Innlevering 1: Tic Tac Toe

PG4600 - Mobil utvikling

Navn: Lars Erling Westbye Dahl
Dato: 18/02-2016

Beskrivelse av løsning:

Løsningen inneholder 3 activities, *MainActivity*, *GameActivity* og *ResultActivity*.
MainActivity viser som navnet tilsier, hovedskjermen som brukeren møter når han eller hun åpner applikasjonen. Her får brukeren muligheten til å fylle ut navnet på begge spillerne før brukeren starter spillet ved å trykke på "Play!"-knappen. I denne aktiviteten ligger det lite kode og logikk, men en validering av bruker-input var nødvendig. Metoden sjekker kun at navne-feltene ikke er tomme. Placeholder-navn som 'Player1' og 'Player2' er tillatt dersom ikke bruker ønsker å fylle inn navn.

Videre blir brukeren sendt til *GameActivity*, som er selve spillskjermen. Her får brukeren opp spillebrettet, en oversikt over hvem som spiller mot hverandre, hvem sin tur det er og mulighet til å plassere en markør som X eller O, avhengig av hvem som starter.
GameActivity fungerer som en GameManager og implementerer noe av spillogikken, men benytter også hjelpeklasser som Gameboard, Cell, Player og Scoreboard.
I tillegg implementerer GameActivity en nøstet indre klasse, *TableData*, for å tag'e alle rutene med hvilken rad og kolonne de befinner seg på for å enkelt kunne hente ut denne informasjonen og sende den til Gameboard for å sjekke om ruten er ledig når brukeren klikker på en rute. I Gameboard er det i tillegg implementert en metode som viser en Toast med spillerens navn dersom en spiller har vunnet, dersom det ender med uavgjort, eller om man prøver å markere en rute som allerede er tatt. Når et spill er ferdig, er det lagt inn en tidsforsinkelse på 2 sekunder før spillbrettet nullstilles og spillerne kan starte på et nytt spill. Spillet 'fryses' og det er ikke mulig å markere noen ruter i dette tidsrommet. Spillet fortsetter helt til brukeren selv ønsker å avslutte.

Klassen Gameboard holder styr på spillebrettet og hvilke celler/ruter som er ledig og ikke. Klassen tilbyr metoder metoder for å plassere en markør, sjekke om en spiller vinner horisontalt, vertikalt, diagonalt og om brettet er fullt. Metodene brukes av GameActivity for å kontrollere om en bruker kan plassere en markør i valgt rute, om en spiller har vunnet eller om utfallet av spillet blir uavgjort (spillbrettet er fullt).
Gameboard bygger et eget spillebrett i 'bakgrunn' og kommuniserer ikke direkte med GUI elementer. Dette ble gjort for å skille logikk fra GUI og deretter gjøre det enklere å bytte ut GUI eller skalere løsningen på andre måter. Alternativet var å la Gameboard styre GUI-elementer som TextView (eksempelvis Cell arver fra TextView), men av ovennevnte grunn valgte jeg ikke å gjøre dette. En stor ulempe i denne klassen er at flere av metodene er hardkodet for å fungere med et spillebrett på 3x3 ruter. Dersom man ønsker å utvide spillebrettet må denne logikken endres.

Klassen Cell representerer en rute i spillbrettet. Klassen implementerer metoder for å sette et symbol (X eller O), hente et symbol og for å se om et objekt er lik et annet. Symbolet representeres med den primitive datatypen *Char*. Et Value Objekt med andre ord.

Klassen Player representerer en av de to spillerne. Dette er også et Value Objekt som kanskje ikke er helt nødvendig grunnet de få metoden (gettere og settere), men jeg valgte allikevel å implementere den for å holde koden ryddig og oversiktlig.

Klassen Scoreboard holder styr på resultatene og er implementert som en Singleton. Denne brukes av GameActivity til å loggføre resultatet av et spill. GameActivity sender en String med spillernavn og resultat til Scoreboard. Scoreboard legger til tidspunkt i begynnelsen av strengen og lagrer dette i en liste. Listen leses av et ArrayAdapter som brukes i en ListView i ResultActivity for å vise brukeren resultatene siden spillet ble startet. Scoreboard holder resultatene til applikasjonen avsluttes (onDestroy()).

I GameActivity har brukeren mulighet til å navigere seg til Scoreboard eller tilbake til startskjermen for å starte et nytt spill ved å trykke på toppmenyen. Dersom brukeren velger Scoreboard blir han, eller hun sendt videre til *ResultActivity* som henter inn listen med spillresultater og viser dette. Her vises det klokkeslett og utfallet av spille(ne) med navn på begge spillerne. I toppmenyen får brukeren valg om å gå tilbake til spillskjermen. Dette kunne nok muligens vært utelatt da brukeren har en tilbake-knapp på enheten.

