
BATTLE ROYALE CHESS

Een historische game in een modern jasje

Ontwikkeld door Peter Iedema

ten behoeve van de bootcamp "Full Stack Developer"

aan NOVI Hogeschool

04-02-2022

Inleiding

Het doel van deze eindopdracht was om te leren hoe ik verschillende technieken in onder andere Javascript, Java en React kon toepassen. Het leek mij leuk en uitdagend om dit door middel van een game te doen. Omdat ik wel eens schaak speel en hou van Battle Royale games wilde ik dit graag combineren.

Volgens mij is dit goed gelukt. Het spel is speelbaar en ziet er leuk uit. Na verloop van tijd zullen er meer functies in komen en zal het echt een volwaardige game worden. Door middel van de verschillende technieken in de front- en backend heb ik het gevoel dat ik goed heb laten zien wat ik kan.

Toch zijn er verschillende dingen die ik een volgende keer anders zou doen, waar ik van geleerd heb en wat mij weer een betere developer maakt. Dat hoort natuurlijk ook wel bij het developen. Iedereen leert continu bij.

Bedankt voor de leuke opleiding en wellicht tot een volgende keer!

Inhoudsopgave

Inleiding	2
Inhoudsopgave	3
Probleemstelling	4
Oplossing	4
Functionele eisen	5
Niet-functionele eisen	6
Inspiratiebronnen	7
Use cases	9
Wireframes	10
Verantwoordingsdocument	12

Probleemstelling

Omdat ik graag een Battle Royale game speel (Apex Legends) en daarnaast ook graag schaak leek deze opdracht me een uitgelezen mogelijkheid om de 2 genres te combineren. Er zijn op internet wel wat voorbeelden te vinden van schaakspellen in Battle Royale variant maar deze zijn grafisch en qua gameplay heel anders dan ik voor ogen heb. Het zou fijn zijn als er een applicatie zou zijn die via de browser en mobiele browser benaderbaar is waar makkelijk en snel een spel gespeeld kan worden.

Oplossing

Het moet mogelijk zijn om in te loggen. Hierna moet de game de mogelijkheid bieden om multiplayer online te schaken in Battle Royale stijl. Ook moet er een highscores, profiel en speluitleg pagina te bekijken zijn.

Functionele eisen

1. De speler moet in kunnen loggen.
2. De speler moet een avatar kunnen kiezen.
3. De speler moet bij registratie een gebruikersnaam kunnen kiezen.
4. De speler moet bij registratie een wachtwoord kunnen kiezen.
5. De speler moet zijn gebruikersnaam kunnen wijzigen.
6. De speler moet plaats kunnen nemen in de lobby voor een nieuw spel.
7. De speler moet een zet kunnen doen in het spel.
8. De speler moet kunnen zien welke zetten hij mag doen.
9. De speler moet kunnen zien hoeveel tijd hij nog heeft voor een zet.
10. De speler moet kunnen zien in welke ronde hij zit.
11. De speler moet kunnen zien in welke rondes het speelveld kleiner wordt.
12. De speler moet zijn huidige score kunnen zien.
13. De speler moet de score van andere spelers kunnen zien.
14. De speler moet kunnen zien welke stukken hij nog heeft.
15. De speler moet kunnen zien welke stukken andere spelers nog hebben.
16. De speler moet de highscores kunnen bekijken.
17. De speler moet in de highscores op naam kunnen zoeken.
18. De speler moet zijn eigen positie op de highscore lijst kunnen zien.
19. De speler moet zijn vorige games terug kunnen zien.
20. De toeschouwer hoeft niet in te kunnen loggen.
21. De toeschouwer moet een lijst met actieve games kunnen zien.
22. De toeschouwer moet een game kunnen kiezen om te bekijken.
23. De toeschouwer moet kunnen zien hoelang de ronde nog duurt.
24. De toeschouwer moet kunnen zien in welke ronde het spel is.
25. De toeschouwer moet kunnen zien in welke rondes het speelveld kleiner wordt.
26. De toeschouwer moet de score van de spelers kunnen zien.
27. De toeschouwer moet kunnen zien welke stukken de spelers nog hebben.
28. De toeschouwer moet de highscores kunnen bekijken.
29. De toeschouwer moet in de highscores op naam kunnen zoeken.
30. De speler moet het bord kunnen inzoomen.
31. De speler moet het bord kunnen verplaatsen.
32. De toeschouwer moet het bord kunnen inzoomen.
33. De toeschouwer moet het bord kunnen verplaatsen.
34. De admin moet users kunnen aanmaken.
35. De admin moet user gegevens kunnen wijzigen.
36. De admin moet nieuwe spelvarianten kunnen aanmaken.
37. De admin moet nieuwe spelvarianten kunnen wijzigen.
38. De admin moet in het spelvariant het spelbord kunnen maken.
39. De admin moet zijn eigen instellingen kunnen wijzigen.

