Unknown Othello App

Bram Arends en Luuk Linders

Samenvatting

In dit document wordt onze Othello App besproken. Hoe de app in elkaar zit en hoe je het moet gebruiken.

Inleiding

Onze applicatie is een denkspel bedacht in 1883. Je kan er het originele tweespeler mee spelen, maar ook een aantal varianten. Voor de regels van het denkspel zie de applicatie zelf, onder het knopje "Rules". Je kan offline multiplayer spelen tegen andere mensen of tegen onze zelfgemaakt computerspeler spelen. Het is bedoeld voor mensen die gewoon een leuk spelletje willen doen op hun smartphone.

Voorwoord

Dit document is bedoeld als documentatie van onze app "Othello". In het eerste gedeelte van dit document wordt beschreven hoe onze app er qua structuur uit ziet. In het tweede gedeelte staat onze evaluatie van ons onderzoek en in het derde gedeelte staat onze reflectie van dit project.

1.Beschrijving

Specificaties

Met deze app kan je offline Reversi spelen tegen andere mensen of tegen een computerspeler. Dit kan met de traditionele 2 spelers, maar ook met 3 of 4 spelers. Ook is er een variant waarbij je juist zo een min mogelijk aantal stukken van jezelf wilt hebben, genaamd "Reversed Reversi". De spelregels staan in de app zelf. Als de gebruiker aan het spelen is kan de gebruiker een zet ongedaan maken met de "undo" knop. En met de "help" knop kan de gebruiker de mogelijke zetten zien die hij kan doen.

Structuur

Het systeem is ruwweg opgebouwd uit vier onderdelen: De logica, de graphical user interface, de layout en de computerspeler.

GUI:
De GUI zorgt voor alle menu's, pop-up schermpjes,het speelbord en een beetje logica, omdat alle Views (Buttons, ImageViews, etc.) niet buiten de Activity aangepast kunnen worden.

De klassen:

- **Mainscreen:** Dit is het hoofdmenu, dit krijg je te zien als je het spel opstart en het bevat een Button naar HelpScreen en naar SelectPlayers.
- **HelpScreen:** Hierin staan de spelregels van het spel, ook van de andere varianten.
- **SelectPlayers:** Hierin kan de gebruiker het aantal spelers regelen of voor het speltype Reversed Reversi kiezen, die standaard voor twee spelers is. Als de gebruiker klaar is met kiezen gaat hij naar SelectOpponents.
- **SelectOpponents:** De gebruiker kan hier het aantal computerspelers en het aantal menselijke spelers kiezen. Als de gebruiker op Start klikt gaat hij naar de BoardGUI.
- **BoardGUI:** Hierin wordt het grafische speelbord weergegeven en bijgehouden, ook een beetje logica is hierin verwerkt. Het zorgt ook voor de pop up vensters tijdens het spel. De gebruiker kan hier zijn zetten doen.

Logica:
Hierin zitten alle spelregels van het spel en worden de zetten van de gebruiker(s) gecontroleerd op geldigheid. Ook het bord bevindt zich in de Logica.

De klassen:
- **Board** houdt de huidige situatie van het spel bij. Alle velden van het speelbord vormen samen één matrix, waarin de kleur die elk vakje op dit moment heeft wordt opgeslagen (zie Colour);
- **Colour** is een enumeratie van kleuren. 'Empty' staat voor een leeg vakje, 'help' voor een vakje waar de volgende beurt een zet kan worden gedaan;
- **Coord** representeert een coördinaat op het bord en heeft een 'column' en een 'row' waarde;
- **Direction** is een enumeratie van de acht windrichtingen;
- **Player** representeert een speler die deelneemt aan het spel. Hij heeft een kleur, een spelertype (zie PlayerType) en houdt bij hoeveel stenen de speler op het bord heeft liggen. Ook heeft hij een computerspeler aan zich gekoppeld, die gebruikt wordt als de speler van het type 'cpu' is. 'score' is eigenlijk vooral bedoeld voor de computerspeler, maar zou in een uitbreiding ook als extraatje voor alle spelers getoond kunnen worden;
- **PlayerType** is een enumeratie voor de klasse 'Player', die onderscheidt maakt tussen een menselijke speler ('human') en een computergestuurde speler ('cpu').

