

DOKUMENTASJON OBLIG3

Android app, utviklet av: Anders Simonsen, s198739



GREEN BASTARD GAME APP

FORORD

Dette er dokumentasjonen for mappeoppgave 3 i faget applikasjonsutvikling ved Høgskolen i Oslo og Akershus høsten 2015.

Dokumentet legger fram bakgrunn for produkt, rammebetingelser, verktøy og tilleggsfunksjonalitet.

Dokumentasjonen er beregnet for faglærer og sensor.

Litt om det tekniske bak appen og testing:

Minimum API er satt til 16, og target API til 22, jeg har ikke testet på alle API'ene imellom.

Jeg har kjørt tester på emulatoren med API 19,

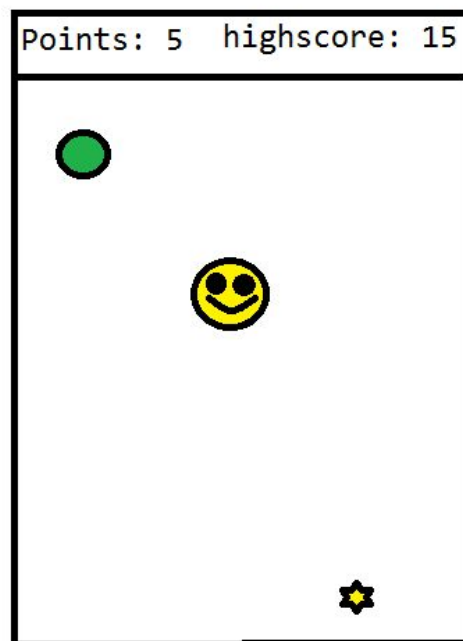
Jeg har bare lagt inn engelsk språk.

Appen er låst i portrait mode, da jeg ikke følte det var gunstig å tillate landscape.

OPPGAVEBESKRIVELSE

Tenker å lage et spill hvor man styrer en liten kule ved å tappe på skjermen for å få den til å skifte retning, kulas fart skal være konstant og den skal sprette når den treffer kanten av skjermen.

Man må unngå å bli truffet av en annen kule(fienden), som har høyere fart. Man sanker poeng ved å sanke små objekter mens man unngår den store kula. Det lagres highscoreliste.



RAMMEBETINGELSE

- Android platformen
- XML
- SQLite

UTVIKLERVERKTØY

Android Studio IDE

PROSJEKTET

Til å begynne med hadde jeg “Roll a Ball” som working title. Dette er nå endret til “Green Bastard”, en referanse til karakteren Bubble’s sitt wrestling-alias i TV-serien “Trailerpark boys”.



ARBEIDET

Jeg begynte med å lage et mini-framework av ting jeg trengte for å mekke appen, da java på android ikke er fullt så komplett som java for pc.

Jeg begynte med å lage mitt eget “framework” med generiske klasser og metoder.

Deretter gikk jeg løs på å finne ut hvordan jeg kunne tegne og animere spillgrafikk.

Oppgaven viste seg å være mer tidkrevende enn jeg først tenkte, og jeg har i perioder sittet oppe flere netter og programmert.

Package MyFramework består av

- BoxCollider-klassen definerer et rektangel og inneholder metode for å finne midten av rektangelet, og metode for å sjekke om rektangelet kolliderer med et annet rektangel, samt metode for å oppdatere rektangelet.
- CircleCollider-klassen tilsvarer BoxCollider-klassen, bare at den definerer det samme for en sirkel.
- MoveableObject Denne klassen var tenkt som en base for bevegelige objekter, til animasjon etc. Klassen har sin egen boxCollider, og circleCollider, valgfritt hvilken man vil bruke. Har også metoder for å sette posisjon(x,y) og vektorer for kraft på X og Y akse, samt metoder som beregner bevegelse utifra objektets retningskraft og generelle fart.
- Vector2D denne klassen er litt feil definert, da den bare er et subset av en ordentlig 2D-vector.

