

DANMARKS TEKNISKE UNIVERSITET



02128 - SOFTWAREPROJEKT

COUNT DRACULA



GRUPPE 7

WICTOR LANG JENSEN	s184197
ERIC KRISTIAN GYLLING	s173896
NICKLAS OLIVER ASKJÆR	s184199
MARTIN CHRISTOFFERSEN	s184218
ANJA MYLORD KAMPMANN	s180070
MATHIAS CHRISTOFFERSEN RIIS	s173914

AFLEVERINGSDATO

25. JUNI 2019

Indhold

Introduktion	2
Hvad er Count Dracula?	2
Arbejdsfordeling	2
Analyse	3
Design	3
Implementering	4
Firebase	4
Spillet starter	5
Create Game	5
Fordeling af roller	5
Under spillet	6
Valgrunde	6
Lovgivende runde	6
Spillet slutter	6
Test af appen	7
Konklusion	7
Litteraturlise	8
Bilag	9
Oversigt	9
Spilregler	9
Roller	9
Mål	9
Set-Up	9
Spillets gang	9
MoSCoW	12
Must have	12
Should have	12
Could have	12
Won't have	12

Introduktion

Hvad er Count Dracula?

Count Dracula er et lov-baseret rollespil. Man bliver inddelt i to hold, vampyrer og bønder. Det gælder for vampyrene om at få indført vampyrlove, som gavner dem, men det gælder også for dem at holde deres identitet hemmelig fra bønderne. For bønderne går det ud på at få indført nok anti-vampyrlove, eller får dræbt vampyrernes leder Dracula. Spillet går ud på at overbevise de andre om, at man ikke er vampyr, men samtidig få indført de nødvendige love der skal til for at vinde spillet. Dette kan gøres ved at lyve og anklage andre spillere. Man skal derfor snakke med hinanden igennem spillet.

Meningen med appen er, at den skal erstatte brætspillet. På denne måde kan man altid spille Count Dracula med sine venner, hvor end man er, bare ved at have sine venner, sin mobil og downloade appen.

Arbejdsfordeling

Da arbejdet startede fik vi hurtigt fordelt roller i gruppen, der var nogen som fokuserede på at få Firebase op og kører samt logikken bag spillet, og andre som fokuserede på grafik og layout. Vi har dog under hele forløbet arbejdet sammen som gruppe, og hjulpet hinanden hvor der var mest brug for det.

Hovedkomponent	Logik	Grafik
MainActivity	MATHIAS	ERIK
WaitingRoom	MATHIAS	ERIK
GameOverview	WICTOR	NICKLAS
Endgame	ANJA	ANJA
Rulebook	MARTIN	MARTIN

Analyse

For at appen kan fungere som et spil, er der mange funktioner som er nødvendige at få implementeret. For at få en bedre forståelse for hvordan spillet fungerer, kan spilreglerne læses i bilag. Det første vi valgte at få implementeret var database og multibruger integration, som også kan ses under Must Have i MoSCow. Disse to var nødvendige, da spillet ikke kan fungere med én enkel spiller, og derfor var vi nødsaget til at have muligheden til at forbinde flere enheder til samme server. De er lavet ved brug af Firebase Realtime Database. Udover disse to funktioner, havde vi også fokus på et layout til spillet, da dette er en vigtig del af brugerens oplevelse af appen. Da vi havde styr på en database og forbindelse mellem enheder, hvilket inkluderede muligheden for at oprette og deltage i bestemte spil, kunne vi begynde at lave funktionerne for selve spillet.

Appen skal kunne fordele roller og ture til spillerne, samt holde styr på hvilken fase spillet er i. Spillet starter i nomineringsfasen, hvor Kongen skal nominere en Greve. I denne fase er det kun Kongen som har mulighed for at gøre noget. Når han har valgt hvem han vil nominere, vil alle spillere have mulighed for at stemme ja eller nej. Appen holder styr på hvor mange spillere der har stemt hvad, og vil herefter afgøre om nomineringen går igennem eller fejler. Når dette er afgjort, vil appen enten gå videre til en ny nomineringsfase eller til en lovgivende fase. I den lovgivende fase er det igen kun Kongen som har mulighed for at vælge et kort, som han vil kassere. Når dette er valgt, vil de resterende to kort blive sendt til Greven, som vælger hvilket af kortene han vil indføre. Når det er valgt, vil appen opdatere pointene for vampyrer og bønder.

