

Programvaruutvecklingsplan

Projektets namn: Diagra

Version: 0.5

Datum:

Författare: Björn Granholm

Changelog:

0.1 -> 0.2

Filstruktur

0.2 -> 0.5

Finslipa mål

1. Inledning

1.1 Syfte

I grund botten ett system som tolkar ett diagram och graf-språk som via en tolk genererar output i svg för visualisering och vidare bearbetning.

Detta skall kunna genereras med ett cli

1.2 Mål och omfattning

- Huvudmål är att få ett enkelt gränsitt tolka språket och generera output i svg
- Ytterligare utveckling sker modulärt från denna grund
- Språkets syntax skall vara läsbart
- Detta verktyg och språk ska kunna utnyttjas av olika sorters system och kunna användas i lag via versionkontroll
- Språket öppen källa.
- Eventuellt göra det möjligt att använda som bibliotek.
- CLI /TUI
- Eventuellt en TCP server som man kan få tillgång med via ssh eller netcat och liknande
- Sista steg om tiden räcker till. kunna generera ett AST av nånting. typ en repo

1.3 Moduler

- Terminal gränssnitt
- Kunna generera utifrån information såsom excel dokument, databaser och JSON.
- Generera HTML/CSS
- Generera

1.4 Referenser

2. Projektorganisation

2.1 Team och roller

Björn Granholm är huvudutvecklare till att börja med men är öppen att jobba i ett mindre lag, rollfördelning sker då det händer.

2.2 Ansvar och befogenheter

Björn Granholm har huvudansvar. Om fler lagmedlemmar förvävas delas ansvaret upp.

3. Teknisk Översikt

3.1 Systemarkitektur

- Språket liknar en funktion för diagram
- Output genereras i SVG
- TUI
- Eventuellt ett terminalgränssnitt via tcp server

3.2 Teknologier och verktyg

Huvudmodul:

- Go
- Charm biblioteket för ett terminalbaserat gränssnitt

3.3 Integrationer

Integrationer kommer att ske vid behov, i nuvarande läge inget men senare kanske det behövs tredjeparts verktyg för att spara tid

4. Utvecklingsprocess

4.1 Metodik

- Anpassat Agilt för små lag och soloutvecklare
- Kanban för moduler

4.2 Kvalitetssäkring och testning

Pröva enhetstestning med Go's test ramverk. Om omständigt överge och kör med snabb kompilering med go run och loggning

5. Filstruktur

```
|— assets
|  |— info.md
|— cmd
|  |— info.md
|  |— main.go
|— example
|— go.mod
|— interpreter
|  |— info.md
|  |— lexer.go
|  |— parser.go
|— renderer
|  |— info.md
```

```
|— runtime
|  |— info.md
|— test
|  |— info.md
|  |— lexer_test.go
|  |— parser_test.go
```

info.md i mappar innehåller information av filer och todo

6. Riskhantering

Identifiera potentiella risker och hur de ska hanteras.

- om tcp server realiseras behöver det finnas nån slags sandlåda och bra sanering pga säkerhetsskäl

7. Kostnadsuppskattning och Resursplan

Tid, svett och tårar.

8. Underhåll och Support

Dikteras av motivation och intresse.