Matematikken bak CircleColliders

Jeg brukte en del tid på å finne ut hvordan jeg skulle beregne kollisjoner mellom to sirkler eller CircleCollidere, men matematikken viste seg å være relativt enkel:

La oss tenke oss at vi har to ulike sirkler.
Hver sirkel har et senter definert som et punkt(x,y), og en radius r.
Hvis vi drar en linje mellom disse 2 punktene får vi avstanden mellom de 2 punktene.
Vi beregner avstanden mellom de to punktene slik, vi navngir den L.

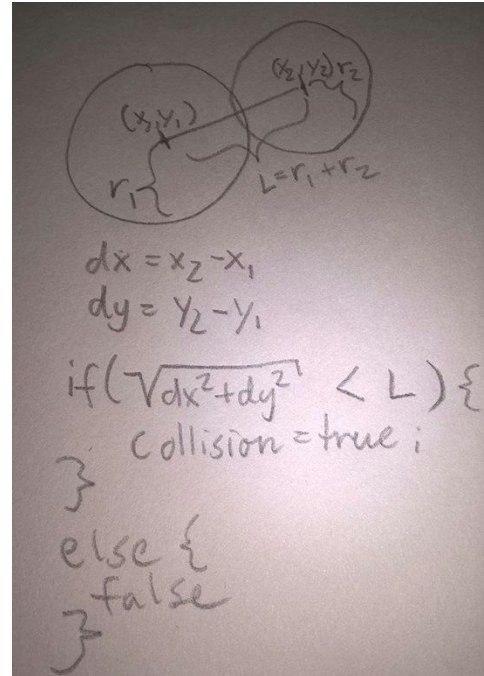
$$\begin{aligned} dx &= x_2 - x_1 \\ dy &= y_2 - y_1 \\ L &= \sqrt{dx^2 + dy^2} \end{aligned}$$

Den minste avstanden vi kan ha mellom de to sirklene er summen av de to sirklens radius.

