19. marts 2021

Jeanette Nord Aa

https://github.com/NordAa/Pacman.git

Erhvervs Akademi Aarhus

Pacman

Mobil Application Developement Android

Indhold

[Introduktion 2](#_Toc67081352)

[Kontrol knapper 2](#_Toc67081353)

[Bevægelse 3](#_Toc67081354)

[CollisionCheck 3](#_Toc67081355)

[Nyt spil 5](#_Toc67081356)

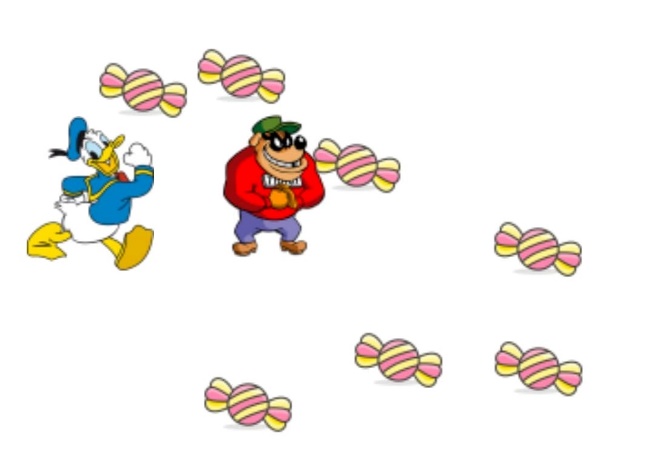
[Timer 5](#_Toc67081357)

[Ekstra features 6](#_Toc67081358)

[Testning af spillet 7](#_Toc67081359)

[Konklusion 7](#_Toc67081360)

The PacMan Game on Android



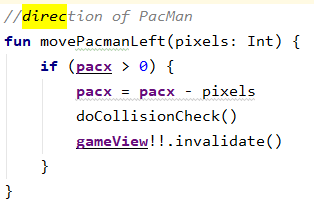
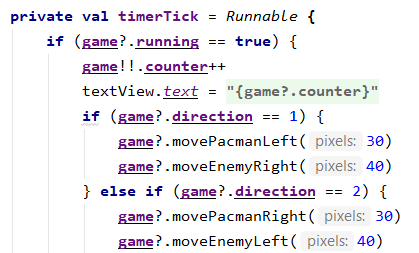
# Introduktion

Jeg har igennem Android studio brugt kotlin til at udvikle et spil til Android enheder, baseret på det kendte spil PacMan. I min forsimplede version, er det i stedet Anders And på jagt efter slik, og Bjørne Banden der prøver at fange ham. Spillet er bygget op af 5 classes, der hver indeholder de forskellige funktioner, variabler og metoder som sammen får spillet at virke. I spillet har jeg tilføjet navigationsknapper til at styre i hvilken retning Anders skal bevæge sig, venstre, højre, op og ned. Samler han et stykke slik op, kommer der en toast til syne hvor der står ”Haps!”. Hvis han derimod rammer en bandit, kommer beskeden ”Got you!” og spillet starter forfra. På selve spillefladen er Anders And, 12 stykker slik og et antal banditter svarende til det level man spiller. Øverst ser man en menubar med spillets navn, en knap til at gå til et nyt spil og under de tre prikker ligger indstillinger. Ved hjælp af de 3 knapper øverst er det muligt at starte spillet, sætte det på pause og resette det. Ved siden af knapperne tælles point for hvert enkelt spil og total points angiver den samlede point scorer igennem de levels du når. For at det er tydeligt hvilket level man har nået, er der også indsat en level-tæller. For at holde styr på længden af spillet, og om spillet er i gang, er der indsat to timere - den ene tæller op og viser at spillet er i gang, imens den anden tæller ned fra 30. Når denne timer nul inden alt slikket er samlet op, er det GAME OVER!!