40. De speler moet zijn installingen kunnen wijzigen.
41. De speler moet een chess.com account naam op kunnen geven.
42. Elke gebruiker moet van een speler die een chess.com accountnaam heeft opgegeven zijn statistieken in kunnen zien.
43. De spelerslijst moet op rank gesorteerd zijn.
44. De toeschouwer moet een lijst van spelers kunnen zien.

Niet-functionele eisen

1. Elke ronde moet synchroon lopen bij elke speler.
2. De verbinding met de server moet stabiel zijn.
3. Mocht een speler toch de verbinding verliezen dan moet deze weer terug in het spel kunnen komen en zetten kunnen doen.
4. Als de speler af is moet deze geen zetten meer kunnen doen, maar nog wel mee kunnen kijken met het spelverloop.
5. Een game moet gestart worden zodra er voldoende spelers in de wachtrij staan.
6. Elke applicatie moet regelmatig om updates (gamedata, games, queue) vragen aan de server.

Inspiratiebronnen



Deze afbeelding komt van chess.com. Daar kun je een 4 player variant spelen. Deze image laat ongeveer zien hoe ik de game wil weergeven op de desktop en hoe ik het speelbord wil opbouwen. In mijn applicatie moeten de verschillende spelers ook weergegeven worden door middel van verschillende kleuren.



Het kleurgebruik van het bord in bovenstaande afbeelding vind ik erg mooi.



Deze image geeft weer hoe ik in de mobile weergave de elementen om het spel heen wil positioneren. De informatie die op die plekken getoond wordt zal iets anders zijn maar positioneel moet het er zo uit komen te zien.