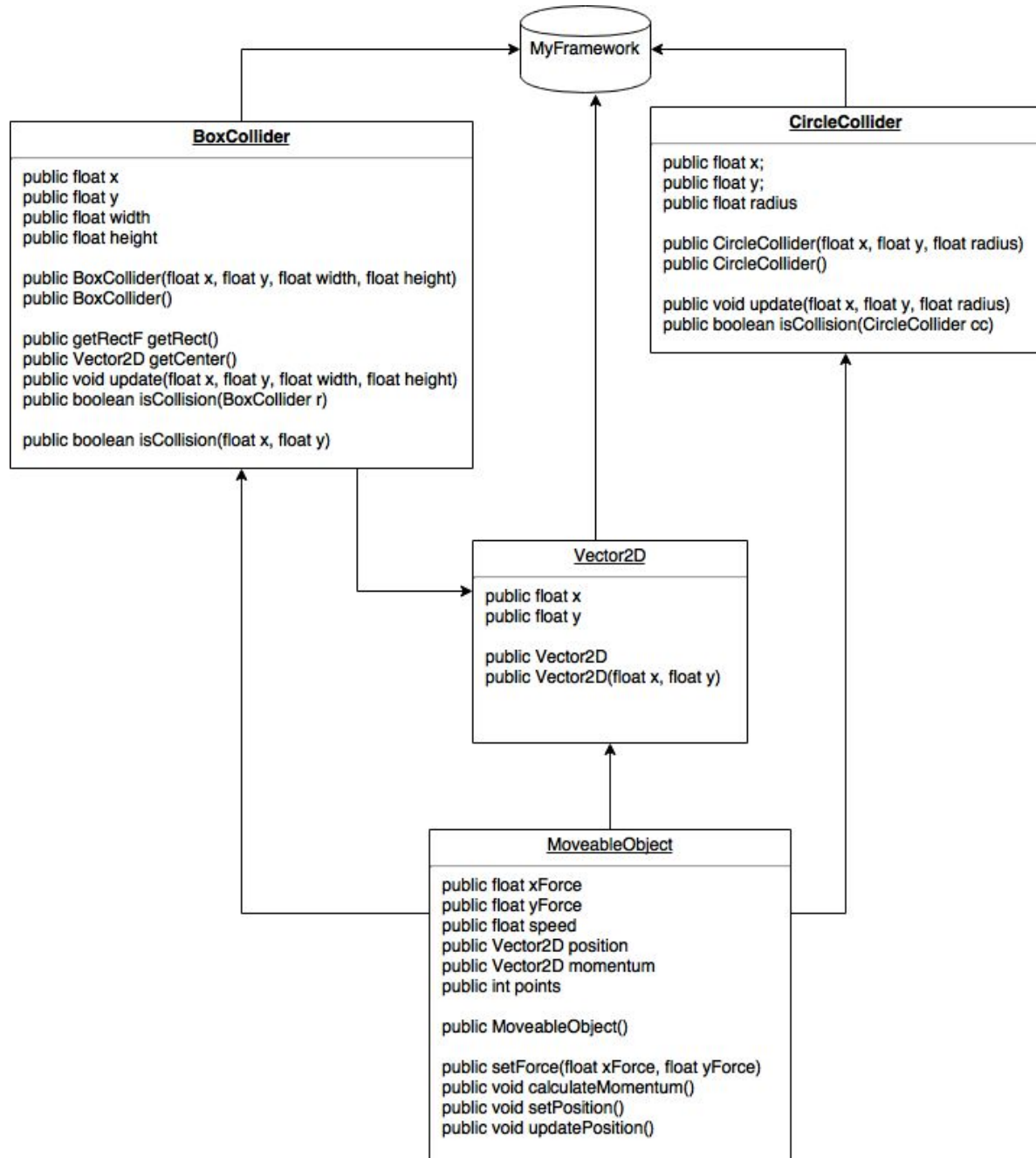
$$L_{min} = r_2 + r_1$$

Dersom linjen L er mindre enn Lmin har vi en kollisjon.