# Kontrol knapper

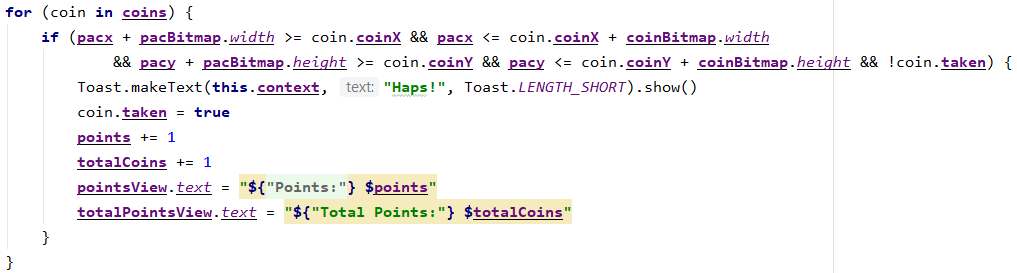
Da jeg begyndte med et allerede påbegyndt projekt, var første skridt i spillets opbyggelse at tilføje kontrol elementer til at bestemme i hvilken retning Anders And skulle bevæge sig. Jeg valgte at indsætte knapper som jeg tilføjede inde i activity\_main.xml. Her er alle spillets elementer sat op i et RelativeLayout. For at få knapperne i bunden af skærmer og til at stå på linje, tilføjede jeg dem i et LinearLayout øverst i xml-filen. Til højre vises et eksempel på den ene af disse knapper. Da alle 4 knapper var tilføjet, sørgede jeg for at oprette hver af dem som en string, så de snere kunne tilføjes i koden.

# Bevægelse

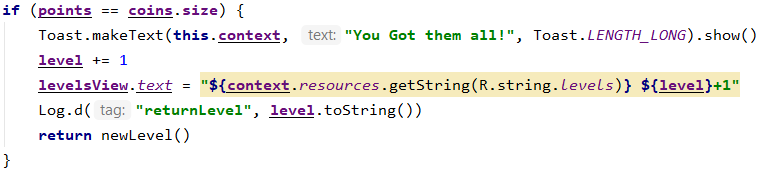
Til at starte med rykkede Anders And sig kun når man trykkede på en af knapperne, så ville man have ham til at gå hele vejen over på tværs af skærmen måtte man blive ved at klikke til man nåede sit mål. Dette blev sat op ved hjælp af onClickListeners på hver af de fire retnings-knapper. Men senere opdaterede jeg denne del, sådan at han bliver ved med at bevæge sig i en retning indtil han får andet at vide. Jeg tilføjede en ny variable inde i game classen: var direction = 0 og oprettede en funktion for hver retning (se eksempelet til højre) som ændrer værdien af pacx og pacy. Her er pacx værdien Anders Ands placering på den horisontale linje og pacy er værdien for placeringen på den vertikale linje. Derudover sørger vi i funktionen også for at holde Anders inden for rammerne af gameviewet. For at sætte Anders i bevægelse, er der tilføjet en timer til koden. Timeren tæller (running == true) fra at Anders starter med at løbe og indtil han har samlet alle mønter, eller rammes af en bandit. Derudover så forbinder den direction værdierne med funktionerne for de angivne retninger.

# CollisionCheck

Slikket er placeret random på skærmen i hvert spil og billedfilen er oprettet som Bitmap inde i game. Det samme gælder banditterne fra Bjørne Banden. Begge dele er kald ind i game filen fra deres egen class igennem ArrayList. Derefter har jeg modificeret onDraw således at den kun viser slikket, hvis ikke Anders allerede har taget det. Derfor har jeg inde i GoldCoin classen oprettet en variabel taken, som er sat til true. Slikket vises så kun når taken = false. Til at tjekke om hver enkelt stykke slik er samlet op, har jeg oprettet et loop som sørger for at tjekke hvert enkelt stykke:

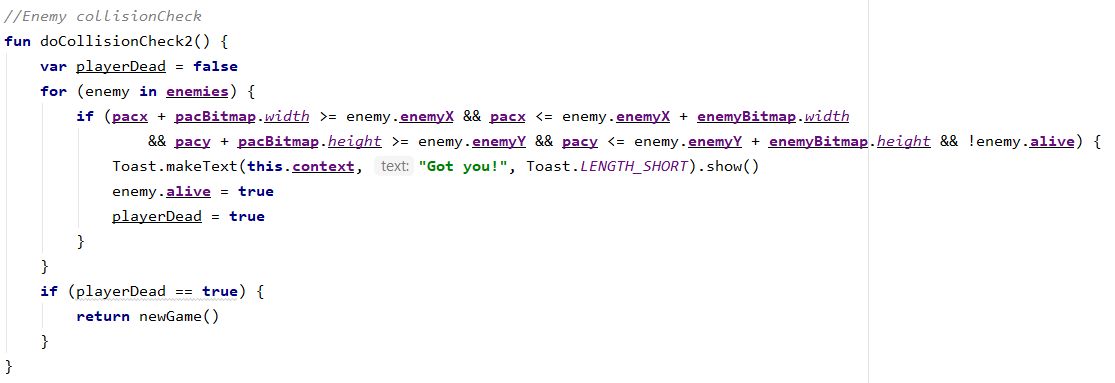


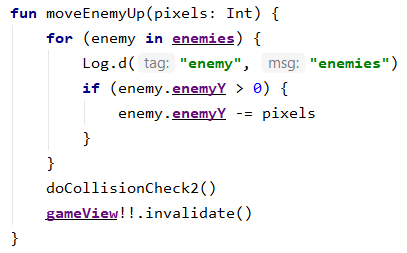
I loopet har jeg angivet en Toast der siger ”Haps!” for hvert stykke slik Anders samler op, samtidig med at spilleren skal have 1 ekstra point for hvert stykke. Når alle stykker så er samlet op, så hopper den videre til et nyt level:



Det er vigtigt at denne del er udenfor loopen, da det ellers kan være skyld i at spillet starter et nyt level hver gang et stykke slik samles op.

Nogen lunde det samme er gældende for banditterne:

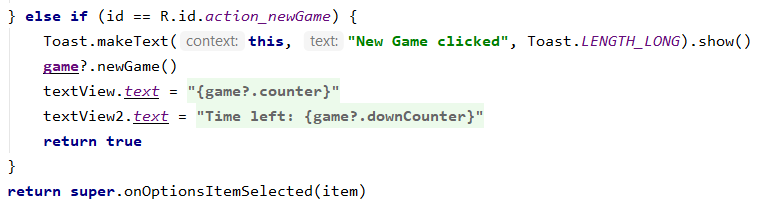


Her gør det sig bare gældende at når Anders støder sammen med dem, så forsvinder de ikke, i stedet ”dør” Anders og der startes et nyt spil. For at dette kunne lade sig gøre, var jeg nødt til at lave en ny variabel: playerDead = false, som ændrer sig til true, når loopen registrerer at en bandit fanger Anders. Derefter fortæller vi den, at hvis den registrer at Anders er død, så skal den returnere et nyt spil. Jeg fandt derefter ud af at jeg havde sat funktionen for mine bandits bevægelse forkert op, da jeg til at starte med havde placeret doCollisionCheck2() inde i loopen. Det betød at appen kunne finde på at crashe, da den i sin loop over bevægelsen også tjekkede for sammenstød med Anders. De blev derfor rykket udenfor loopen.

Min tanke var at lave en hjælpe funktion, der gjorde at slikket først blev samlet op når Anders rørte helt ved dem, men det nåede jeg ikke til, og derfor er den nogen gange lidt hurtig til at opsamle slikket, da de bare er de yderste at Anders’ Bitmap og det yderste af slikkets Bitmap der skal røre hinanden. Det mest optimale havde været at centrere Kollisions-punktet.