Udover de nødvendige implementeringer for at spillet kan køre, har vi valgt at implementere et velkomstbillede, en regelbog og toast beskeder som både informere om begivenheder i spillet og giver fejlmeddelelser.

Vi har valgt at nedprioritere implementationen af Kongens magter og derved også forskellige spilmuligheder, ud fra hvor mange spillere der er med i spillet, da vi mener at spillet sagtens kan fungere uden disse funktioner.

I en udvidet version af spillet, kunne man implementere flere roller, så alle spillere havde forskellige ting de kunne gøre i løbet af spillet. Derudover ville vi gerne have haft muligheden for at man skulle spidde andre spillere, ved armbevægelse af telefonen.

Design

Målet for designet af app'en er, at user interface skal være så simpelt som muligt. Derfor har knapper, ændringsbare tekstfelter og ikke-redigerbare tekstfelter hvert deres design, som bruges konsekvent igennem spillet. Dette giver brugeren et bedre overblik.

Unødige muligheder for brugeren er også blevet fravalgt. Mange muligheder kan resultere i at app'en bliver svær at navigere rundt i. Derfor er unødige muligheder blevet fravalgt, så brugeren har så få muligheder som muligt.

Spillet er desuden bygget op med få tilstande, således at brugeren hurtig bliver velkendt i app'en.

Der er Menu-tilstand, WaitingRoom-tilstand, GameOverView-tilstand og EndGame-tilstand.

Brugeren kan tilføje et spil med sig selv som vært til databasen fra start menuen ved at trykke "CREATE GAME". Andre brugere kan tilmelde sig dette spil på databasen ved at skrive koden i "CODE"tekstfeltet og trykke "JOIN GAME". Brugeren bliver så ført til en lobby som er bygget op med en ting-liste, som indeholder alle spillere i spillet. Brugeren kan forlade lobbyen og gå tilbage til start menuen, ved at trykke på "LEAVE". Værten i spillet kan starte spillet ved at trykke "START GAME". Spillet bliver nu initialiseret på databasen, således at spille-rollerne bliver uddelt til deltagerne. Spillerne bliver nu ført til GameOverView-tilstanden. I GameOverView-tilstanden er der tre tabs, som der kan blive valgt. Tabbaren er placeret i bunden af skærmen, så den er lettest tilgængelig for spilleren.

Når et nyt tab bliver valgt, bliver et nyt indhold vist over tabbaren. Når de to tabs, "ACTION" og "GAME RULES", bliver valgt, vil der blive vist fragment-indhold, som udelukkende viser information til spilleren. Dette indhold kan ikke ændres og er tilgængelig igennem hele spillet. Når den midterste og sidste tab, "BOARD", bliver valgt, bliver et fragment-indhold vist, som ændres i løbet af spillet, da noget af dette fragment-indhold holder styr på scoren af spillet. Dette indhold ændres når en ny afstemning bliver afholdt, eller når en ny lov bliver valgt af regeringen. Disse ændringer bliver gemt i databasen, som værten holder styr på. Brugers telefon kan lytte efter disse ændringer i databasen via begivenhed-værdi-lyttere, som opdaterer brugers telefon, således brugeren får vist det nyeste indhold. Når en ny afstemning bliver afholdt vibrerer brugers telefon. For at få lov til at bruge telefonens vibrator, bruges tilladelsen "android.permission.VIBRATE" i manifest filen. Når spillet er ovre vil brugeren blive ført til "END GAME"-tilstanden. Her kan brugeren vælge at lukke app'en eller gå tilbage til start-menuen.

Implementering

Firestore

Firestore er den vigtigste implementation i vores app. Det er Firestore Realtime Database, som sørger for at appen kan fungere som et multiplayer spil. Firestore Realtime Database gemmer informationer i Firestore skyen, og synkroniserer disse informationer imellem de enheder som er forbundet til databasen. Dette betyder, at alle enheder forbundet til databasen, altid vil have den samme information.

Ved brug af Firestore Realtime Database, kan vi oprette et spil med en pinkode og et navn. Herefter kan andre mobiler tilslutte sig til samme spil, ved at indtaste den givne pinkode fra vært-telefonen, og et navn. Spillerne bliver herefter tilføjet til spillet i Firestore under deres valgte spillernavn. Alt information om spilleren vil blive gemt under dette navn i Firestore. Når værten af spillet vælger at starte spillet, vil der blive tildelt roller til alle spillere fra vært-telefonen. Denne information bliver så tildelt spillerne i Firestore. Herefter bliver der lavet en liste, som bestemmer hvilken spiller der starter som kongekandidat, og turene herefter.