$$L < L_{min} = \text{kollisjon}$$



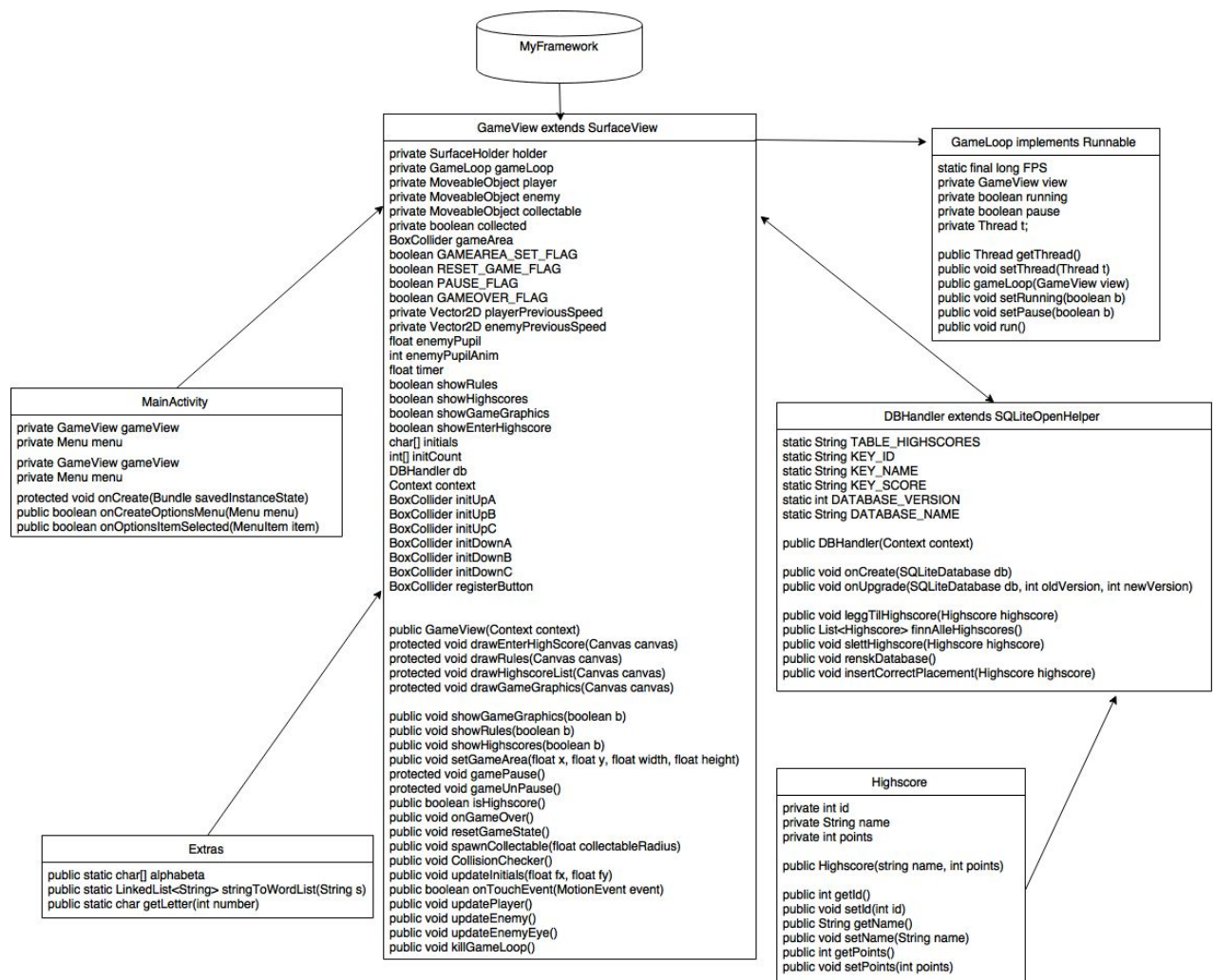
Detaljert bilde av frameworken



APPEN

Appen inneholder kun en aktivitet MainActivity som benytter seg av GameView som arver fra SurfaceView. GameViewet inneholder diverse draw metoder for å tegne på surfaceViewet sitt Canvas, samt metoder for å oppdatere spillobjekter, samt ta touch input for å oppdatere spillerens posisjon og kontrollere enkelte menyvalg. GameLoopen inneholder en tråd som sjekker et utvalg av booleans og velger hvilken draw metode som skal tegnes.

Detaljert bilde av appen



DESIGN

Av Googles design prinsipper har jeg brukt :

“Simplify my life”

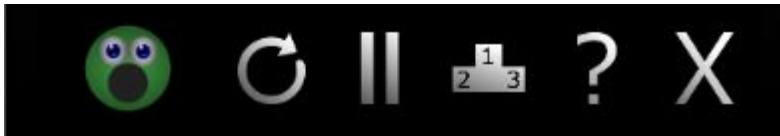
- **Keep it brief:** Jeg bruker korte fraser og enkle ord.
- **Pictures are faster than words:** Jeg benytter meg av enkle former og kjente og kjære symboler for å gjøre det enkelt for brukeren.
- **Only show what I need when I need it:** Viser til enhver tid bare det brukeren trenger å se.

Jeg har også tildels benyttet meg av **“Enchant me”** da appen er interaktiv og lar deg røre og manipulere spiller objektet.

Logoen: Jeg benyttet jeg meg tildels av Googles color-design style-guide. Logoen er en mer detaljert versjon av spillfiguren som benyttes i spillet.

