| Regeltekst: (Med markerede udsagnsord og navneord) Skak er et brætspil, hvor der er 2 spillere som spiller mod hinanden på et skakbræt.  I spillet har man nogle bønder, tårne, springere, løbere og en konge samt dronning på nogle forudbestemte pladser.  Man spiller ved at rykke sine brikker rundt på brættet og eventuelt tage modstanderens brikker.  Man vinder ved at sætte modstanderes konge i skak mat eller hvis modstanderen giver op.  Man kan stå lige, kaldt remis, hvis den spiller som har tur ikke kan lave nogle lovlige træk. Man kan også stå lige hvis begge spillere bliver enig om et remis, eller hvis ingen er villig til ikke at lave de samme træk igen.  Man kan ikke rykke sine brikker over i et tomt felt eller over for at tage en modstanders brik så længe at man ikke sætter sig selv i skak under det. Man kan ikke sætte sin brik over på et felt hvis en af sine egne brikker står på det.  Hvis man rykker en bonde op til modstanderens baglinje så kan den blive forfremmet til en vilkårlig officer.  Man kan rokere kongen og et vilkårligt tårn hvis begge af de valgte brikker ikke har blevet rykket i løbet af spillet. Det foregår ved at man rykker kongen 2 felter imod tårnet og rykker tårnet om på feltet bag kongen. |

1. <https://unicode-table.com/en/sets/chess-symbols/> [↑](#footnote-ref-0)
2. <https://en.wikipedia.org/wiki/Algebraic_notation_(chess)> [↑](#footnote-ref-1)
3. <https://en.wikipedia.org/wiki/Universal_Chess_Interface> [↑](#footnote-ref-2)
4. <https://www.chess.com/terms/horde-chess> [↑](#footnote-ref-3)
5. Hvis en bonde er nået til bagerste række kan den forfremmes til en brik, promoverings målet er brik notationen for det den forfremmes til [↑](#footnote-ref-4)
6. Brik notationen er det bogstav, som repræsenterer en brik, f.eks. betyder ‘K’ konge og ‘Q’ dronning (eng. Queen) [↑](#footnote-ref-5)
7. Et *intersection* *felt* et et felt på skakbrættet, som mere end 1 brik sigter på/truer [↑](#footnote-ref-6)
8. [www.youtube.com/watch?v=siwpn14IE7E](https://www.youtube.com/watch?v=siwpn14IE7E) [↑](#footnote-ref-7)