Computerspeler:
De computerspeler bestaat uit twee klassen:
- **CPU**: dit is de klasse die het grote werk doet. Hij heeft een aantal variabelen, die bijgesteld kunnen worden om de computerspeler te verbeteren. De variabele 'level' geeft aan hoeveel zetten de computerspeler vooruit kan denken, en bepaalt daarmee het niveau van de computerspeler. De functie 'nextMove()' zet een aantal initiële waarden, en laat dan de functie 'bestMove'() het echte werk doen. 'bestMove()' is een recursieve methode, die voor zijn meegegeven speler ('player') de optimale zet bepaalt. Elke mogelijke zet ('validCoords') wordt gedaan op steeds een nieuwe kopie van het meegegeven bord, om zo te kijken welke zet de hoogste score geeft (zie ValueBoard). Als de computer nog meerdere zetten vooruit mag kijken (als 'depth' nog kleiner is dan 'level'), roept de methode zichzelf na iedere zet weer aan om de beste zet voor de volgende speler te bepalen (de computer gaat er dus vanuit dat iedere speler een zo goed mogelijke zet doet). De meegegeven kopie van de spelers ('ghosts') wordt geupdate en uitgelezen om tot een score voor de betreffende zet te komen. Ook het aantal omliggende stenen (keer de modifier 'frontierValue') en het aantal zetten dat de volgende speler kan doen (keer de modifier 'moveValue') spelen een rol in de totale score van de zet.
- **ValueBoard** is een object dat weergeeft welke waarde elk vakje op het bord heeft.Het is bedoeld om het strategisch vermogen van de computerspeler te vergroten. Dit berust op het feit dat de hoeken strategisch gezien veel meer waard zijn, omdat die nooit meer te veranderen is naar een andere kleur. Een vakje aan de rand, twee vakjes van een hoek vandaan, is strategisch belangrijk omdat je daarmee meer kans maakt op de betreffende hoek.

De layout (XML):
De XML files zorgen allemaal voor de layout van de menu's en het speelbord.

Ontwerpverantwoording

We hebben geprobeerd de logica, de GUI en de computerspeler zoveel mogelijk gescheiden te houden, alleen door wat beperkingen (je kan bijvoorbeel niet van buitenaf een Activity veranderen) is er toch wat logica de GUI van het bord binnen gekomen. Op dit na, is de scheiding toch gelukt.

De app neemt weinig geheugen in, omdat we overbodige animaties en 3D graphics hebben weggelaten. De computerspeler is gebaseerd op een aantal bekende strategische 'trucjes', waar we op internet onderzoek naar hebben gedaan.

2. Evaluatie

Onderzoeksvraag

Wat is de beste computerspeler? De computerspeler moet wel wat uitdaging geven, dus niet te makkelijk of te moeilijk, dus de computerspeler heeft verschillende denkniveau's nodig. De performance moet ook goed zijn, op een smartphone heb je immers minder geheugen en de processor is minder krachtig.

Achtergrond

We hebben veel onderzoek gedaan op internet, twee bronnen die we vooral hebben gebruikt zijn:

- http://www.site-constructor.com/othello/Present/Basic_Strategy.html
- http://files.boardgamegeek.com/file/download/1maqhnh224/Rolit_Rules.pdf?

Op de eerste site staat een concrete invulling van de waardenmatrix (valueBoard), in de tweede een opzet, die ruimte laat voor vrije invulling. We hebben vier invullingen van de waardenmatrix gemaakt, één met een matrix die overeenkomt met die op de eerste site, één waarvan de invulling dicht bij de tweede site komt. De overige twee zitten daar tussenin.
Over hoe zwaar de overige aspecten van de computerspeler moeten wegen, hebben we niks kunnen vinden.

Methode

In onze zelfgemaakte computergestuurde speler is er ruimte voor een aantal variabelen (zie 1. Beschrijving → structuur → computerspeler). We maken een aantal invullingen van de variabelen en organiseren een toernooitje tussen deze verschillende computerspelers. De uitslagen slaan we op in een tabel en we kijken zo "wie" de beste is. We testen alles op level 2, het maximale dat een smartphone aan kan. Omdat toeval, het aantal spelers en in welke beurt de speler aan de beurt is ook een rol kunnen spelen, zullen we elke ronde een aantal keren uitvoeren, in verschillende volgordes en met verschillende aantallen spelers. Iedere setting wordt 1000 keer uitgevoerd;

Eerst maken we een aantal invullingen van het 'valueBoard' (en zetten de overige variabelen op 0).
Hierna testen we een aantal invullingen van 'frontierValue' en tot slot een aantal van 'moveValue'.

Resultaten

In de bijgesloten resultaten is te zien hoe de computerspelers tegen elkaar presteerden. De volgorde van namen komt overeen met de volgorde van wanneer welke speler aan de beurt was.

