# Examensarbete JSD-23: Gloomhaven Helper

## Bakgrund

Projektet som kommer att fungera som mitt examensarbete är en applikation för ett brädspel kallat ”[Gloomhaven](https://boardgamegeek.com/boardgame/174430/gloomhaven)”.

***“Gloomhaven*** *is a game of Euro-inspired tactical combat in a persistent world of shifting motives. Players will take on the roles of wandering adventurers with their own special sets of skills and their own reasons for traveling to this dark corner of the world. Players must work together out of necessity to clear out menacing dungeons and forgotten ruins. In the process, they will enhance their abilities with experience and loot, discover new locations to explore and plunder, and expand an ever-branching story fueled by the decisions they make.”*

Jag spelar detta brädspel tillsammans med vänner som jobbar inom (webb)utveckling. Under utbildningens gång har jag således diskuterat med dessa vad ett rimligt projekt skulle kunna vara för att lära mig så mycket som möjligt. Idén om att göra en applikation som underlättar och förstärker denna delade brädspelsupplevelse sprang således därifrån.

## Syfte

Syftet med projektet är att jag ska lära mig så mycket som möjligt om webbutveckling i största allmänhet men också om hur större projekt kan te sig ute i arbetslivet. Då projektet kommer att vara ett fullstack-projekt så kommer jag att röra en del på både frontend och backend, varför jag tycker att detta projektet är optimalt för att lära mig och repetera hela paketet.

Bortsett från lärandet och min nuvarande utbildning så hoppas jag även kunna skapa någonting som jag och mina vänner kommer ha faktiskt nytta av, nämligen en smidigare och mer rik brädspelsupplevelse.

## Mål och koncept

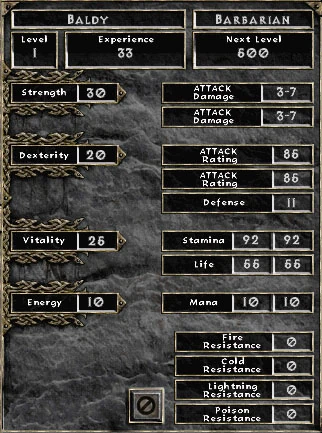
**Mål och planering (proof of concept):**

Jag hoppas kunna skapa en applikation som gör Gloomhaven roligare, enklare och mer rikt. Gloomhaven är ett stort spel med mycket information i form av flera böcker, bland annat regelbok, storybok, karaktärsavsnitt, mm.

Det jag hoppas kunna göra med min applikation är helt enkelt att låta användaren ta till sig och använda sig utav denna informationen. Jag kommer nedan att konkretisera genom att illustrera konceptet bakom det jag beskriver. Avsnitten som följer går djupare in på det praktiska, dvs. hur dessa koncept sedan ska implementeras.

Karaktären.  
*I Gloomhaven så spelar man en karaktär. Denna karaktär ”levlar” under olika scenarion, skaffar pengar och annan loot, köper items, uppgraderar sina skills och liknande. Med applikationen ska du på förhand kunna testa olika specs, olika sorters equipment, olika sorters kombinationer av vapen, etc. På så sätt kan du med hjälp av applikationen optimera din ”build”. Detta skulle jag säga är huvuduppgiften och den delen jag primärt vill implementera. För att ge Dig en så tydlig bild som möjligt så hänvisar jag till Diablo II karaktärsmenyn, vilket är idén jag i nuvarande läge har i huvudet:*

**

**

*Detta är alltså idén jag vill implementera på applikationen. Man ska kunna navigera en sådan meny.*

Regelboken.  
*Då regelboken är tjock och väldigt informationstät så vill jag underlätta för användaren att filtrera genom relevanta regler. Är man t.ex. ute efter en specifik fråga och inte vill behöva läsa igenom tjocka stycken, då ska man enkelt kunna hitta regeln.*

Användarprofil.  
*Jag vill koppla en användarprofil till sitt Gloomhaven-äventyr. Min tanke här är att återskapa sin brädspelskaraktär virtuellt. Således finns alla stats från karaktären sparad. Det är även här man kan ”komma åt” sina karaktärer och göra de testscenariorna jag nämnde i punkt 1.*