Det meste af spillet vil blive kørt over Firestore. Firestore vil holde styr på hvis tur det er, hvornår der skal stemmes, og hvilken magt Kongen kan udføre. Derudover vil Firestore i samarbejde med vært-telefonen fortælle hvilke kort der bliver trukket, sende kort videre til Greven, hvilken lov

Greven vælger at vedtage og resultatet af afstemningen. Vært-telefonen blander kortene, samt holder styr på stemmerne fra afstemningen. Disse informationer bliver sendt videre til Firebase, som herefter deler informationen med de relevante spillere - f.eks. sender Firebase kun information til Kongen, om hvilke tre kort der er blevet trukket, eller til alle spillere efter afstemningen, så de kan se hvem der har stemt hvad.

Alt logikken på telefonen er styret med begivenhedshåndteringer. Når et element i databasen ændres, vil alle telefoner reagere. Vært-telefonen vil tælle stemmer når en spiller trykker på en knap for at stemme, alle telefoner vil tjekke om de er den nye konge osv. Appen lytter efter bestemte lag i databasen til de forskellige dele af logikken. Når der stemmes, vil den kigge i players-elementet, og kun reagere når det opdateres. Dog er der ofte behov for at tilgå data længere oppe i rækken, så de fleste funktioner lytter til spil- eller rod-elementerne.

Spillet starter

Create Game

Når der trykkes Create Game i spilmenuen, vil enheden som opretter spillet blive sat som vært-telefonen. Dette gøres ved brug af en Boolean, `isHost`, som bliver sat til `true`. Herefter bliver der genereret en tilfældig kode, som bliver lavet i metoden `randomStringGenerator`. Denne kode er navnet på Firebase serveren. Herefter bliver der på serveren oprettet et `gameSetup`, og værten bliver tilføjet til `gameSetup` i Firebase, under det angivne spillernavn. Værten bliver efterfølgende sendt videre til den næste aktivitet, `WaitingRoom`, når der trykkes på knappen Join Game.

Andre enheder kan nu tilføjes til spillet ved at indtaste et spillernavn og koden, som blev genereret da værten oprettede spillet. Enheden bliver efterfølgende sendt videre til `WaitingRoom` aktivitet. Hvis der indtastes en kode, som ikke eksisterer, vil der komme en toast, som informere spilleren om at der ikke findes en server med det angivne id.

Fordeling af roller

I `WaitingRoom` aktivitet opholder enhederne sig, indtil alle spillere har fået forbindelse til serveren. Herefter kan værten starte spillet. Nu vil der ved hjælp af et for-loop og if-statements blive tildelt roller til alle spillere. For-loopet tæller antallet af spillere, ved hjælp af `playerCounter`, så der vil blive fordelt roller indtil alle spillere har fået en rolle. Herefter bruges `Math.floor(playerCounter-1)/2` til at udregne hvor mange vampyrer dette spil skal have, ud fra antallet af spillere. Dette tal bliver gemt som `numberOfEvil`. Inden rollerne bliver fordelt, bliver spillerne tilfældigt fordelt i et array, ved hjælp af en `Randomizer` metode. Der bliver herefter brugt et if-statement som fordeler vampyrroller, når en vampyrrolle er blevet fordelt bruges `numberOfEvil`-, dette if-statement kører så længe `numberOfEvil` er større end 0. Den første vampyr der bliver valgt, er også valgt til Dracula, da endnu et if-statement kører så længe `i = 0`. Når `numberOfEvil = 0`, og alle vampyrrollerne er fordelt, vil de resterende spillere blive tildelt rollen som bonde. Når alle har fået tildelt en rolle vil spillet gå videre til `GameStart` aktiviteten.

Under spillet

Valgrunde

Den første fase går ud på at kongen, som er besluttet på firebase, bliver bedt om at nominere en greve. Firebase sender her en besked ud om at nomineringsfasen er startet, og hver spillertelefon sammenligner kongens navn med deres eget. Hvis de er kongen vil de få vist en liste over spillere som de kan nominere til greve. Når dette sker, vil det ændre countNominee i firebase, og spilfasen ændres til playerVote.