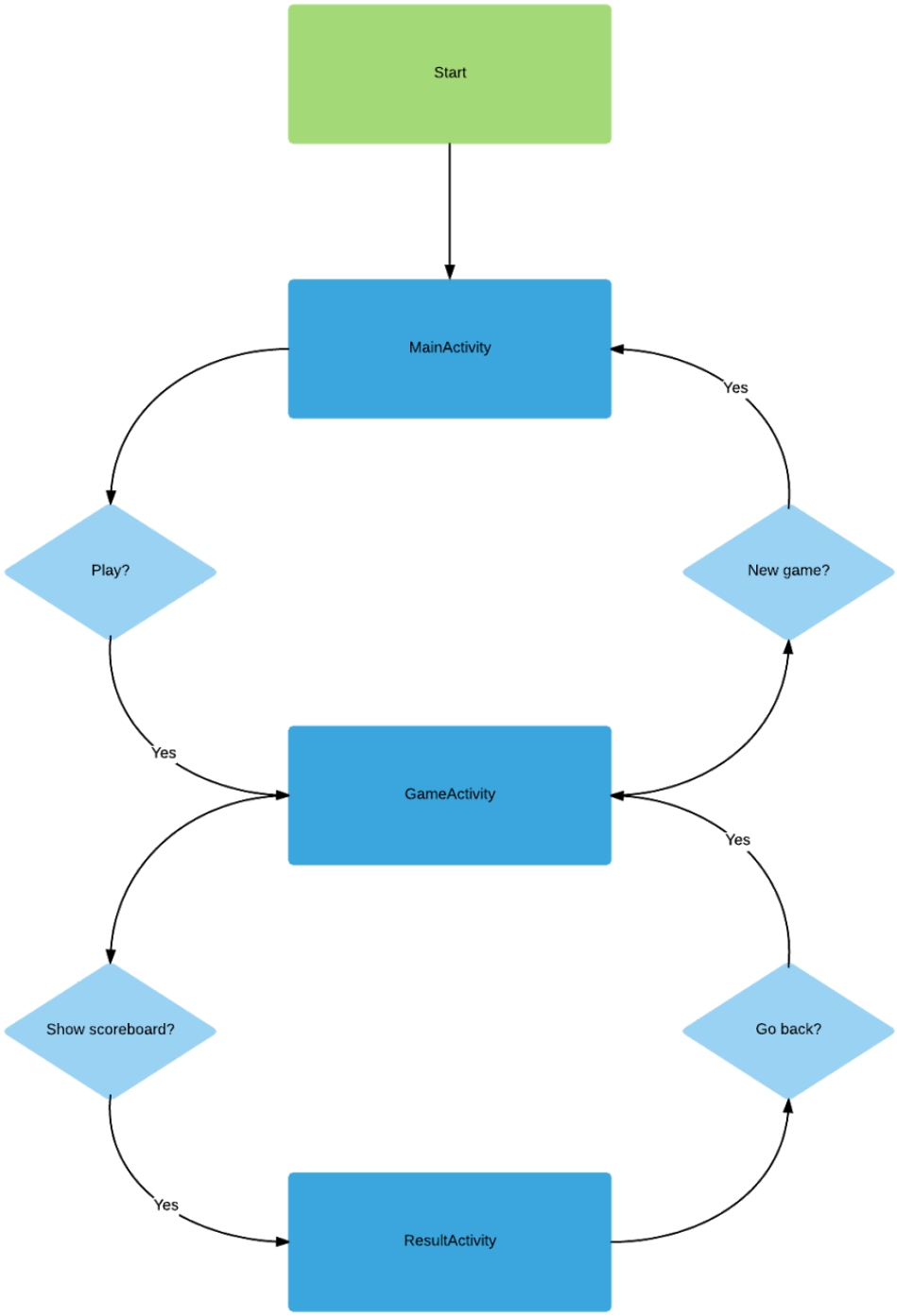
Layout
Layouten i spillebrettet bygges med LinearLayout (vertikal), TableLayout og TableRows. Tablelayoutene blir pakket inn i en LinearLayout. Tablelayout benyttes til spillertekst, hvem sin tur det er og selve rutene.
Layout og utseende er relativt enkelt, og her er det store rom for forbedringer. Jeg valgte å fokusere mest på funksjonalitet i denne oppgaven.

Videre har jeg sørget for følgende:
- Alle strenger er definert i values/strings.xml
- Alle farger er definert i values/colors.xml
- Alle stiler er definert i values/styles.xml (Hvordan spillersymbolene representeres)

Teknisk informasjon
Android API:
Løsningen er utviklet med Android API level 21, og er derfor rettet mot Android 5.0 Lollipop og nyere. Strengt talt kunne løsningen blitt basert på et lavere API-nivå da det ikke tas i bruk nye funksjoner som inkluderes i API 21 og oppover. Valget kan uansett forklares med undersøkelser gjort av Google som viser at 34,1% av alle Android-enheter som var med i undersøkelsen benytter Lollipop (Android Developers, 2016), og at det trolig også øker i antall brukere fremover.

