

Ghostwolf Dota



DOTA 2

datum: 17 december 2017

jort hoogerhuis

Projectomschrijving

Project naam: Ghostwolf dota

Project idee: Een website waar ik guides voor dota 2 kan plaatsen en waar mensen hun gameplay kunnen laten analyzen voor tips.

Project functies:

- Guides bekijken
- Lijst van recent games bekijken met average stats overview
- Analysis van recent games

Doelgroep: Beginnende Dota 2 spelers.

Plan van aanpak / Planning

Basis voor het project aanleggen

- Mongo DB opzetten
- Benodigde Mongo DB collections aanmaken
- Node JS opzetten
- Angular en Express installeren

Back-end functioneel maken

- Routing
- Database functies aanleggen
- Node JS services aanmaken

Front-end aanleggen

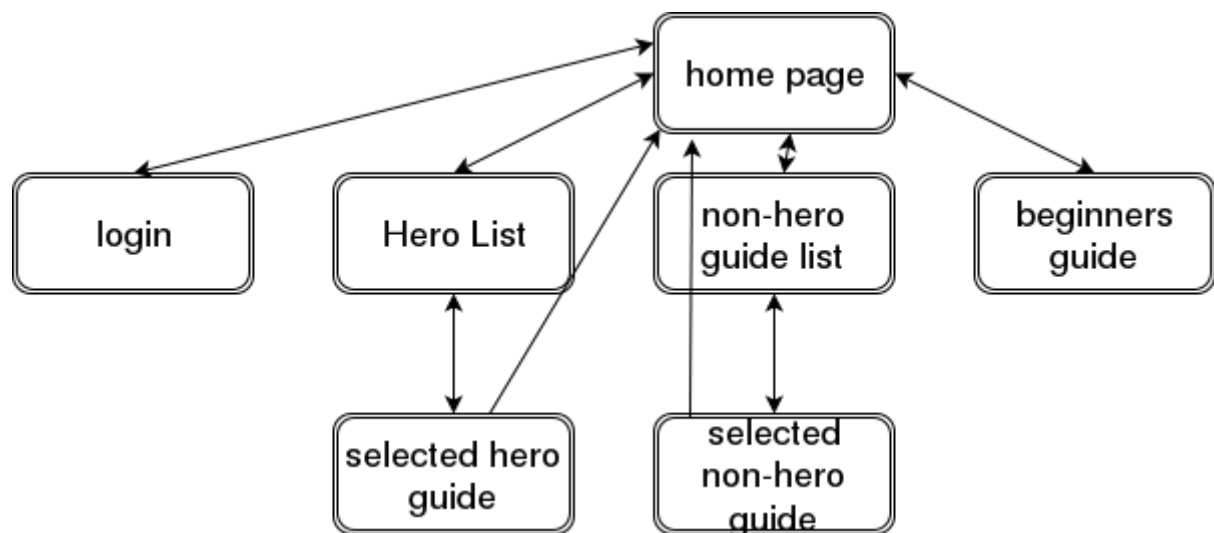
- Layout maken
- UI design
- Benodigde angular views aanmaken

- “test” content toevoegen

Fine tuning

- Bug fixing
- performance verbeteren
- Search Engine Optimalization

Website layout



Data plan

verschillende mongo collections

- Guides
- Heroes
- Abilities
- Items

Guide collection model

```
intro: {type: String, default: ""},  
skills: {type: String, default: ""},  
playstyle: {type: String, default: ""},  
itemization: {type: String, default: ""},  
drafting: {type: String, default: ""},  
heroID: {type: Number, default: 0},  
skillbuild: {type: [buildSchema], default: ""}
```

buildScema model

```
notes: {type: String, default: ""},  
title: {type: String, default: ""},  
build: {type: [Number], default: []}
```

Heroes collection model

```
name : {type : String, default: ""},  
heroID: {type: Number, default: 0},  
localized_name: {type: String, default: ""},  
primary_attr: {type: String, default: ""},  
attack_type: {type: String, default: ""},  
roles: {type: [String], default: []},  
legs: {type: Number, default: 2},  
has_guide: {type: Boolean, default: false},  
id: {type: Number, default: 0},  
stats: {  
  agiBase: {type: Number, default: 0},
```

```
agiGain: {type: Number, default: 0},  
strBase: {type: Number, default: 0},  
strGain: {type: Number, default: 0},  
intBase: {type: Number, default: 0},  
intGain: {type: Number, default: 0},  
armorBase: {type: Number, default: 0},  
speedBase: {type: Number, default: 0},  
damageBase: {type: String, default: ""}  
}
```

Abilities collection model

```
data: {type: [String], default: []},  
name: {type: String, default: ""},  
cooldown: {type: String, default: ""},  
mana: {type: String, default: ""},  
description: {type: String, default: ""},  
icon: {type: String, default: ""},  
liveIcon: {type: String, default: ""},  
lore: {type: String, default: ""},  
heroID: {type: Number, default: 0},  
id: {type: Number, default: 0}
```

Items collection model

- Name
- Cost
- Lore
- Description
- Data
- Recipe
- Icon
- LiveIcon

Technisch ontwerp

Node JS voor de back-end
Angular voor de front-end
Mongo DB als database
Express als application manager

python voor hulp applicaties
zoals voor de scripts die de database updaten als er een patch is en het uploaden van guides.