I playerVote fasen får alle spillere en besked om hvem den nominerede greve er. Der vil desuden dukke en ja- og nej-knap op, som sender deres stemme til firebase under spilleren. Et ja giver 1 point, og et nej giver -1 point. Vært-telefonen tæller stemmer sammen hver gang en af disse værdier ændres på Firebase, og hvis alle har stemt afgør den om summen af stemmer er over eller lig 0. Hvis dette er tilfældet, bliver kongen og greven valgt ind. Hvis ikke, bliver turn-variablen ændret, og den næste spiller får kongemagten.

Lovgivende runde

Hvis valget lykkedes, ændres spilfasen til kingChoose. Her vil vært-telefonen trække 3 kort fra kortbunken og sende dem til Firebase. Telefonen tilhørende konge-spilleren vil så hente de tre kort ned, og vælge et kort som skal fjernes. Dette kort bytter om nødvendigt plads med det tredje kort i databasen, og de første 2 kort sendes videre til greven. Igen vil greven vælge et af kortene, og et point vil blive tildelt det hold som kortet repræsenterer. Herefter går turen videre, den næste konge afgøres og spillet går tilbage til nomineringsfasen.

Knapperne genereres semi-dynamisk. De får alle den samme eventhandler, og kun teksten på knappen ændres af kortbunken på serveren. Eventhandleren for kongen ved bare hvilken knap der er trykket på, og bytter det tilsvarende kort med det tredje af de tildelte kort.

Grevens knapper læser teksten på kortet, og afgør ud fra det om der skal gives et point til bønderne eller vampyrerne. Begge eventhandlers giver derefter turen videre.

Spillet slutter

Spillet vindes på tre måder, enten har bønderne en score på 5, vampyrerne har en score på 6 eller Dracula er blevet valgt til Greve, efter at vampyrerne har indført tre love. Disse tre krav er sat i et if-statement, og hvis en af disse er sande vil der blive lavet et nyt intent, som indeholder playerId, serverID, samt en boolean vampireWin. Disse ting bliver sendt videre til en ny aktivitet ved navn endgame. I denne aktivitet har vi et if-statement som bestemmer hvilket TextView der bliver vist på skærmen, ud fra hvilket hold der har vundet spillet. Udover dette er der to knapper, den første vil sende spilleren tilbage til MainActivity ved at bruge et intent, og den anden vil lukke applikationen ved brug af finish(). Når der bliver trykket på disse knapper, vil serverens værdi blive sat til null, så spillet slettes fra serveren. Dette gøres ved rootRef.setValue(null).

Test af appen

Appen er blevet testet ved brug af Android Emulators samt egne telefoner. Vi har her kørt programmet, testet om der virkede samt prøvede at gøre ting som man ikke burde kunne, for at se om dette var muligt eller om det gav fejl. Derudover er der blevet brugt debug og Log funktionen, til at lokalisere og se eventuelle fejl der har været i koden. Da de individuelle funktionaliteter blev afprøvet for sig selv, virkede appen som den skulle, men da det fulde spil blev spillet igennem et par gange, blev der fundet en mindre række fejl. Med mere tid ville de kunne udbedres, men spillet er ikke desto mindre funktionelt.

Konklusion

Der er blevet implementeret alt fra "Must Have" og "Should Have" fra MoSCoW-analysen. Dette betyder at spillet er funktionelt.

Spillet kan videreudvikles let, eftersom udviklingen af app'en har fulgt en agile tilgang.

Ved videre udvikling kunne ting fra "Could Have" fra MoSCoW-analysen blive implementeret.

Det kunne have været smart at følge et design-mønster, da koden er rodet.

Litteraturlise

Litteratur

- [1] <https://www.coursera.org/learn/android-programming/home/welcome>
<https://www.coursera.org/learn/android-programming-2/home/welcome>
Coursera: Programming Mobile Applications for Android Handheld Systems: Part 1 & 2
 - [2] <https://firebase.google.com/docs/database>
Firebase realtime database vejledning
 - [3] <https://www.dafont.com/october-crow.font>
Den fancy font, October Crow, som vi har brugt til vores spil.
 - [4] <https://www.learnhowtoprogram.com/android/data-persistence/firebase-reading-data-and-event-listeners>
Vejledning til Listeners
-

Bilag

Oversigt

I begyndelsen af spillet, vil hver spiller blive tildelt en af tre roller: **Dracula**, **Vampyr** eller **Bonde**. Bønderne vil altid være i overtal, men kender kun sin egen rolle. Alle vampyrer kender hinandens og Draculas identitet. Da Dracula og hans vampyrer er i undertal, må de arbejde fra skyggerne i deres forsøg på at overtage landet, og prøve at overbevise bønderne om, at de ikke er vampyrer.