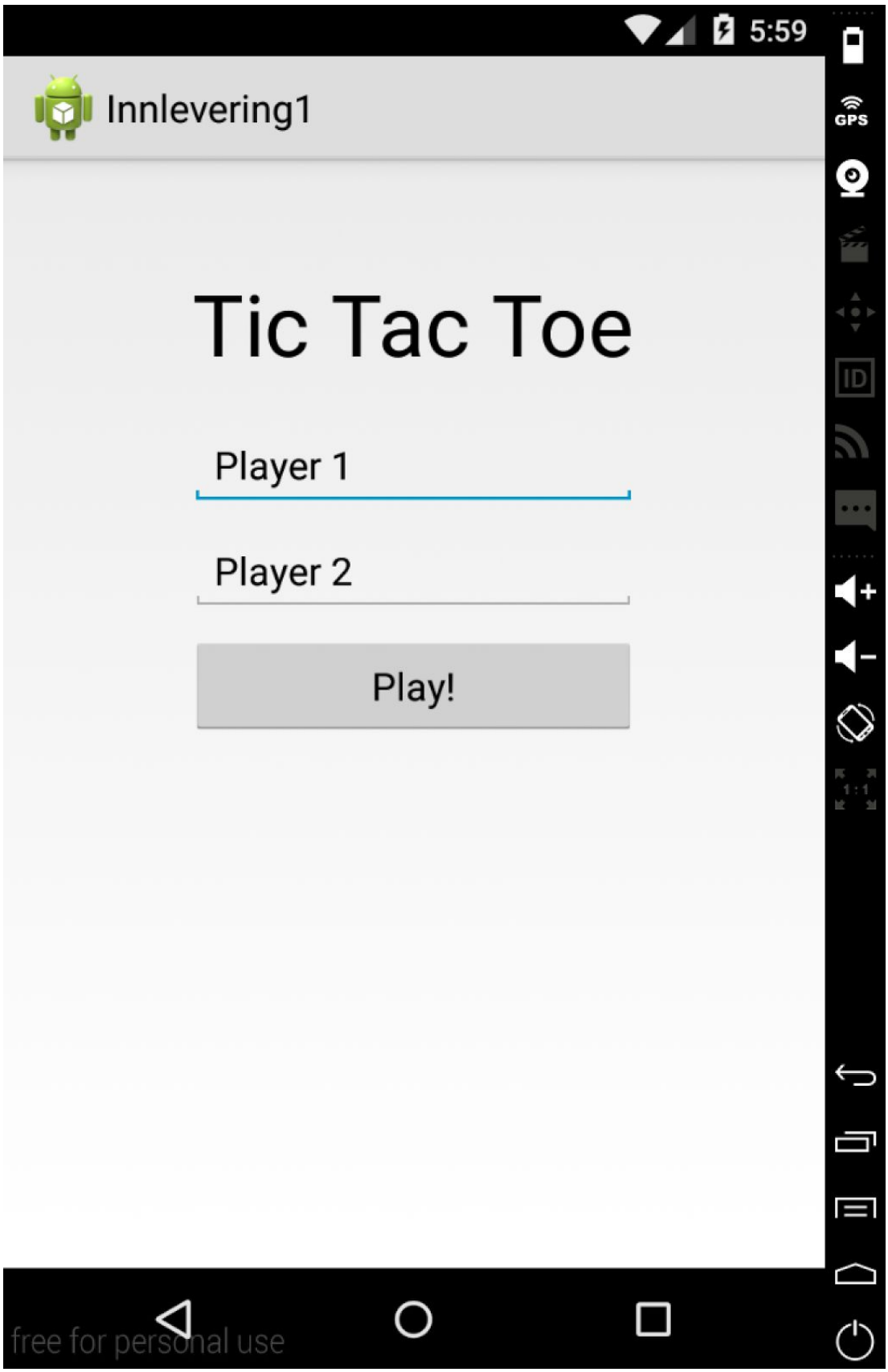
Avhengigheter:
- JUnit 4.12
Brukt i forbindelse med enhetstesting av hjelpeklasser.

