Schaken functioneel ontwerp



Gemaakt Door: Sven Keimpema

Klas: AvicosC

**Inleiding**

Schaken is een spel dat wereldwijd door allemaal mensen wordt gespeeld. Schaken heeft altijd fascinatie opgeroepen voor nieuwkomers en professionele spelers. Met de technologieën zoals rust en Tauri kunnen we tegenwoordig een cross-platform applicatie bouwen met redelijk gemak en ervoor zorgen dat onze applicatie altijd snel blijft. Het functioneel ontwerp richt zich op het ontwikkelen van een schaak API in rust met een GUI-component met behulp van Tauri De API zal functionaliteit toevoegen voor het manipuleren van het schaak spel zoals het genereren van bewegingen, bewegen van stukken, controleren als de game gewonnen is en ongedaan maken van bewegingen. De GUI zal een vrij wel een eenvoudige gebruikersinterface worden waarmee ze schaakspellen kunnen spelen tegen zichzelf en kennissen.

**Doelstellingen**

1. Ontwikkel een schaak API in rust die eenvoudig te gebruiken is en heel erg snel presteert.
2. Integreer de schaak API met een eenvoudige GUI met behulp van Tauri waardoor de speler het schaakspel kan spelen zonder moeilijkheden
3. Bied veel functionaliteit aan binnen de API en GUI, het moet kunnen voldoen aan zetten genereren, zetten maken e
4. Voldoende inline documentatie zodat mensen weten wat de functies doen wanneer ze de API proberen te gebruiken

**Doelgroep**

Dit functioneel ontwerp is gericht op zowel eindgebruikers en softwareontwikkelaars. Natuurlijk is vooral voor de softwareontwikkelaars de API het meest belangrijk waarbij de API supersnel moet zijn en vrij wel makkelijk te gebruiken. En voor de ein

**Functionele vereisten**

1. Spelbeheer: bewegen van stukken, ongedaan maken van bewegingen, controle als game is gewonnen en generatie van bewegingen.
2. Spel Interface: 8x8 schaak bord waarbij het front-end interacteert met de backend, hierbij hoort elke schaak vakje een andere kleur te hebben dan die in de rij en die erboven (zie chess.com voor een exemplaar)
3. Weergave dat de game is gewonnen wanneer het spel is gewonnen. Dit zal GEEN opnieuw/herstart knop nodig hebben. Dit komt doordat als je iets zoals multiplayer zal willen implementeren zal het compleet anders gedaan moeten worden en ik vind het persoonlijk niet nodig voor dit exemplaar
4. Om het programma simpel te houden kan je als je op het Z knop klikt wordt de zet/beweging ongedaan gemaakt.

**Niet functionele vereisten**

1. Een herstart knop is niet nodig voor het spel. Want als je iets zoals multiplayer implementeert vind ik dit vrij wel onveilig doordat iedereen dit kan aanroepen
2. Naast dat het programma WEL heel erg snel moet zijn hoeft het NIET ingebouwd een AI hebben, dit wordt gezien dat iemand die de API wil gebruiken zelf zal moeten implementeren. Er zijn hier meerdere redenen voor maar de vormelijkste is dat er een heleboel verschillende soorten manieren zijn om een AI te implementeren.
3. Multiplayer is ook niet nodig, de API is hier het meest belangrijk een GUI implementeren is alleen een test dat de API ook werkt.

**Slot**

Het schaakprogramma zal zowel functioneel rijk zijn als technisch robuust. De vormelijkste focus van het programma zal de API zo snel mogelijk te maken en nog steeds vrij wel simpel te gebruiken als API.