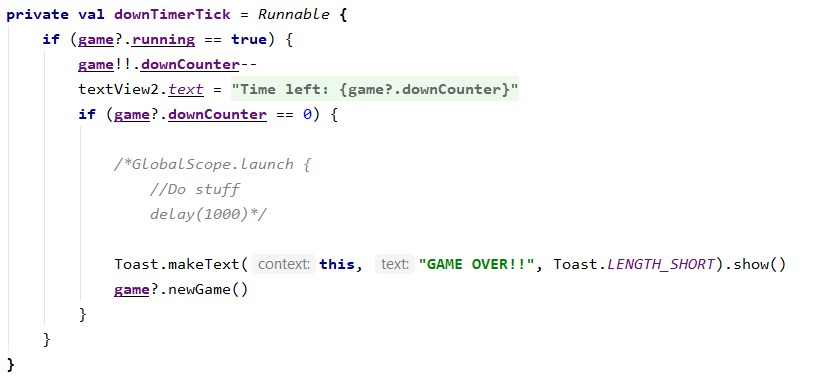
# Nyt spil

Til at starte med, havde newGame() en masse værdier inden i sig, men de fleste af disse blev flyttet over til newLevel, for at jeg bedre kunne kontrollere hvad der blev reset hvornår. Derfor ser min newGame() ud som til højre. Her fortæller jeg den at når et nyt spil retuneres, så skal totalCoins igen være = 0 og det samme gælder level. Derudover fortæller jeg den også at den samtidig skal tage værdierne fra newLevel funktionen. Hermed bliver alt reset når newGame retuneres, hvilket den bland andet gør når Anders bliver fanget af Bjørne Banden, når tiden løber ud, eller når man klikker New Game i menubaren.



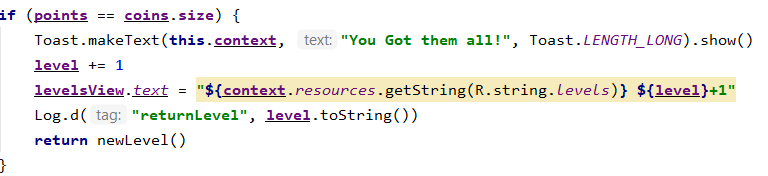
# Timer

Jeg har allerede kort været inde på hvordan den ene timer fungere, men da jeg har implementeret to forskellige, er det den anden jeg vil fokusere på her. Udover timeren, har jeg nemlig også implementeret en downTimer metode. Denne sætter en nedtælling i gang når der trykkes play og running == true, og spilleren har så 30 sec. til at gennemføre spillets level og gå videre til næste level. Hvis ikke man når det kommer en toast med beskeden ”GAME OVER!!” og newGame funktionen kaldes sådan at spilleren skal starte forfra. En del af timernes funktion er, at de kan pauses, hvis spilleren trykker på pause knappen, og de kan resettes hvis der trykkes reset. Timerne er også begge sat til at stoppe, hvis spillet f.eks. lukkes for at læse en besked.

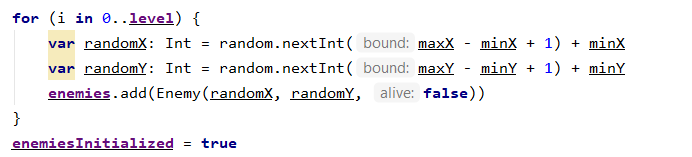


# Ekstra features

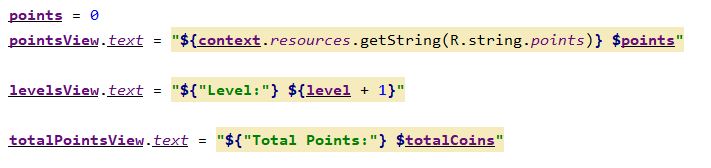
Som ekstra features har jeg i mit spil valgt at tilføje levels og en samlet scorer af point op gennem de levels man når. Næste skridt ville være at tilføje highscore, og gøre det muligt at dele den med sine venner. Men det nåede jeg ikke lige i denne omgang. Mine levels er oprettet ved var level = 0, og så er der tilføjet et textView som viser værdien af level + 1, da den ellers starter på 0. Dette kaldes inde i functionen newLevel(). Derudover har jeg returneret funktionen inde i doCollisionCheck, for at sikre mig at spilleren stiger i sværhedsgrad hver gang det lykkes at samle alt slikket op inden tiden løber ud, og uden at blive fanget af Bjørne Banden. For at øge sværhedsgraden for hvert level, har jeg også sørget for øge værdien for level:



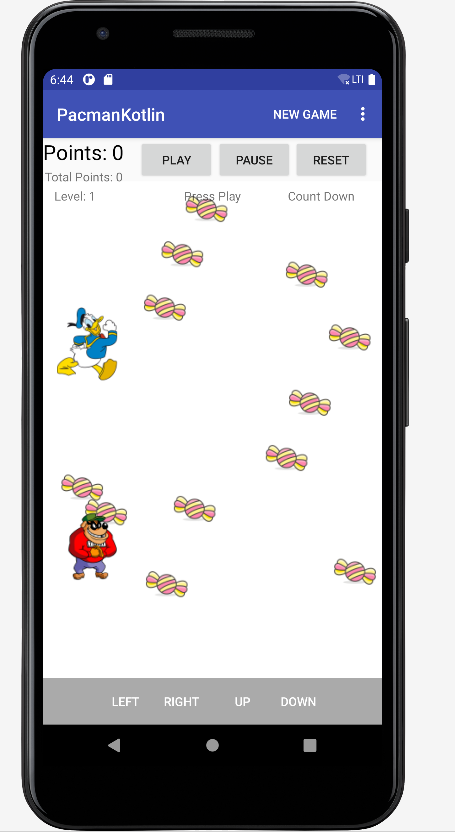
Inde i min loop for banditterne, har jeg sørget for at øge antallet af banditter, så det svarer til det level spilleren er i. Det har jeg gjort ved at skrive: (i in 0..level) - derved tager den det givne level og returnerer antal banditter ud fra det:



For at vise den sammenlagte sum af point op gennem de forskellige levels, så har jeg oprettet endnu en variabel: var totalCoins = 0. Her har jeg også oprettet et textView til, hvori summen af de sammenlagte point viser sig. TotalCoins er sat op på præcis samme måde som point - den eneste forskel er, at når funktionen newLevel() kaldes, så sættes point = 0 og det gør totalPoints ikke før der retuneres et nyt spil.



# Testning af spillet

I løbet af spillets udvikling er det blevet testen ved hjælp af Android Studios Emulator med API 30, og via USB til min Huawei p30 pro. Det at kunne teste løbende, har gjort det muligt at løse mange af spillets problemer undervejs. Testene har bland andet hjulpet mig til at ændre rækkefølgen på hvordan jeg havde sat min kode op som f.eks. i doCollisionCheck. Jeg er også blevet gjort opmærksom, hvis mine mønter eller banditters værdi var sat forkert, for så skete der enten ingenting når Anders rørte dem, eller også viste de sig slet ikke på spillefladen.   
Testning har også givet indblik i om funktionaliteten af spillets elementer fungerede som de skulle og var forståelige for andre end mig selv. Jeg har bland andet ladet min mand spille spillet på min telefon, og jeg har send mit spil som apk-fil til Ruth, som testede den på sin samsung - SM-G973 uden problemer. Jeg har dog selv oplevet mindre problemer, men jeg oplever at det kan være forskelligt fra gang til gang når jeg prøver appen, og derfor er det ikke lykkes mig at finde fejlen og få det udbedret. En af de fejl jeg har oplevet, er at spillet kan finde på at resette til nyt spil, og ikke gå videre til næste level, selvom alt slikket bliver samlet op og man ikke er stødt ind i en af Bjørne Bandens medlemmer.

# Konklusion

Der er altså rigtig mange elementer der skal spille sammen for at udvikle en app som denne forsimplede version af PacMan. Det er helt klart gået op for mig hvor vigtigt det er at få placeret de forskellige elementer de rigtige steder og kaldt dem på de rigtige tidspunkter. Jeg har været nødt til at holde tungen lige i munden og spørge om hjælp undervejs for at få tingene sat rigtigt sammen. Samtidig føler jeg også at min forståelse har udviklet sig, og at jeg er begyndt at få overblik over de basale opbygninger af classes, functions og loops. Dette spil har helt bestemt udfordret mig, men jeg er også sikker på at det har udviklet mig til at have mere mod på de næste opgaver indenfor android programmering.   
Det har været en lærerig proces, hvor jeg specielt har følt mig udfordret af min tålmodighed med programmet/min computer og på det punkt hvor jeg oplevede at appen ikke altid fungerede på samme måde, på trods af at intet nyt var ændret.