Flytdiagram

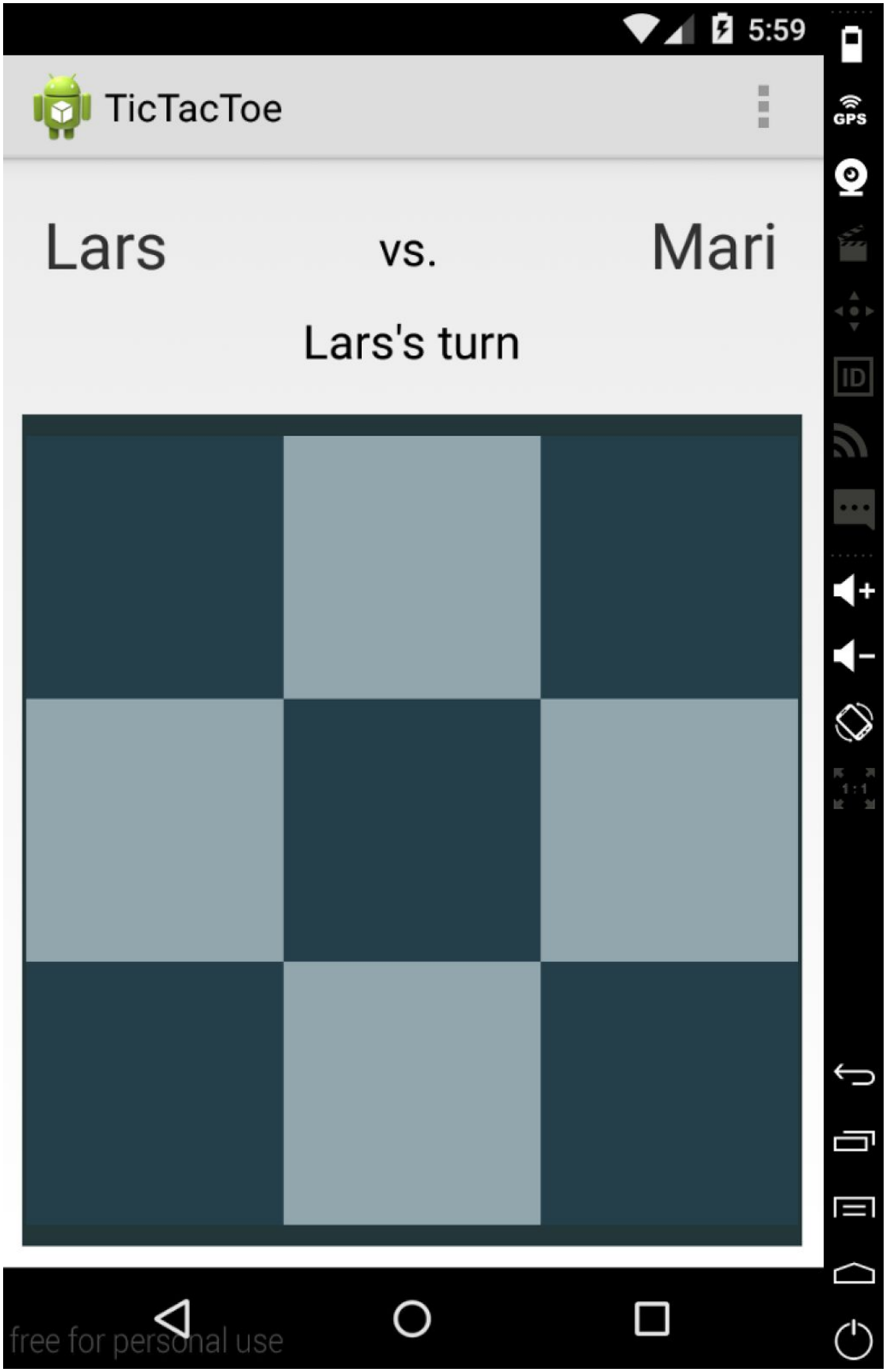


Skjermbilder

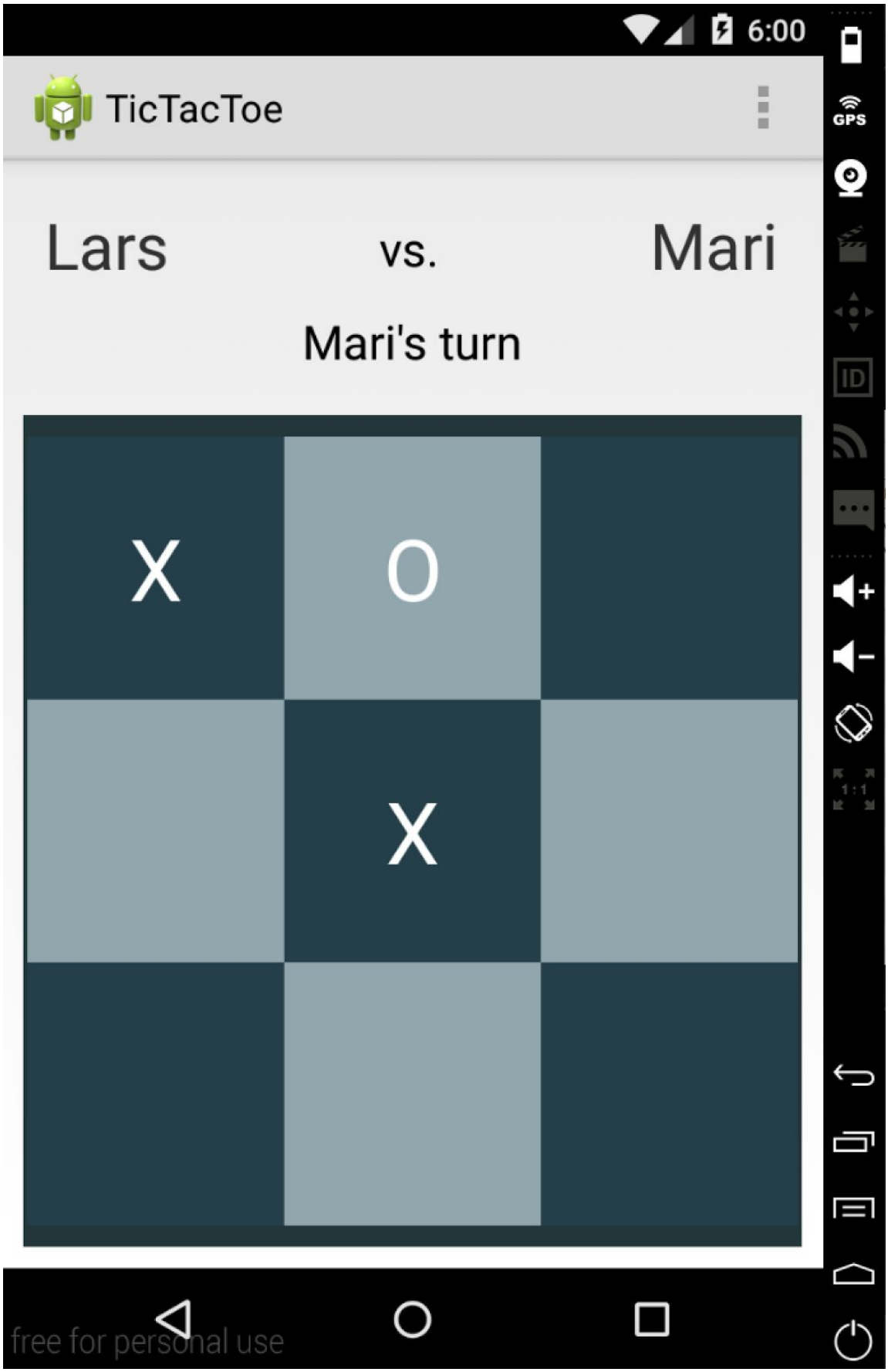
Startskjerm (MainActivity)