Use cases

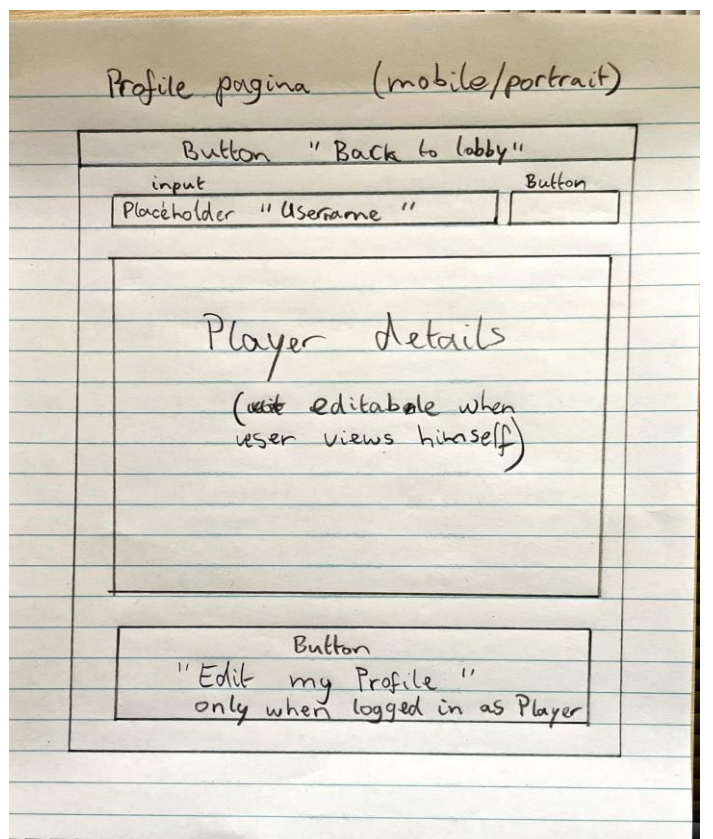
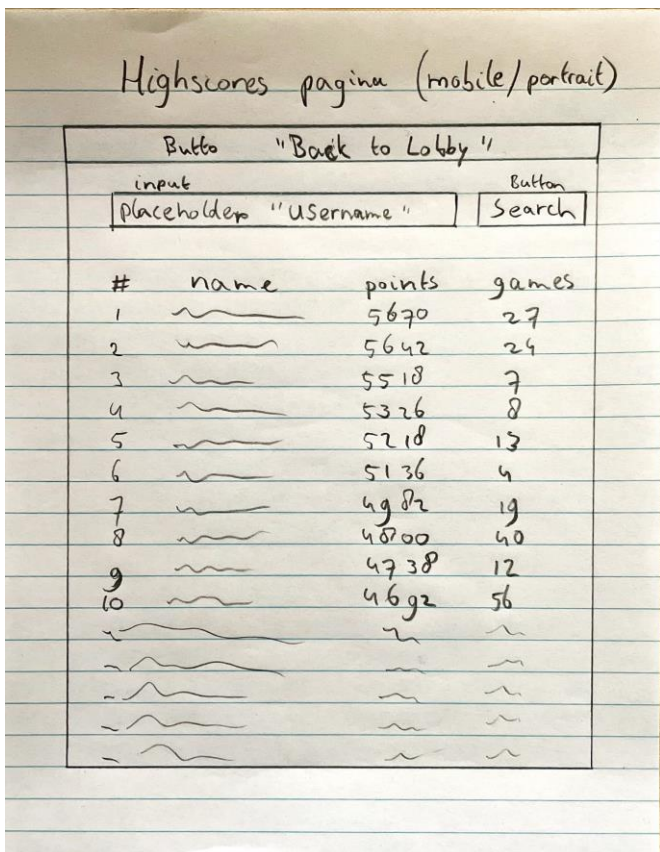
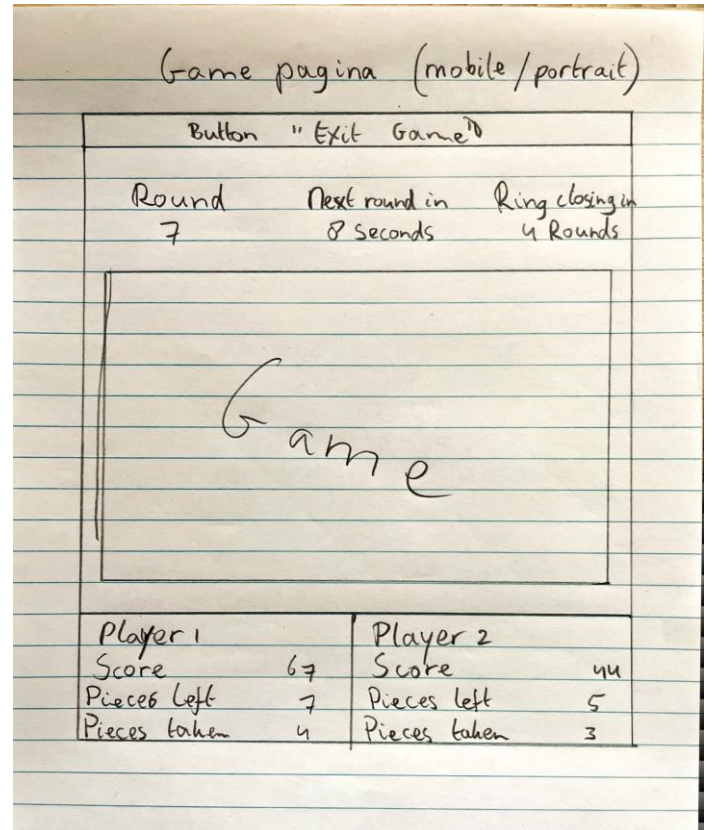
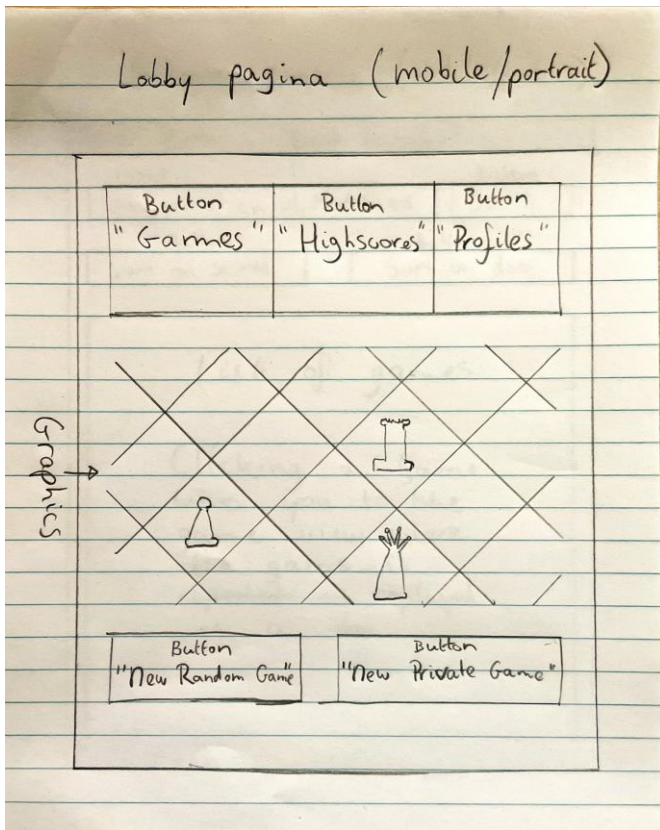
Use Case 1	Spel starten
Actor	Speler
Trigger	De speler komt in de lobby
Preconditie 1	Er zijn voldoende beschikbare spelers
Preconditie 2	Elke speler heeft zich gereed gemeldt
Main success scenario	Het spel wordt gestart en de spelers kunnen dit gaan spelen.
Postconditie	De spelers zijn het spel aan het spelen