Spilregler

Roller

- Kongekandidat
- Grev / Grevinde (Count / Countess)
- Dracula
- Vampyr
- Bonde

Mål

Bønderne vinder hvis en af de følgende sker

- Fem anti-vampyrlove bliver vedtaget.
- Dracula bliver spiddet.

Vampyrene vinder hvis en af de følgende sker

- Seks vampyrlove bliver vedtaget.
- Dracula bliver valgt som Greve, efter der er blevet vedtaget tre vampyrlove.

Set-Up

Oversigt over fordelingen af bønder og vampyrer i et normalt spil, kan ses i tabellen forneden:

Players	5	6	7	8	9	10
Bønder	3	4	4	5	5	6
Vampyrer	1 + D	1 + D	2 + D	2 + D	3 + D	3 + D

Spillets gang

Hver runde i spillet består af en **valgrunde**, **lovgivningsrunde** og en **udøvende runde**. Rollen som kongekandidat går på runde med urets retning.

Valgrunde

I valgrunden nominere kongekandidaten en greve, de andre spillere skal herefter stemme for eller imod dette regeringsforslag. Kongekandidaten kan ikke nominere Kongen og Greven fra sidste regering. Hvis regeringen bliver nedstemt, går rollen som kongekandidat videre til næste spiller. Bliver regeringsforslaget nedstemt tre gange, vendes det øverste lov-kort og denne lov bliver vedtaget, derudover er alle nu valgbar til greve. Hvis flertallet stemmer ja for regeringen, går spillet videre til lovgivningsrundet.

Lovgivningsrunde

I lovgivningsrunden trækker Kongen tre kort, han kassere et af kortene og giver de resterende to kort videre til Greven, Greven vælger et af kortene, som herefter bliver vedtaget. Imens dette foregår, må Kongen og Greven ikke kommunikere.

Udøvende runde

I den udøvende runde, kan Kongen bruge en magt som bliver låst op, alt efter hvor mange vampyrlove der er blevet vedtaget. Der er fem magter Kongen kan få.

Efterforske loyalitet giver Kongen mulighed for at se en anden spillers rolle, han kan herefter dele rollen med de andre eller lyve om denne.

Speciel valgrunde giver Kongen mulighed til frit at vælge den næste kongekandidat, efter denne runde går rollen som kongekandidat videre til personen til venstre for Kongen som kaldte den specielle valgrunde.

Smugkig giver Kongen lov til at kigge på de tre øverste kort.

Spidning Kongen får lov til at spidde en spiller. Hvis denne spiller er Dracula, så ender spillet og bønderne har vundet.

Veto Når Greven modtager de to lov-kort fra Kongen, kan han kalde Veto, hvis Kongen er enig med Greven vil begge kort blive kasseret og spillet går til næste valgrunde.

Kongens magter bliver låst op ud fra hvor mange vampyrlove der er blevet vedtaget, og hvor mange spillere der er med i spillet.

Ved 5-6 spillere:

lov nr. 1	Ingen effekt
lov nr. 2	Ingen effekt
lov nr. 3	Kongen kan bruge smugkig
lov nr. 4	Kongen kan spidde en anden spiller
lov nr. 5	Kongen kan spidde en anden spiller, og veto er muligt
lov nr. 6	Vampyrene vinder

Ved 7-8 spillere:

lov nr. 1	Ingen effekt
lov nr. 2	Kongen kan efterforske loyalitet
lov nr. 3	Kongen kan starte en speciel valgrunde
lov nr. 4	Kongen kan spidde en anden spiller
lov nr. 5	Kongen kan spidde en anden spiller, og veto er muligt
lov nr. 6	Vampyrene vinder

Ved 9-10 spillere:

lov nr. 1	Kongen kan efterforske loyalitet
lov nr. 2	Kongen kan efterforske loyalitet
lov nr. 3	Kongen kan starte en speciel valgrunde
lov nr. 4	Kongen kan spidde en anden spiller
lov nr. 5	Kongen kan spidde en anden spiller, og veto er muligt
lov nr. 6	Vampyrene vinder

MoSCoW

Must have

- Database
- Multibruger integration
- Standard grafiske layouts
- Mulighed for at deltage i et bestemt spil eller lave sit eget spil

Should have

- Regelhæfte
- Brugervenlige fejlmeddelelser
- Toasts beskeder og dialog notifikationer, som informerer spilleren om events i spillet

