Contents

1 Indledning

Der skal designes og implementeres en 2D game engine.

2 Forord

Tak til folk...

3 Tidsplan

Der bliver ikke lagt en tidsplan. Der bliver lagt en plan der ikke er begrænset af tid. SCRUM måske?

4 Problemformulering

Der skal designes en simpel Game Engine som man kan arbejde ud fra når der bliver lavet spil.
Der skal nemt kunne laves elementer i et vindue, give elementer forskellige attributter
(som lyd, kollision, kamera) og man skal kunne bevæge elementerne i forhold til deres relative position. \$\

Det skal være en Game Engine der kan tage i mod et TileMap og bruge det som map.

5 Systemvision

5.1 Funktionelle krav

Kan håndtere oprettelse og manipulation af et vindue Kan tilføje ubegrænsede elementer til vinduet og prioritere hvilken rækkefølge det skal vises i. Kan afspille lyd på de tilføjede elementer. Kan håndtere kollision i mellem elementerne. Det skal kunne vise de elementer i en manér der ikke er komprimerende for performance.

5.2 Minimumskrav for performance er:

Et baggrunds element. Elementer der kører ovenpå baggrunds elementet, min. 10 elementer. Netværks håndtering (skal måske være en pakke for sig).

6 Ikke-funktionelle krav

GUI.

7 Teknologi

- Hvilken teknologi (web, windows, mobil, touch, rfid osv) og hvorfor?
- Windows 10.
- java.
- Laptop preferably.
- Swing.

- Awt.
- Keyboard og mus.

8 Domænemodel

9 Arkitektur

Komponent arkitektur, hver komponent kan have egen arkitektur til de behov som komponenten nu har brug for. hver ting fungere for sig, de kan dog sættes sammen på en eller anden måde.

design-klassediagram

10 Kodestandarder

- Tuborg klammers position:

```
public void method(...)
{
// method body.
}
```

- One liners til simple metoder såsom getters og setters:

```
T member = null;
public T getMember() { return member; }
public void setMember(T member) { this.member = member; }
```

- Ternære operationer er tilladt:

```
int foo = 1;
int bar = 1;
public boolean ternary() { return foo == bar ? true : false; }
```