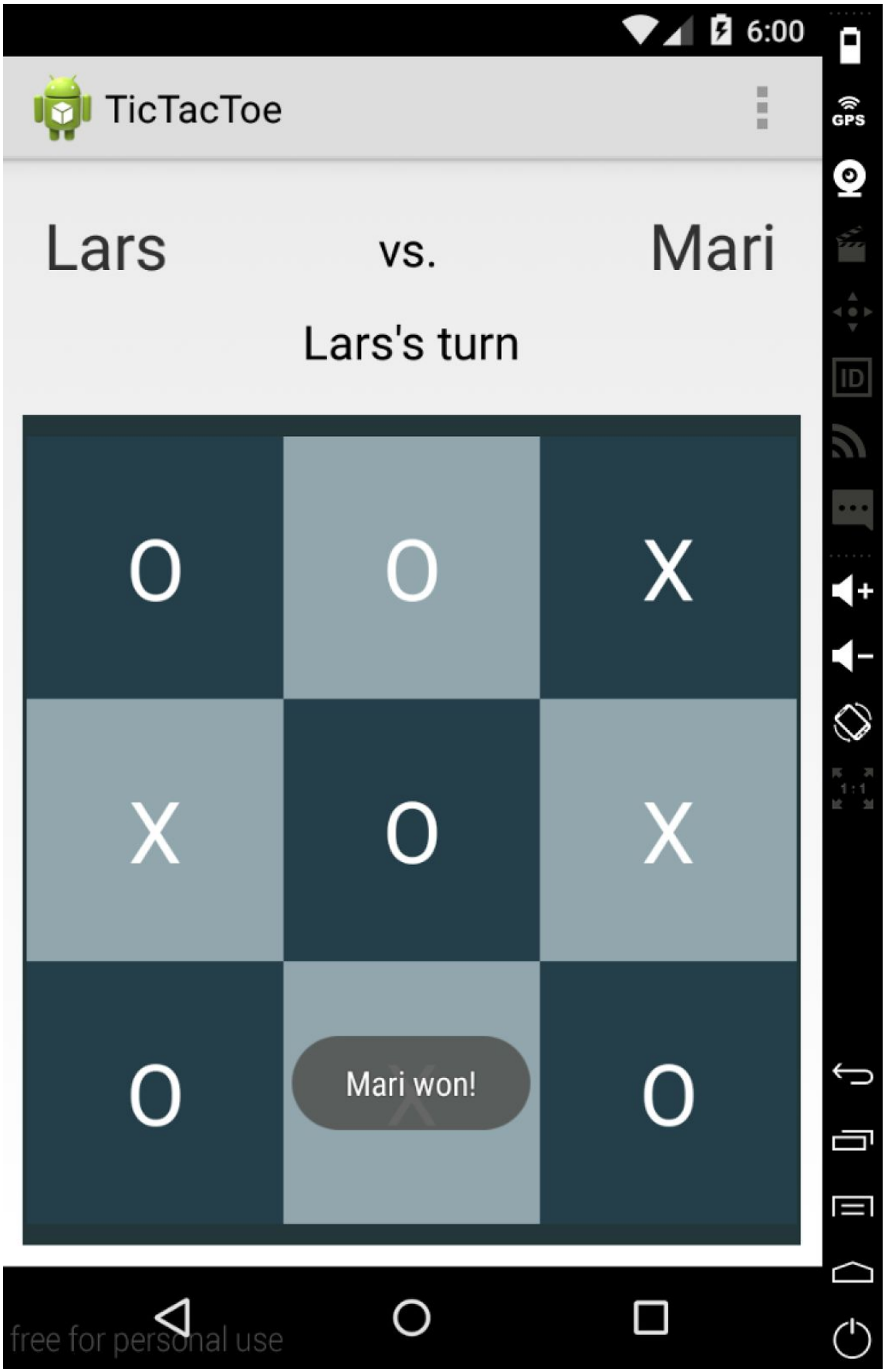
Spillskjerm (GameActivity)