Een “kleine” server zou genoeg moeten zijn om de website te kunnen draaien. Vanwege dat de website niet data heavy is.

Functioneel ontwerp

- Guides browsen
- Guides bekijken
- overzicht van gespeelde games kunnen opvragen
- overzicht van recent games opvragen
- Analysis over gespeelde games opvragen
- Analys over recent games opvragen
- Database gemakkelijk updaten als er een game patch is
- Guides uploaden is alleen voor mij, dus word gedaan via python scripts en hoeft geen implementatie voor in de front-end

Ontwikkel omgeving

Elke text editor met support voor:

- Html
- CSS
- Python
- Javascript
- JSON

op een linux OS.

Userinterface Layout

Nav-bar

Login

content

Gegevens verzameling

de databases zijn gevult met de bij behorende data, met behulp van een python script.

De “hero” database heeft alle 117 heroes.

De “ability” database heeft ~90% van alle abilities.

de “guide” database heeft alle tot nu toe gemaakte guides.

De “item” database is work in progress.

Bouwfase Rapport

gestart met het main server JS bestand.

Waar de initial config plaats vind en waar de applicatie start.

Vervolgens is home page toegevoegt zodat de website een begin punt had.

Daarna is de nav-bar toegevoegt met een stel placeholder links zodat die werkend gemaakt kan worden.

Nadat de front-end een basis had met in ieder geval een place-holder page zodat het mogelijk was om dingen te displayen en de back-end te testen.

Is het eerste deel van de back-end toegevoegt de API voor de database, zodat de front-end met de database kon communiceren.

Tot nu toe ging alles redelijk soepel omdat ik hier al eerder mee heb gewerkt.

Nadat de front-end bij de database kon moest er data in de database.

Hier voor is een python web-scraper ontwikkeld die de benodigde data van de officiële website afhaalde en de juiste files download.

Site: <http://www.dota2.com/heroes/>

vervolgens stuurt het deze data door naar de applicatie die het via de eerder

gebouwde api in de database insert via een validation schema zodat de data altijd correct ge-parsed is.

Achter af gezien was het een beter idee geweest om op een of andere manier deze data direct uit de game files uit te lezen want de officiële site is niet 100% compleet

voor de “items” database zal een 2e web-scraper gebouwt moeten worden of als nog via de game files gedaan worden deze is nog work in progress.

Om de data aan van de database aan de front-end te connecten is er op de back-end per database een service aangemaakt in de NODE js server.

Nu er data is om mee te werken moet een deel van die data gedisplayed worden op de front-end, als eerste een lijst van alle heroes zodat mensen de guide voor de hero die ze zoeken kunnen vinden, deze lijst is sortable bij naam, main attribute en attack type en heeft een search.

Hier voor is een aparte controller aangemaakt in de Node JS server.

Nu er een home page is en de guides browsable zijn moet er nog een manier toegevoegt worden om de guides te bekijken.

Hiervoor is een controller in de server aangemaakt en een template waar de guides in worden in geladen. De guides zijn opgedeeld in verschillende tabs en elke tab word los in geladen. Alle guides staan in de database en worden direct vanuit daar in geladen.

Om dit process soepel te laten werken en de CSS in de goeie richting te krijgen ging moeizaam het rescalen van de browser levert nog steeds wat problemen op. En dus waarschijnlijk ook op mobile devices.

Testverslag

Home page test:

werkend en redelijk tevreden over de content erop en de styling alleen de nav-bar heeft nog wat werk nodig.

Hero List test:

hero search dropdown niet correct geplaatst maar voorde rest prima

Guide Test:

CSS van de “skills” tab is niet helemaal ok
de “my games as” tab is te lang en on overzichtelijk
de “analysis” tab werkt niet

de “items” tab mist nog item hover over descriptions vanwege dat de item
database WIP is.

overhet algemeen is de guide te onleesbaar een voelt aan als een saaie
muur van text.

Evaluatie verslag

over het algemeen was ik redelijk tevreden over hoe het verlopen is maar ik denk wel dat ik beter had kunnen kijken naar opties en wat meer dingen had kunnen proberen, zoals data direct uit game files lezen en misschien verder kijken naar Node JS libraries zodat werk aan mijn kant lichter is.

Ook denk ik dat ik met de CSS meer en eerder hulp had moeten vragen ik denk dat ik daar telang op ben blijven haken. Ook omdat ik dat lastig vind.

Verder is er misschien nog wat aan temerken op mijn tijds besteding. Mijn focus en “drive” om iets effectief te doen en het hoofd er bij te houden is vaak niet 100%, onderander is mijn focus de afgelopen 2 weken afgedwaald naar een java project, een mod voor een game, whoops.

Na een tijd aan het zelfde project te werken 1-3 weken, krijg ik er meestal een burnout van en gaat de motivatie om er aan verder tewerken de grond in, totdat ik later terug kom met nieuwe.

Implementatie plan

Installeer de MEAN stack op een server device die 24/7 kan draaien.

Run de python scripts on de databases bij te werken.

Haal een domein naam en host de server erop.
Test of de server bereikbaar is.
Done.

Implementatie verslag

Work in progress.