

# Tetris

Beskrivelse af program til aflevering i Objektorienteret Programmering, ved Keld Helsgaun

Dette er en beskrivelse af mit Tetris-spil, som brugeren oplever det. Det er med andre ord en beskrivelse af hvad mit Tetris kan, og hvordan det opfører sig i alle de situationer der kan opstå.

Programmets forløb, fra det sættes i gang til det terminerer, er:

- 1 **programmet sættes i gang, og Tetris dukker op i et nyt vindue**
- 2 brugeren sætter **spillet** i gang
- 3 brugeren spiller spillet
- 4 spillet ender
- 5 brugeren lukker vinduet, => programmet terminerer

Dette er det overordnede forløb = "rammeforløbet", indenfor hvilket spillet udfolder sig.

Jeg vil nu uddybe hvert enkelt af de ovenstående punkter, og således gennemgå og beskrive alle mit programs og mit spils elementer, efterhånden som de kommer i brug. Spilleets regler og funktion forklares under punkt 2, 3 og 4.

## 1 **programmet sættes i gang, og Tetris dukker op i et nyt vindue**

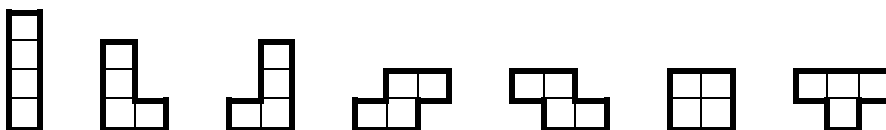
Programmet sættes enten i gang fra Java-udviklingsmiljøet [ IntelliJ IDEA / lign. ] eller ved at dobbeltklikke på dets symbol i den mappe det ligger i / på eventuelle genveje til det.

Tetris-vinduet fylder ikke hele skærmen, men er placeret centralt på den. Det udgør hele den grafiske side af programmet, og det er igennem dét programmet kommunikerer med brugeren. Brugeren får alle sine oplysninger om spillet ved at iagttage dette vindue. Det er inddelt i to felter: en spilleflade og en menu. Spillefladen er placeret i højre side af vinduet, stødende op til vinduets øvre + højre + nedre kant. Det er her Tetris foregår.

I venstre side af vinduet ses spillets menu. Øverst i den ligger knappen " Start ", og derunder nogle tekstfelter der viser brugerens score og det antal brikker der indtil nu er faldet ned.

## 2 brugeren sætter **spillet** i gang

Dette gør brugeren ved at trykke på en stor, flot, rød knap mærket " Start ", med musen. Ved tryk på " Start " falder den første af spillets brikker – de såkaldte " tetriminoer " – ned fra midt i vinduets top. En tetrimino består af fire lige store kvadrater, som er i berøring med hinanden. Der findes syv typer : I, L, J, S, Z, O og T:



Tetriminoerne " ligger " altid enten " ned " eller " står op ". Dvs. at to af siderne i deres kvadrater er vandrette [ og de to andre dermed lodrette ]. Tetriminoen begynder nemlig i en sådan position, og kan kun rotere i spring på 90 °. Kvadraternes placering indbyrdes i tetriminoen er uforanderlig, og når tetriminoen roterer, bevarer den sin form.

En tetriminos **automatiske** adfærd på spillefladen er simpel: den falder indtil den rammer noget – enten spillefladens nedre kant / en anden tetrimino, som er landet tidligere. Når én tetrimino lander, sendes den næste afsted. Den lander også på bunden af vinduet / på en af de foregående tetriminoer, og sådan bliver det ved. Langsomt opbygges der altså et tårn af tetriminoer hvilende på spillefladens nedre kant. Men der er kun **én tetrimino** i fald **ad gangen**.

### 3 brugeren spiller spillet

Brugeren kan ikke standse en tetriminos fald, men han / hun kan påvirke den ved hjælp af piletasterne:

- ➦ [ højre pil ] / [ venstre pil ] rykker tetriminoen til højre / venstre
- ➦ [ pil op ] roterer tetriminoen 90 ° med uret
- ➦ [ pil ned ] flytter tetriminoen nedad, i tillæg til dens automatiske fald

Når tetriminoen lander, kan brugeren ikke længere påvirke den.

Brugeren **spiller** altså med piletasterne, og interagerer med **programmet** igennem musen. At sætte spillet i gang er også en rammehandling = ikke et træk i Tetris, og udføres også med musen.

Hvis brugeren ikke rører en finger, vil den ene tetrimino efter den anden komme til syne i toppen af spillefladen, falde lodret ned og tilsammen danne et én brik bredt tårn op igennem midten af spillefladen, som vokser hurtigt. For at forhindre dette, kan brugeren styre tetriminoerne ud til siderne, så de danner et bredt + lavt tårn. Men dette vil ikke forhindre tårnet i at vokse, blot sinke dets vækst.

Brugerens eneste mulighed for at gøre tårnet **lavere**, er at placere de nedfaldende tetriminoer så deres kvadrater i én vandret række fylder hele spillefladens bredde, uden huller. En ubrudt vandret række af kvadrater = fra spillefladens højre kant til dens venstre, uden et eneste kvadrats tomrum, **forsvinder !!** Dele af tårnet som måtte være placeret højere oppe i spillefladen, falder ned og lander på evt. hvilende tetriminoer / spillefladens bund lodret under dem.

Når en række forsvinder, bliver tårnet således minimum et kvadrat lavere i hele dets bredde.

Det er kampen imellem disse to kræfter:

- ➦ den ubønhørlige tyngdekraft – der får en strøm af tetriminoer til at regne ned fra himlen – overfor
- ➦ brugers geometriske skarpsynethed og fingres hurtighed,

der udgør spændingen i Tetris.

#### Pointsystem

Mit pointsystem er meget simpelt. Brugeren får 1 point for hvert kvadrat som intræder i spillefladen uden at brugeren dør af det. Og da samtlige tetriminotyper består af 4 kvadrater, får han / hun 4 point per tetrimino. Når en række af kvadrater forsvinder, giver hvert af de forsvindende kvadrater 10 point.

Eftersom mit spil slutter automatisk efter 100 tetriminoer, kan en bruger maksimalt score  $100 \cdot 4 + 10 \cdot 400 \Leftrightarrow 400 + 4000 = 4400$  point.

#### 4 spillet ender

Vokser tårnet af tetriminoer sig så højt at det overstiger en vis maksimalhøjde, har brugeren tabt spillet, og det ender. Brugerens score samt antallet af nedfaldne brikker, vises i deres respektive tekstfelter. Hvis brugeren vil spille dette hylende morsomme spil igen, er han / hun nødt til at sætte programmet i gang på ny.

Men er brugeren dygtig nok, fortsætter regnen af tetriminoer, til og med tetrimino nummer 100. Herefter bliver der " tørvejr ", og brugeren har vundet.

Ønsker brugeren at gemme sine data, er det frem med et Word-dokument / pen og papir – programmet har på dette stadie ingen " Gem "-funktion.

#### 5 brugeren lukker vinduet, => programmet terminerer

Brugeren kan til enhver tid lukke vinduet med musen. Dette vil få programmet til at terminere – og dermed afbryde spillet.

Men brugeren skal lukke programmet: det terminerer ikke automatisk. Lukker brugeren det ikke, vedbliver programmet derfor at køre indtil selve computeren lukkes ned.

Slut !