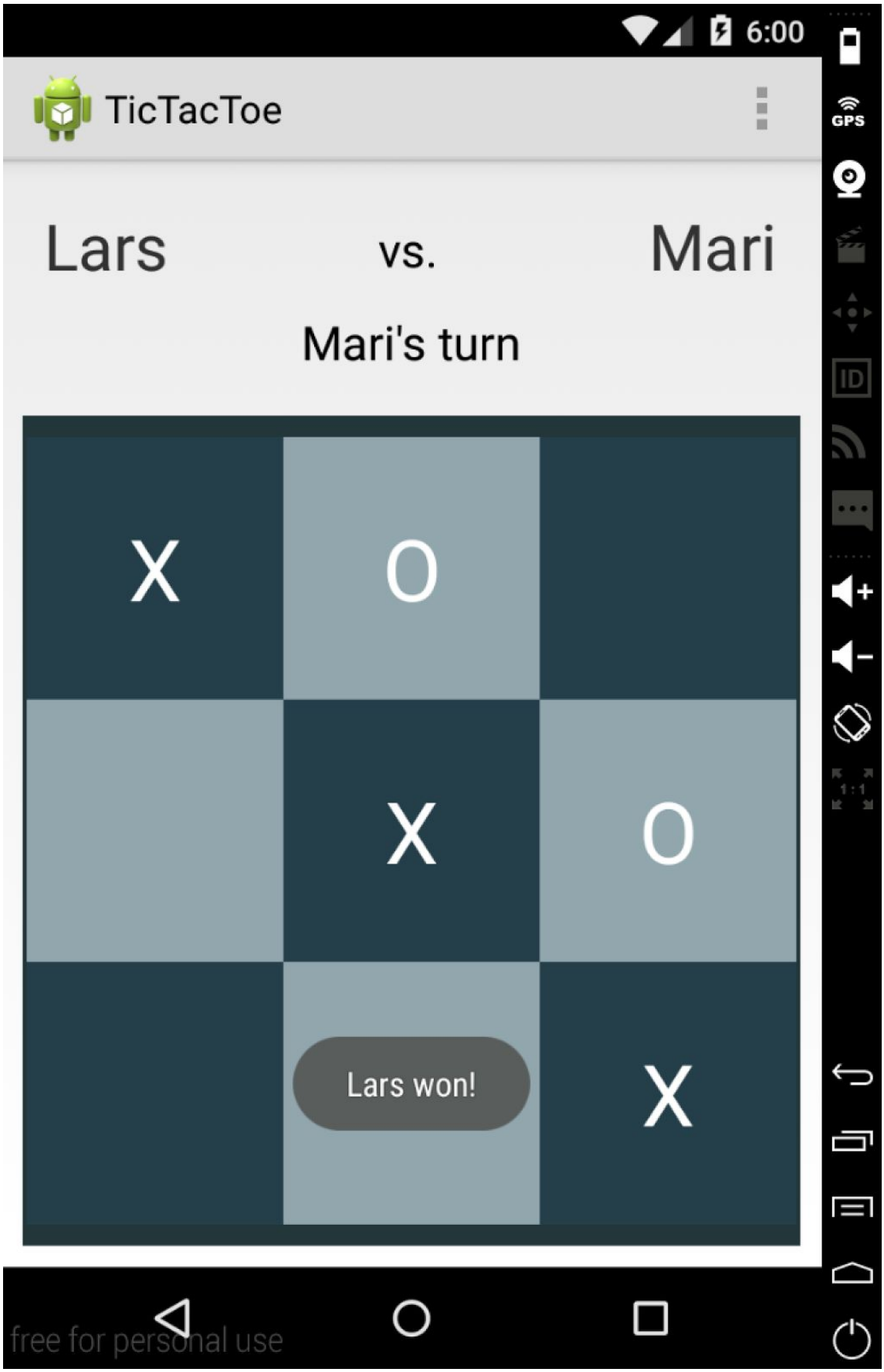
Pågående spill



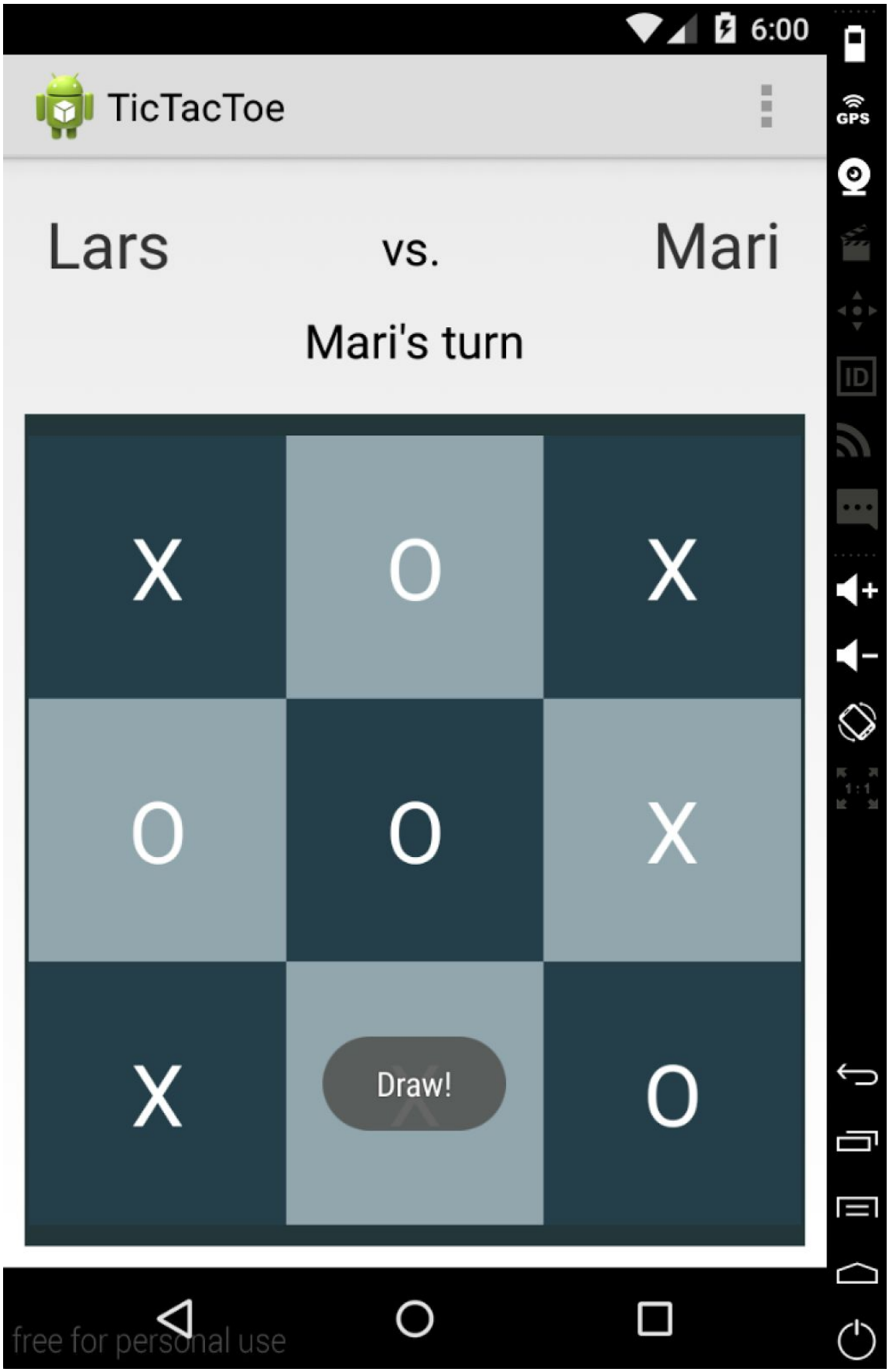
Spiller O vinner:



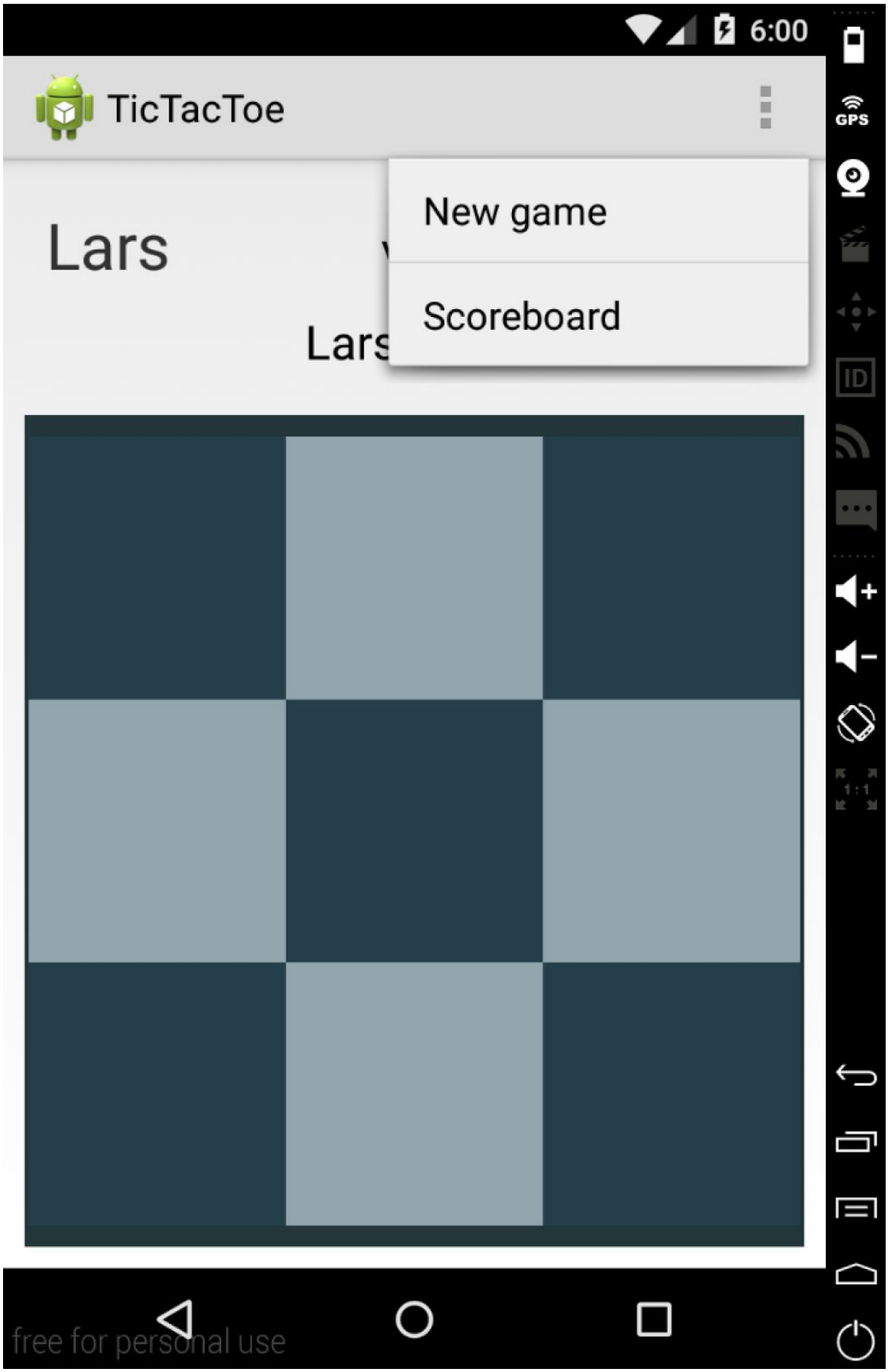
Spiller X vinner:



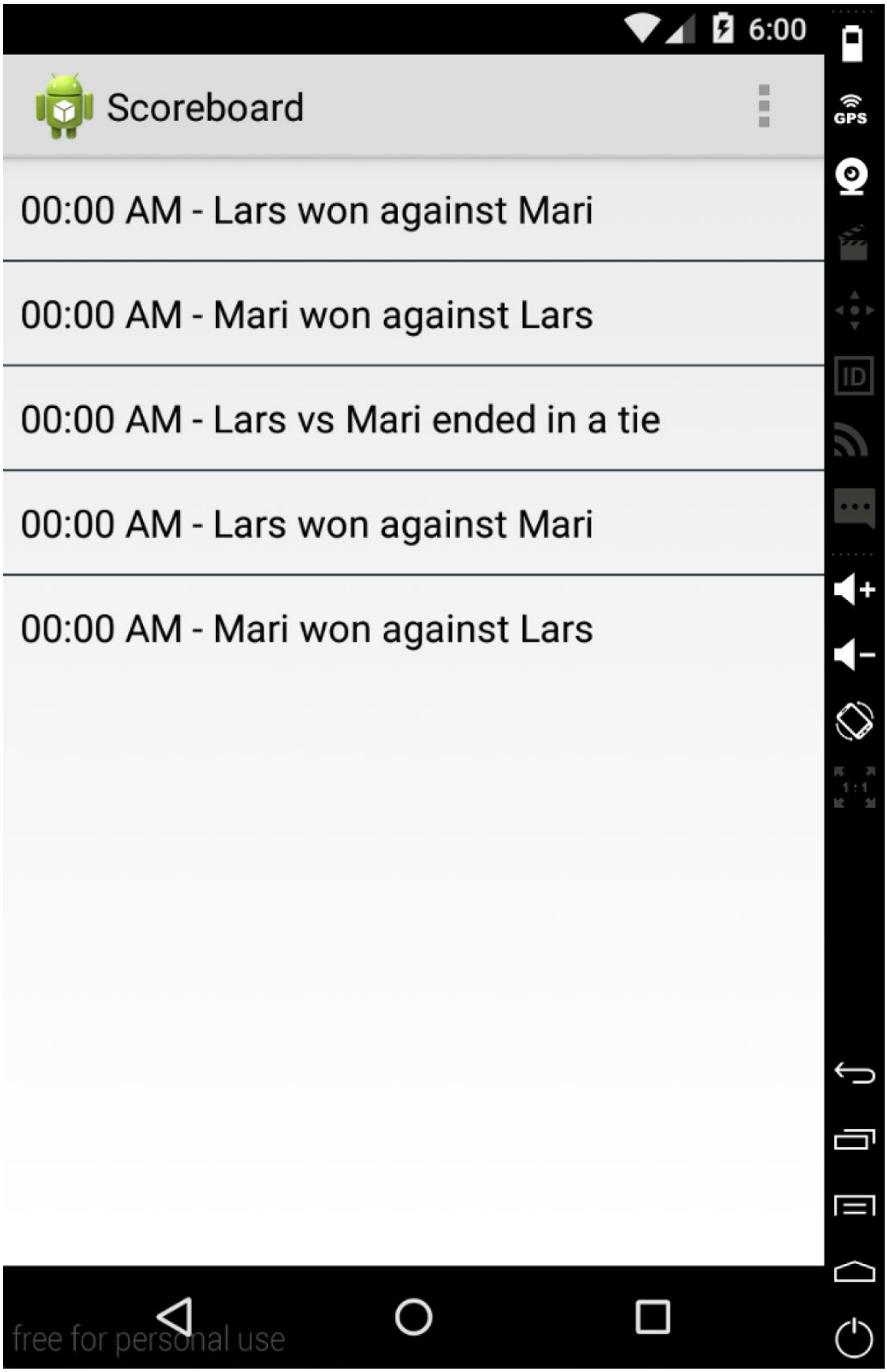
Spillet ender i uavgjort:



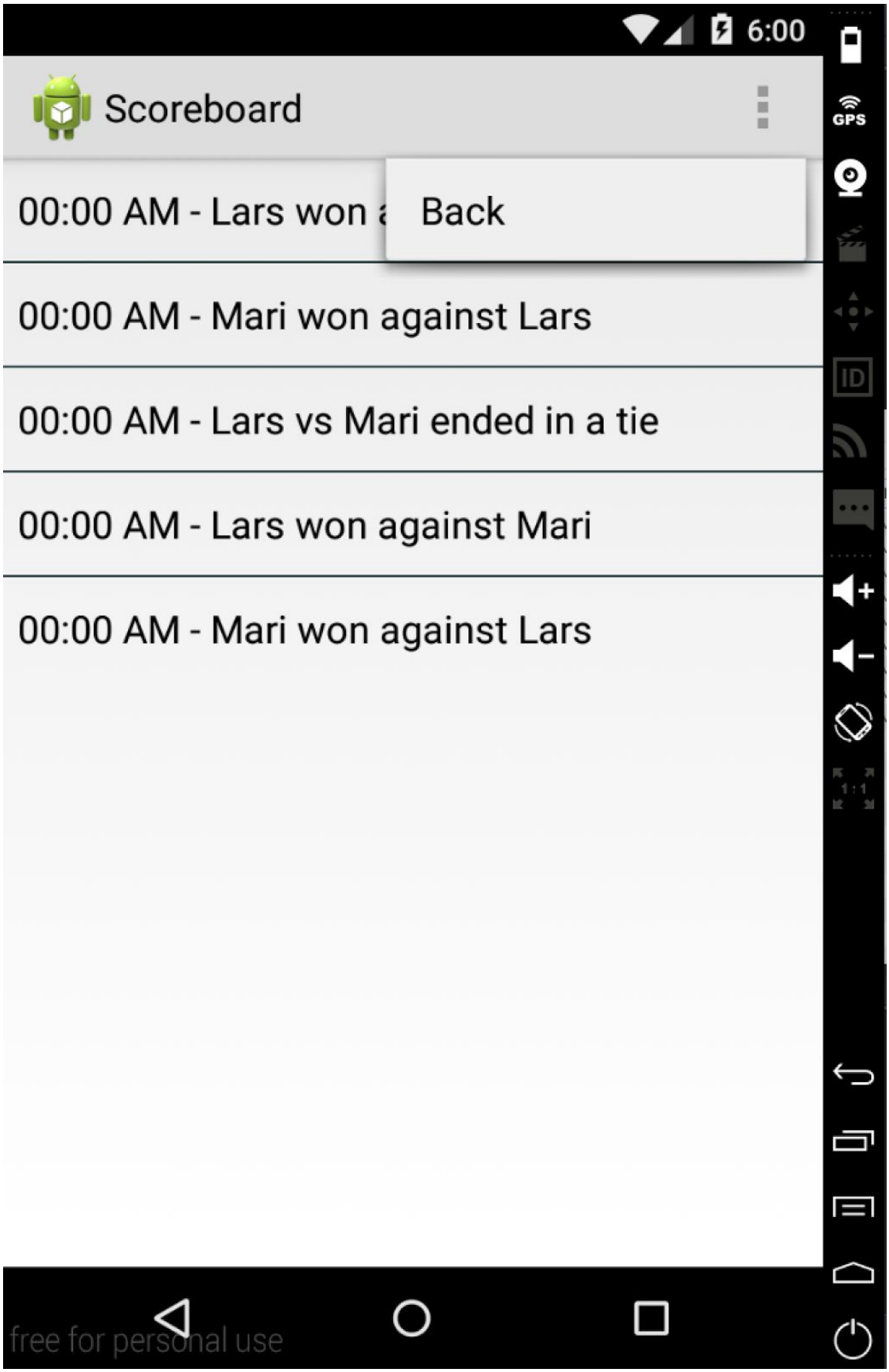
Toppmeny på spillskjerm



Resultatliste (ResultActivity)



Resultatliste, toppmeny



Kilder:

Stack Overflow (2013) *"Delay actions in android*" [Internett]. Tilgjengelig fra: http://stackoverflow.com/questions/14186846/delay-actions-in-android [Lest 14. Februar 2016].

Stack Overflow (2014) *"How to find the row and column index in Table Layout?"* [Internett]. Tilgjengelig fra: http://stackoverflow.com/questions/27644660/how-to-find-the-row-and-column-index-in-table-layout [Lest 14. Februar 2016].

Stack Overflow (2012) *"How to get the current time in android"* [Internett]. Tilgjengelig fra: http://stackoverflow.com/questions/11913358/how-to-get-the-current-time-in-android [Lest 14. Februar 2016].

Android Developers (2016) *Platform Versions* [Internett]. Tilgjengelig fra: http://developer.android.com/about/dashboards/index.html#Platform [Lest 18. Februar 2016].