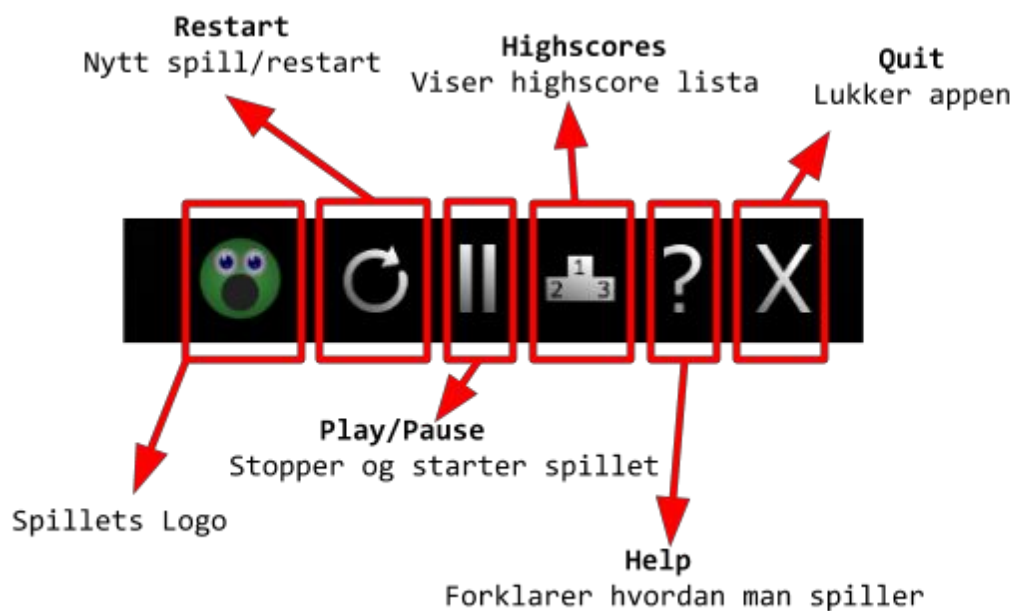


Action-baren



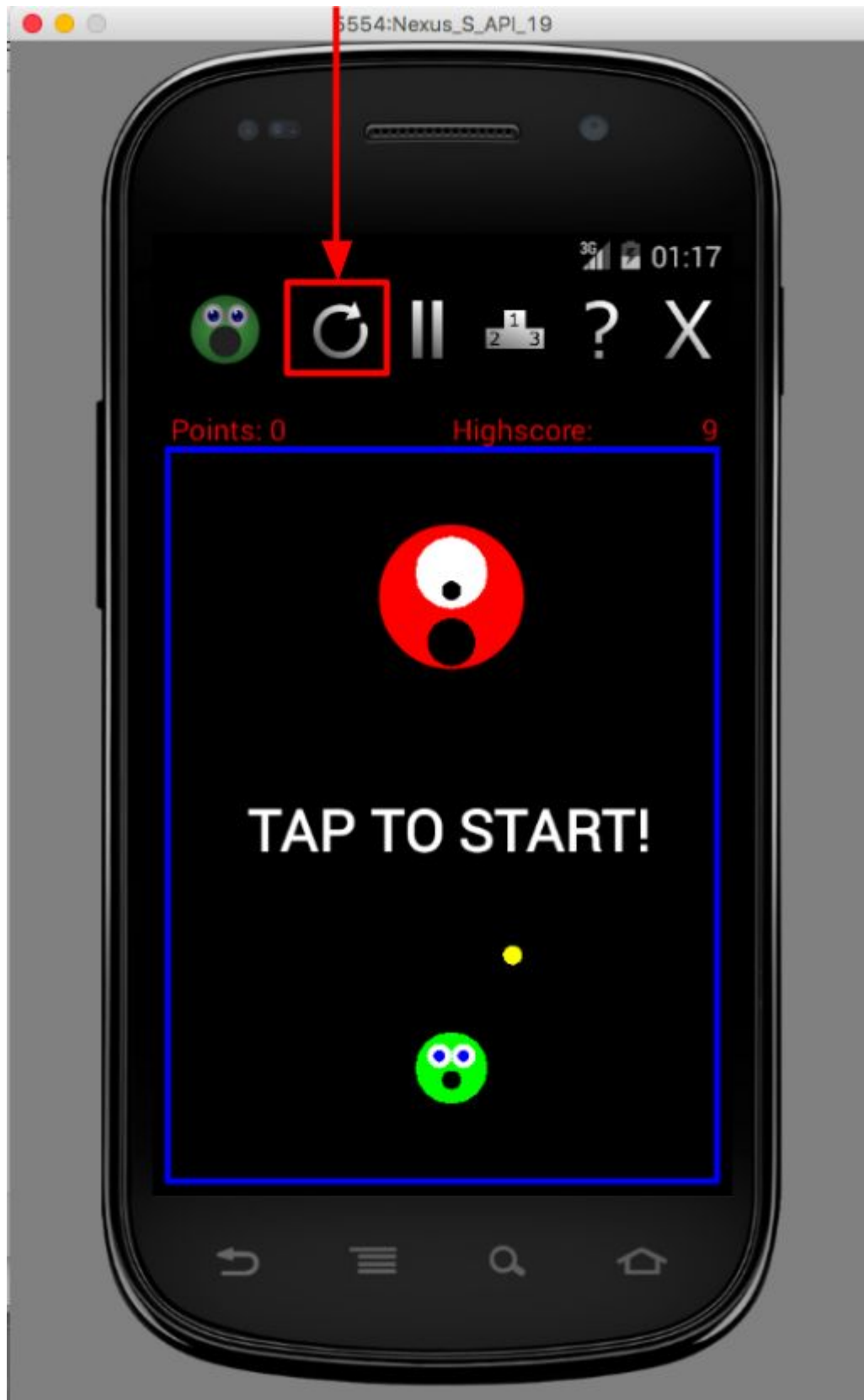
Ikonene til menyknappene i Action-Baren har jeg designet i Inkscape, jeg har prøvd å gjøre dem enkle og selvforklarende ved å ta i bruk kjente symboler.