Storyn.  
*Storyn i Gloomhaven är gripande, så på något sätt vill jag få med den. Jag har inte riktigt landat i hur jag vill implementera detta, då det i slutändan inte är huvudpoängen (se punkt 1). Jag tänker mig en sektion där storyn kanske är enklare att komma åt än att bläddra i regelboken. Varje karaktärs story kan vara bunden till den karaktären man har valt, t.ex.*

## *Planering och tekniker*

**Tidsplan**

Då min tidsram sträcker sig från januari 2025 fram tills ungefär första juni så kan vi konstatera att det rör sig om cirka 5 månader. På 5 månader så ska jag komma så långt som möjligt på en brädspelsapplikation, med funktionaliteter såsom regelsökning, inloggningssystem, profilsystem med karaktärsfärdigheter och ”stats” kopplade, ett ”deck-building system som kan hjälpa en välja bättre eller sämre kort inför spelen, mm. Jag kommer i slutet av detta avsnitt återkomma med en rimlig tidsestimering på hur långt jag kan förväntas hinna på dessa 5 månader.

**Arbetsmetod**

Jag kommer att arbeta enligt ett s.k. ”agilt förhållningssätt” där jag förvisso strävar efter att ha en sån utarbetad plan som möjligt innan projektets start men att jag ändå jobbar flexibelt och med en förväntan att saker inte kommer att bli exakt som jag planerat. Med tanke på att jag dessutom gör detta parallellt med min praktik och de övriga vardagssysslor jag sysslar med så är det inte säkert när eller hur länge jag kommer sitta vid mina olika tillfällen. Därav krävs ett agilt förhållningssätt, där jag kan justera efter tid och omständigheter.

Rent konkret så kommer detta förhållningssätt att överföras i projektets github-repository. Under ”projects” så kommer min plan finnas, där jag sätter upp backloggar, planerar sprintar, iterationer, etc. Mitt mål är att regelbundet kolla och uppdatera denna och se till att följa den till högsta möjliga mån – men med den inbyggda flexibiliteten jag nämnde ovan.

**Milstolpar**

Projektet kommer att delas upp i mindre ”milstolpar”. Detta kommer vara de större målen i projektet som varje ”sprint” jobbas mot. Ett sådant exempel skulle kunna vara: ”Gör klart applikationens API”. Detta är då den större milstolpen som varje iteration/sprint kommer att jobba emot. Detta är också tydligt under ”github projects”, vad som är en milstolpe och hur sprintarna är designade runt dem.

**Tekniker och slutlig estimering:**

Fas 1: Grundläggande struktur och funktionalitet (Månad 1–2)

* Backend (Express + Mongoose)
  + Grundläggande autentisering med JWT och bcrypt (registrering, inloggning, auth-middleware).
  + Databaser med grundläggande modeller (Users, Characters, Cards, Rules).
  + API-endpoints för CRUD-operationer (skapa, läsa, uppdatera, radera).
  + Grundläggande felhantering och validering.
* Frontend (React + Vite)
  + Skapa sidstrukturen (React-router-dom: /login, /dashboard, /rules, /deck-builder).
  + Implementera användarregistrering och inloggning.
  + Grundläggande UI med Material UI-komponenter.
  + Zustand för global state management (auth state, karaktärsdata).

Fas 2: Utökad funktionalitet (Månad 3-4)

Backend

* Lägga till endpoints för filtrering av regler (t.ex. sökning efter nyckelord).
* Implementera en struktur för sparade stats på karaktärer.
* Eventuell integration av en extern API (om tillgänglig) för kortdata.

Frontend

* Implementera "deck builder" där kort kan dras/droppas och kombineras.
* Förbättra UI/UX med avancerad interaktion (drag & drop, filteralternativ).
* Hantering av användarens sparade karaktärer med CRUD-funktioner.
* Responsiv design för mobilanpassning.

Fas 3: Finjustering och optimering (Månad 5)

* Testning och förbättringar
  + Backend: Enhetstester med Jest/Supertest.
  + Frontend: Testning av UI-flöden med React Testing Library.
  + Prestandaoptimering på både frontend och backend.
  + Deployment (t.ex. Netlify för frontend, Railway/Render för backend).

Notera att detta är en *estimering*, men mitt mål är att åtminstone kunna påvisa en mindre men ändå fungerande version av samtliga koncept/idéer framställda ovan.