Use Case 1	Highscores van een speler bekijken
Actor	Speler/toeschouwer
Trigger	De speler/toeschouwer komt op highscores pagina
Preconditie 1	Er is gezocht op een spelersnaam
Preconditie 2	De gezochte speler bestaat
Main success scenario	<ol style="list-style-type: none"> 1. De score van de gezochte speler wordt opgehaald 2. De score wordt getoond
Postconditie	De score van de speler is te zien

Use Case 1	Een live spel bekijken
Actor	Toeschouwer
Trigger	Toeschouwer komt op de applicatie
Preconditie 1	Toeschouwer kiest een spel om te bekijken
Preconditie 2	Het spel is nog niet afgelopen
Main success scenario	De toeschouwer kan meekijken met het spel zoals het gespeeld wordt
Postconditie	De toeschouwer ziet het live spel en de zetten die gedaan worden

Use Case 1	Profiel van een speler bekijken
Actor	Speler/toeschouwer
Trigger	De speler/toeschouwer komt op Profiel pagina
Preconditie 1	Er is gezocht op een spelersnaam
Preconditie 2	De gezochte speler bestaat
Main success scenario	Er wordt een speler gevonden
Postconditie	De speler/toeschouwer kan de gegevens en spelstatistieken van de gezochte speler bekijken

Wireframes



Games pagina (mobile/portrait)

Button "Back to lobby"

input

Filter games on id, players etc

Button

Search

Button

Sort on scores

Button

Sort on date

List of games

Clicking a game
takes you to the
game view where
the game is
~~repeated~~ as replayed
as a video

Verantwoordingsdocument

Inleiding

In dit verantwoordingsdocument zet ik uiteen welke keuzes ik heb gemaakt bij het ontwikkelen. De app is ook nog niet af, ik ga de game nog verder met ontwikkelen en zal hem later online zetten.

NPM

NPM package momentjs

Momentjs heb ik gebruikt voor het formatten van tijd. Het is een grote library waar erg veel mee mogelijk is en die veel gebruikt en goed onderhouden wordt. Ik heb eerder met momentjs gewerkt en er ook uitgebreidere dingen mee gedaan dan in deze app en bijna alles is mogelijk. De documentatie zou wel iets uitgebreider mogen, want het is vaak nog wel even zoeken.

NPM package react-table

React table heb ik gebruikt voor het weergeven van data. React-table is erg makkelijk te gebruiken en biedt veel mogelijkheden. React-table gebruik ik een volgende keer weer.

Npm package react-draggable

React-draggable wordt gebruikt om het spelbord te kunnen slepen. Dit is nodig om op kleine schermen ook op een groot bord te kunnen spelen. Samen met de mogelijkheid tot inzoomen kan er daardoor goed gespeeld worden.