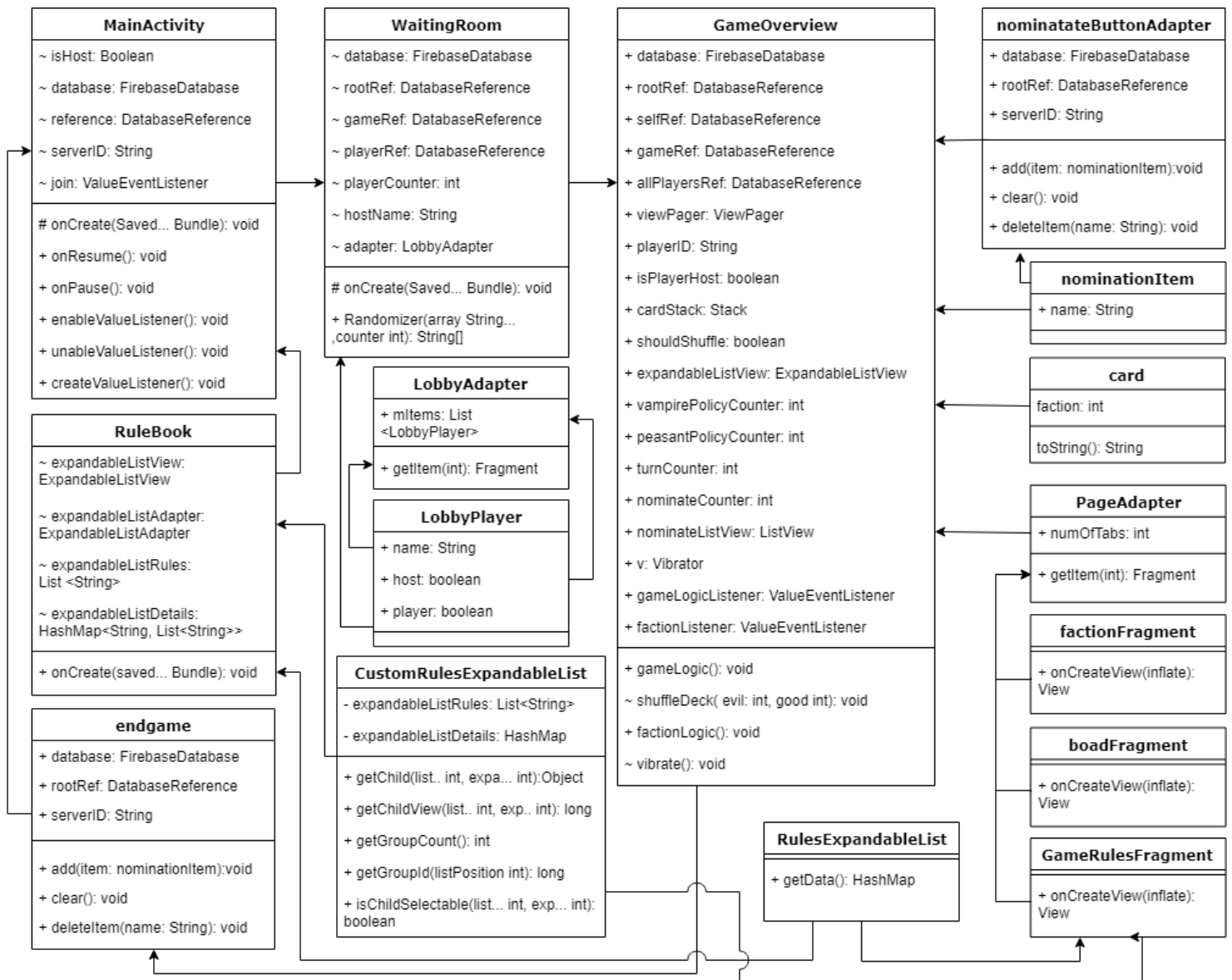
Could have

- Velkomstbillede
- Bedre brugerflade med animationer
- Lydspor som informerer spilleren om events i spillet.
- Vampyr tema og bonde tema
- Spidning af Vampyr, skal udføres vha. armbevægelse, som skal registreres af telefonens gyrometer og accelerometre
- Flere roller
- Kongens magter

Wont have

- Online multibruger matchmaking lobby
 - Statistik til at have styr på brugerens personlige statistik.
-

Klasse Diagram



Figur 1: Klasse Diagram