Pacman opgave

I denne opgave har jeg løst følgende ekstraopgaver:

1. Spilleren kan modificere banens størrelse før spilstart ved at indtaste *run PacMan* efterfulgt af den størrelse, banen ønskes. Banen er kvadratisk, så der skal blot indtastes én værdi. Gives der ikke en specifik størrelse på banen, laves banen automatisk 10x10 felter ved spilstart.
2. Jeg har implementeret et spøgelse i spillet, der forfølger Pacman. Spillet stoppes, hvis spøgelset fanger Pacman, eller går ind i det andet spøgelse.
3. Der er ligeledes et spillerstyret spøgelse, der styres med taster WASD. Spillet stopper hvis spøgelset fanget Pacman eller spøgelset går ind i det andet spøgelse.

Spillet består af en Pacman, et spillerstyret spøgelse og et spøgelse, der følger efter Pacman. Selve banen består af 10x10 feltet, med mindre spilleren taster andet ind før spilstart. Spillepladen genereres tilfældigt og består af piller, bomber og to slags vægge – en nedbrydelig og en ubrydelig. Når Pacman går ind i en pille, tælles scoren én op, og går han ind i en bombe, springes de nærliggende felter i stykker. Bomber fjerner både mad, andre bomber og nedbrydelige vægge hvor de ubrydelige vægge ikke fjernes. Spillet slutter når Pacman bliver fanget af et spøgelse, spøgelserne går ind i hinanden, eller ved at lukke spillets vindue.

# Board.java

I filen Board.java laves selve spillepladen. Spillepladen er et to-dimentionelt array, der derefter repræsenteres grafisk i Pacman.java. Det to-dimensionelle array dannes og fyldes derefter ud med tilfældige heltal fra 0-4. Hver værdi repræsenterer forskellige ting på spillepladen, der repræsenteres grafisk i Pacman.java.

1. Værdien 0 giver os et blankt felt, Pacman kan gå på.
2. Ved værdien 1, tegnes den nedbrydelige væg. Den repræsenteres grafisk ved en blå rektangel
3. Pillerne tegnes i de felter med værdien 2, og illustreres som en rød cirkel.
4. Værdien 3 repræsenterer den ubrydelige væg, der ikke fjernes af bomber, og tegnes som en blå rektangel.
5. Bomberne tegnes ved værdien 4 og illustreres som hvide cirkler.

I Board.java findes funktionerne *isWall*, *isFood* og *isBomb*. Disse er tilknyttet værdierne i arrayet for hhv. vægge, piller og bomber. *isWall* er tilknyttet værdierne 1 og 3, der repræsenterer hhv. nedbrydelige- og ubrydelige vægge. *isFood* er tilknyttet pillerne, der har værdien 2 og ligeså med bomberne (*isBomb*), der er tilknyttet værdien 4.

Efterfølgende er der tre metoder, *eat*, *demolish* og *boom*. Metoden *eat* ændrer det to-dimentionelle arrays værdi til 0 i de koordinater, som Pacman befinder sig i. Ved at ændre værdien til 0, fjernes den grafiske repræsentation, der illustrerer pillen. Herefter tælles variablen *score* én op, og udskriver den, så man kan følge med i hvor mange piller Pacman har spist.

Den næste metode er *demolish*, der definerer hvilke felter, der kan springes i luften af en bombe. Den bruges i metoden *boom*, der bliver beskrevet herunder. Hvis værdien i arrayet er 1,2 eller 4 (dvs. nedbrydelige vægge, piller eller bomber), sættes arrayets værdi til 0 og de fjernes derfor fra boardet. Metoden boom bruges til at definere hvilke felter, der skal springes i luften omkring Pacman.

# PacMan.java

Først importeres de javabiblioteker, der er nødvendige for at bruge resten af metoderne i programmet. Først laves tre objekter af klassen *AffineTransformation*. De navngives *tx*, *ty* og *tz* og benyttes senere, når Pacman og spøgelserne skal bevæge sig rundt på spillepladerne. Det næste er de *Shapes*, der også bruges til Pacman og spøgelserne. De næste variabler er x- og y-koordinater til Pacman og spøgelserne samt bredde og højde på spillepladen.

Det næste i programmet, er muligheden for at bevæge Pacman med piletasterne og spøgelset med WASD-tasterne. Det foregår i den samme *switch*, der køres, når der tastes én af de valgte taster. Det fungerer på samme måde uanset om det er spøgelset eller Pacman. Når der trykkes på en bevægelsestast, kontrolleres der hvad der er i det felt, man ønsker at bevæge sig hen til. Dette gøre ved at indsætte betingelsen at det næste felt ikke må være en væg. Er det felt, man gerne vil hen i, ikke en væg kan Pacman eller spøgelset bevæge sig derhen. Efter *switchen*, translates Pacmans position til de nye koordinater, således at den grafiske repræsentation af Pacman følger med koordinaterne. Herefter kontrolleres hvad der er i dette felt. Hvis Pacman står på et felt med en pille(*isFood*) eller en bombe(*isBomb*), køres hhv. metoden *board.eat* eller *board.boom* alt afhængigt af feltet. Det bevægelige spøgelse translates ligesom Pacman, men her udelades *board.eat* og *board.boom*, da spøgelset jo ikke skal ’spise’ pillerne eller aktivere bomberne.

For spøgelset, der følger efter Pacman, er det dog stillet en smule anderledes op. Her sammenlignes Pacmans koordinater med spøgelsets og spøgelset bevæger sig afhængigt af Pacman. Der er opstillet samme forudsætninger som med de to spillerstyrede figurer, så før at spøgelset kan flytte sig, kontrolleres der om det felt, spøgelset vil ind i, er en væg. Er det ikke en væg, flyttes spøgelset og spøgelsets *shape* translates til den nye position. Metoden *repaint* kalder *paintComponent*, der tegner spillepladen igen, men nu med de nye ændringer. Dette kan eksempelvis være Pacman og/eller spøgelsernes nye position, hvis Pacman har spist en pille eller han har trådt på en bombe. Dernæst laves *Shape s* ud fra den transformerede *Shape pacman*, og *Shape s* farves gul. Det samme sker med *Shape p*, der laves fra den transformerede *Shape ghost* og *Shape m*, der laves fra *ghost2*.

I PacMans main metode, laves først vinduet spillet finder sted i og navgiver det Pacman. Dette kan læses i vinduets top, og at spillepladen skal laves ud fra Board board, der kommer fra Board.java. Den næste metode giver mulighed for at ændre størrelsen på spillepladen før spilstart. Hvis der indtastes en værdi laves spillepladen board ud fra den indtastede værdi. Indtastes der ikke noget, laves spillepladen ud fra den indtastede værdi 10. Dette gør pladen til 10x10 felter. PacMan tilføjes til spillepladen *board*, og vinduets størrelse afhænger af spillepladens størrelse.