Forklaring av Action-baren:



Ellers er det et gjennomført retro-arkade preg på appen. Alt ellers er tegnet i GameView klassen og benytter seg av kun 5 farger: black, red, blue, yellow, white. Dette for å få mest mulig enkelt og rent utseende på appen.

Skjermbilde 1(Restart) Når du starter appen er det dette bildet som møter deg. Trykk på skjermen for å starte spillet.



Skjermbilde 2 Når du trykker på skjermen, starter spillet, og du må bevege deg rundt i tråd med spillets regler.



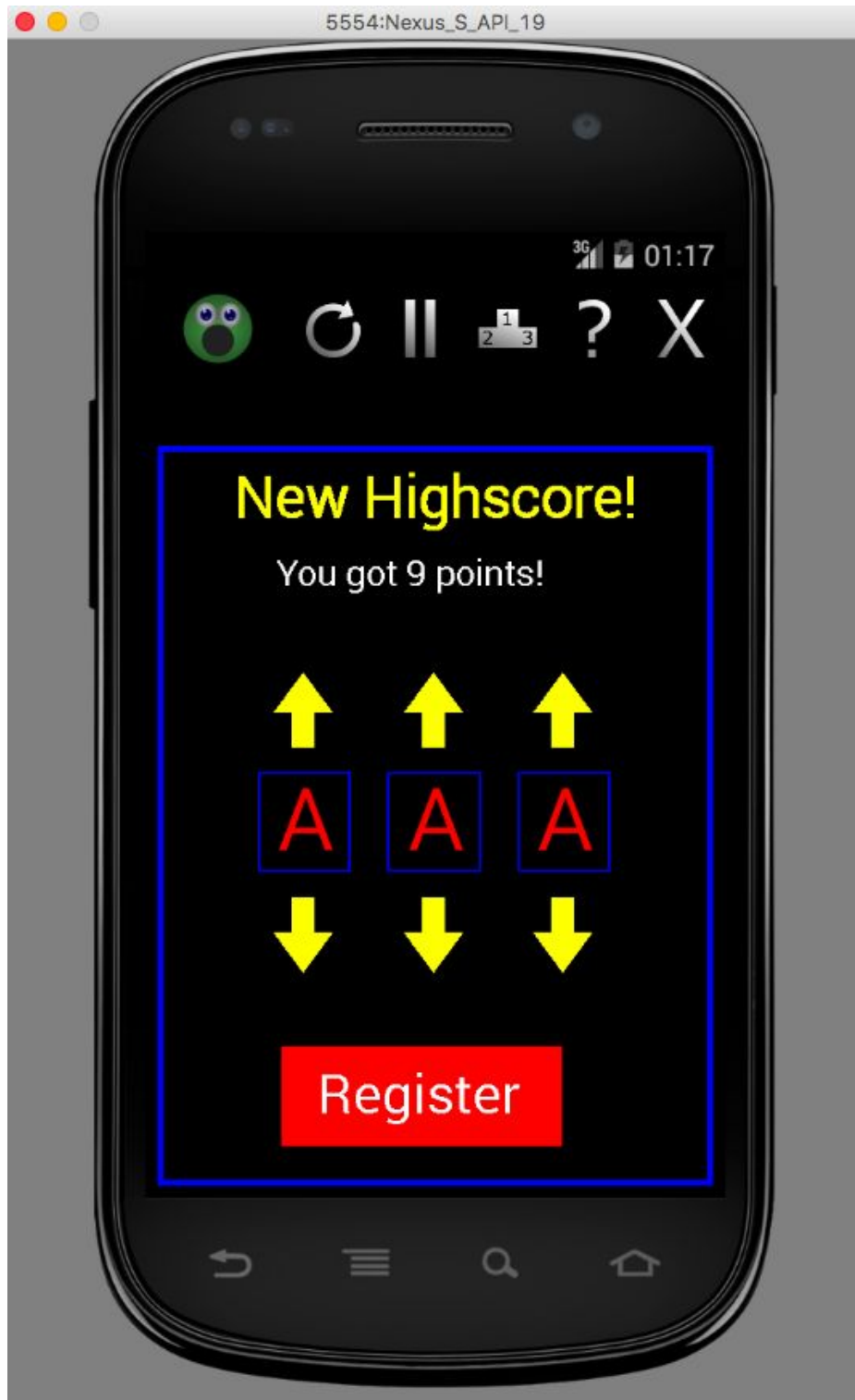
Skjermbilde 3(Help) Regler, her forklares spilletts regler og styringsmekanisme.