De eerste ronde ging over het waardenbord. Hier blijkt dat computerspeler 'Kees' (op één na) elke ronde heeft gewonnen. Het lijkt ons dan ook veilig om te zeggen dat deze de winnaar van deze ronde is.

De tweede ronde ging over de frontierValue.

De derde ronde ging over de moveValue.

Conclusie

De beste computergestuurde speler is die met de waardenmatrix gegenereert uit { 50, -1, 5, 2, -10, 1, 1, 1, 1, 0 } met frontierValue 0.5 en moveValue 0.2.

3. Reflectie

Product

In eerste instantie waren we van plan om een schaak applicatie te maken, maar we kwamen er al gauw achter dat dat veel te moeilijk en tijdrovend was geweest. Uiteindelijk kwamen we uit op een ander bekend denkspel, Reversi. Dit spel is een stuk minder complex, maar wel strategisch. We zijn best tevreden over ons product, vooral de computerspeler, die is nog best lastig te verslaan. Ook zijn we tevreden over de graphics van de app, die zien er best aardig uit voor een paar beginners.
Minder tevreden zijn we over de code in de BoardGUI, die kan die nogal rommelig overkomen, maar dat kwam ook doordat elke goede idee die we hadden niet werkte voor een activity, ook de touchlistener in die activity werkte niet echt mee. Dus na een hoop gepruts zijn we op de klasse gekomen die we nu hebben.
Maar uiteindelijk zijn we toch op een leuke app gekomen, die erg leuk is om te spelen, vooral met meerdere spelers bij elkaar, dat geeft toch een leuker effect dan online multiplayer.

Proces

Omdat er in het begin moeilijkheden waren met het groepje waren we relatief laat aan het project begonnen, daarna was er ook nog een groepslid weggevallen, maar toen we eenmaal begonnen waren verliep het samenwerken prima. We hebben daar geen problemen mee gehad. Alleen we zouden de volgende keer wel eerder aan een opdracht moeten beginnen om de deadlines te halen, vooral als er verslagen ingeleverd moesten worden. Op dat soort inlevermomenten moest het meeste nog op het laatste moment gedaan worden.
De applicatie zelf maken ging verrassend beter dan de documenten, terwijl dat juist het moeilijkste gedeelte is. Dat kwam waarschijnlijk omdat het programmeren leuker is dan de documenten schrijven.

Learner reports

Bram Arends:

Ik vond het wel leuk dat we een applicatie moesten maken voor een smartphone in plaats van voor de computers, dat is weer iets wat anders. Dat we zelf mochten bepalen wat voor een app het zou zijn maakt het alleen maar leuker, omdat je meer vrijheid hebt. Het eerste wat dan in je opkomt is natuurlijk een game, en dat maakten we dan ook.
Ik had hiervoor nog geen enkele ervaring op het gebied van meedoen in "grote" projecten, ik kwam er al gauw achter dat je vlug moet gaan bedenken wat voor een product je wilt maken, want de Pilot fase eindigde vlugger dan ik had gedacht. Ook het leren van een nieuwe "programmeertaal" duurt langer dan gedacht, ik had nog gelukt dat Android programmeren veel op Java lijkt, anders was ik waarschijnlijk nog langer bezig geweest met de user interfaces.
Ook ben ik erachter gekomen dat we toch weer niet goed plannen, omdat we weer vlak voor de deadline allemaal dingen af moeten maken, wat weer een beetje stress oplevert.

Luuk Linders

Het was erg leuk om voor een smartphone een applicatie te maken. Ik had dat nog nooit gedaan, en het is mijn eerste echt grote project. Het daadwerkelijke smartphone-gerelateerde programmeerwerk is echter voor het grootste deel gedaan door Bram, ik heb meer de onderliggende programmatuur op mij genomen. Met tot gevolg dat ikzelf niet alles snapte van waar Bram mee bezig was, en dat was soms lastig. Vooral omdat ik niet doorhad dat als ik kleine veranderingen in mijn programmacode maakte, Bram toch veel tijd nodig had om dit over te zetten naar de juiste code voor op de smartphone. Dat is een puntje van de samenwerking die de volgende keer beter kan. Verder was de communicatie in het begin niet zo goed, waardoor het werk zich op ging stapelen. Toen hadden we ook nog de pech dat onze derde projectpartner ermee stopte. Tegen het einde ging de communicatie gelukkig stukken beter.
Ik ben ook best wel tevreden over het eindproduct, al had hij hier en daar nog verbeterd kunnen worden. Voor een eerste applicatie is het toch een mooi product.