Npm package stringtime

Stringtime heb ik 2 jaar geleden gemaakt. Momentjs mist de functie om een tijdspanne in milliseconden in te geven en als resultaat te krijgen hoe lang dat is in uren, minuten, seconden en dergelijke. Dit gebruik ik om automatisch te berekenen hoe lang elke game maximaal duurt, zodat een speler weet welk spel hij kan spelen in de tijd die hij beschikbaar heeft.

Overige npm packages

De overige npm packages heb ik zorgvuldig gekozen. Meestal ga ik in een react project voor de packages die beginnen met "react", dit omdat deze vaak veel gebruikt en goed gemaintaind worden.

Keuzes die ik heb gemaakt

GameBoard

Ik heb drie keer het een gameboard gebouwd. Een keer voor het spel zelf, een keer voor het menu in de lobby en een keer voor de gameboard editor. Voor het gameboard en de gameboard editor heb ik ervoor gekozen om dit door middel van een table te doen. Normaal dient een table natuurlijk alleen gebruikt te worden voor het weergeven van data, maar in dit geval leek een table me ook een logische oplossing. Het gameboard wordt namelijk altijd opgebouwd uit regels en kolommen. Voor het mainMenu wilde ik het anders doen, gewoon met divs, om te kijken hoe ik dat kon doen.

Het lastige van een spelbord is dat alle vakjes vierkant moeten blijven, ongeacht de breedte, hoogte en eventueel resizen. Dit is ook gelukt. De gameboards hebben een vast formaat en kunnen in en uit gezoomd worden door middel van de scale aan te passen. De gameboards kunnen ook buiten de window vallen en dus gedragged worden als ze te groot zijn.

Het mainMenu is opgebouwd uit divs en pas het aantal rijen en kolommen aan aan de grootte van de window. Derhalve zal dit menu altijd volledig zichtbaar blijven. Het leuke is dat als de window geresized wordt, het menu opnieuw opgebouwd wordt zodat het altijd past.

Context

Er wordt gebruik gemaakt van meerdere contexts. De contexts die gebruikt worden om data uit api calls beschikbaar te stellen hebben een functie die deze data laadt via een api request in service files. Op zich werkt dit goed, ik kan zo 1 context gebruiken voor meerdere views die zelf kunnen bepalen hoe vaak data gerefreshed wordt. Wellicht wil ik dit later nog uitbreiden, zodat de views via een property de interval tijd kunnen aanpassen en de context zelf de interval beheerd.

Chess.com

Bij het afronden van het project kwam ik erachter dat ik ook gebruik moest maken van een openbare api. Hier heb ik eerder overheen gelezen. Gelukkig heeft chess.com een openbare api en is het mogelijk daar profielen en statistieken van spelers te laden. Spelers kunnen bij het aanmaken van een account hun chess.com username opgeven en zo zijn deze gegevens zichtbaar op hun Battle-Royale-Chess profiel. Deze gebruikersnaam kunnen ze ook later nog opgeven of aanpassen.

Spectator

Het is mogelijk om met spellen mee te kijken. Wanneer er als spectator wordt ingelogd is er op de lobby pagina onder het menu een lijst met games te zien. Hier kan ervoor worden gekozen om mee te kijken met games die op dat moment gespeeld worden. Op dit moment is het alleen mogelijk om van het gespeelde games het eindresultaat te zien, maar later kunnen deze games ook (versneld) teruggekeken worden.

Wat ik nog wil doen

Language

Er is al een language context aanwezig en er kan al gekozen worden tussen engels en nederlands op de settings pagina, maar ik heb nog geen language files gemaakt. Dit wil ik sowieso gaan doen voor ik de game lanceer. Ook wil ik dan meerdere talen gaan ondersteunen.

Email validatie

Op dit moment is email validatie nog niet aanwezig. Natuurlijk wil ik dit wel gaan toevoegen, zeker omdat er een shop in de game zit en spelers hun statistieken kunnen bijhouden.

Shop

De shop moet er voor gaan zorgen dat de game minimaal kostendekkend wordt. Het wordt mogelijk voor elke speler om een eigen kleur te kiezen, te kiezen uit verschillende schaakstukken en een andere kleurstelling van het bord.

Games terug kijken

Het moet mogelijk worden om gespeelde spellen terug te kijken. Hier moet de speelsnelheid aangepast kunnen worden en moet er gekozen kunnen worden om naar verschillende rondes te kunnen springen.