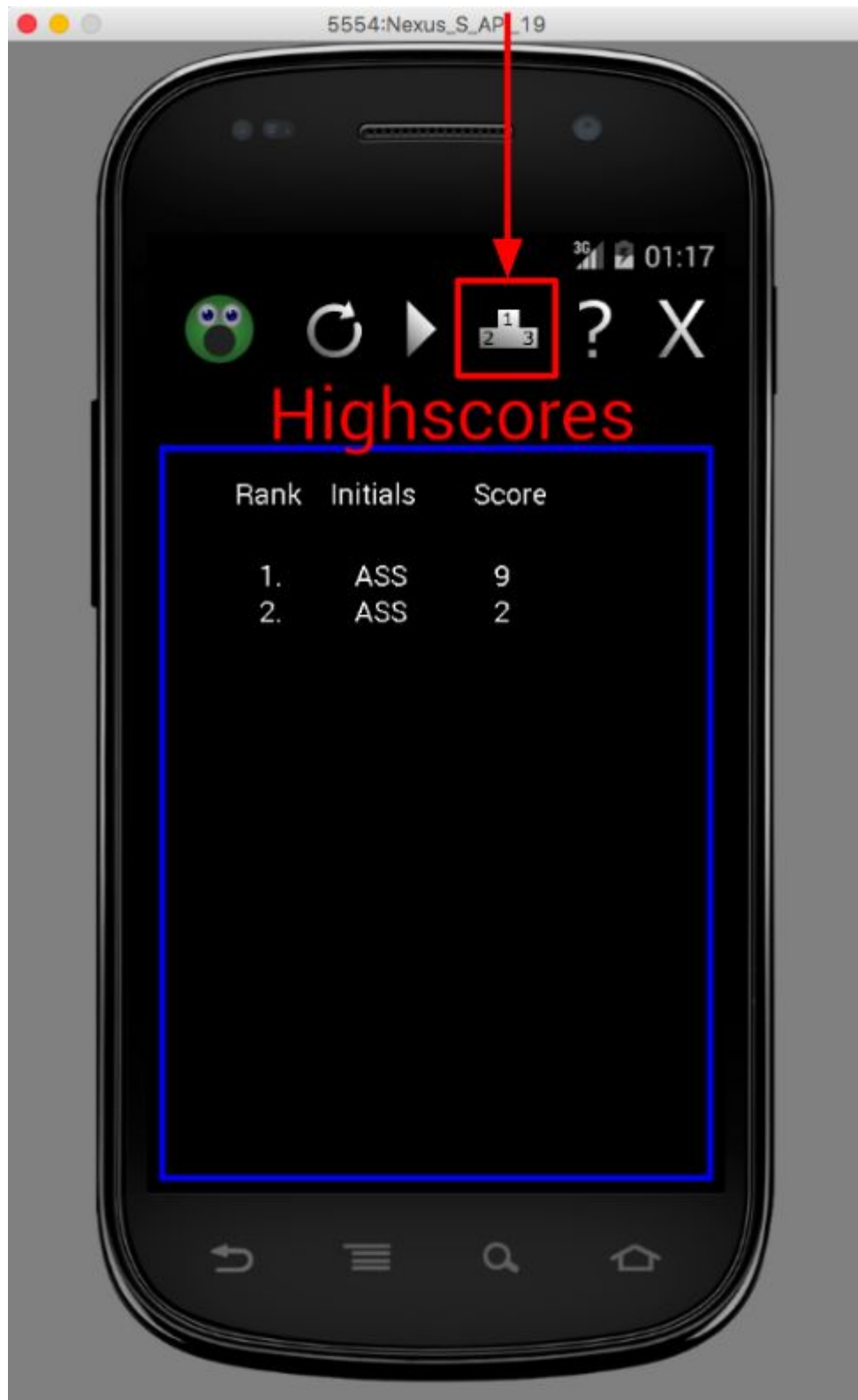
Skjermbilde 5(Game over) Blir du tatt av den røde bastarden er spillet over og “GAME OVER!” klistres i blått over midten av skjermen.



Skjermbilde 6(EnterHighscore) dersom du har slått den tidligere highscoren ved Game Over, tas du videre til denne skjermen. her kan du legge inn initialene som vises ved din highscore i highscorelisten. Disse endrer du ved å trykke på de gule pilene. Den røde registreringsknappen trykkes på for å lagre highscoren. Ved dette skjermbildet er knappene i actionbaren deaktivert for å hindre at du mister highscoren din.



Skjermbilde 3(Highscores) Highscorelista, her vises de tidligere ti beste rekordene tatt i spillet.



REFLEKSJON

Dette er første oblige jeg føler jeg er 98% fornøyd med resultatet på. Det er små ting som kan forbedres, slik som animasjonene, samt optimalisering av koden, samt at jeg mistenker at det er en bug et sted, men jeg har ikke rukket å teste nok til å finne ut hvorfor den oppstår, den oppstår også veldig sjelden, tror det kan være at emulatoren går tom for minne.

KILDER

<http://cgp.wikidot.com/circle-to-circle-collision-detection>

<https://www.linux.com/learn/tutorials/707993-how-to-draw-2d-object-in-android-with-a-canvas>

<http://developer.classpath.org/doc/java/awt/Rectangle-source.html>

<https://www.google.com/design/spec/style/color.html#color-color-palette>

<http://developer.android.com/design/get-started/principles.html>