**Galaxy Cards**

**Grupp 33**

**Designdokument**

**V. 1.0**

**11/04-16**

# Dokumenthistorik

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Version | Beskrivning | Författare |
| 160411 | 1.0 | Första utkast av designdokumentet. Syfte och klassdiagramm över servern infogas till nästa version. | Patrik Lind, Patrik Larsson, Jonathan Wahlstedt och Emil Sundberg |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Innehåll**

[Dokumenthistorik 2](#_Toc448144712)

[Designdokument 4](#_Toc448144713)

[Syfte 4](#_Toc448144714)

[Ordlista 4](#_Toc448144715)

[Systemdiagram 5](#_Toc448144716)

[Användningsfallsdiagram 6](#_Toc448144717)

[Scenarion/Användningsfallsbeskrivningar 6](#_Toc448144718)

[Användargränssnitt 7](#_Toc448144719)

[Klassdiagram 9](#_Toc448144720)

# Designdokument

# Syfte

<Text som beskriver syftet med dokumentet. Vad det innehåller och hur det relaterar till projektet. Det kan också beskrivas vad som inte står här om det förtydligar vad syftet är eller inte är.>

# Ordlista

**Unit:** Kort av typen Unit representerar rymdskepp som användaren kan använda för att initiera en attak eller försvara med när motståndaren attakerar.

**Heroic Support:** Kort av typen Heroic Support representerar en förlängning av hjälten. Dessa kort har speciella förmågor som kan vara fördelaktiga för spelaren eller förödande för motståndern. Heroic Support kort kan inte användas för att attakera med men de har defense attribut för att man ska kunna förstöra de genom att använda förmågor eller attakera med units.

**Tech:** Kort av typen tech har speciella förmågor likt Heroic Support kort men dessa förmågor används endast en gång vid spelande av kortet. Kortet förbrukas sedan och hamnar på skroten.

**ResourceCard:** Kort av typen Resource ger hjälten 1 permanent resurs när de spelas ut. Kortet förbrukas och hamnar på skroten. Man kan bara spela ut ett resurskort varje runda.

**Resurs:** Att spela ut kort från handen kräver att man ska kunna betala för kortet. När man betalar för kortet förbrukas hjältens resurser och återställs först vid uppstart av en ny runda.

**Hjälte:** Hjälten har liv, energisköld och resurser. Det är detta objekt som är den primära måltavlan för att vinna matchen (när motståndarens hjältes liv blir 0).

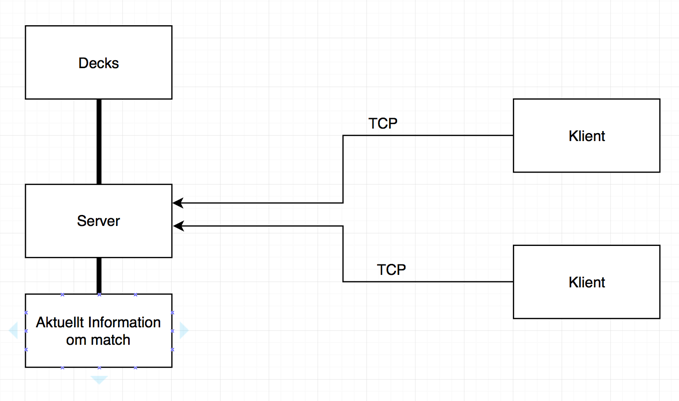
**Fält:** Varje spelare har två fält att placera units på, ett offensivt fält samt ett defensivt fält. Kort som placeras på det offensiva fältet kan spelaren attakera med men inte försvara med och vice versa. Varje fält kan hålla maximalt 6 units.

**Skrot:** En behållare som visar korthistorik, dvs de kort som är förbrukade. Kort kan vara förbrukade vid användning (gäller för Tech och ResourceCard kort) eller när de förstörs ( gäller för Unit och Heroic Support kort).

**Förmåga:** Förmågor gör något unikt vid aktivering som tex: dra ett kort, gör X i skada på måltavlan eller reparera X i försvar på måltavlan.

**Area-of-effect:** Förmågor som har hela fält som måltavla och alla objekt på fältet påverkas av förmågan.

# Systemdiagram



Systemet bygger på en Klient Server arkitektur, där klienterna är relativt tunga och innehåller väldigt mycket av spelets logik. Klienten är event-driven då programmet regerar på olika val som spelaren gör under hens runda. Servern och klienten körs med skilda program vilket innebär att för att spela en match så krävs en server instans och två klient instanser.

Serven ansvarar för att skicka data mellan klienterna via TCP protokollet. Under en match kommunicerar klienterna genom servern som tillhandahåller aktuell information om matchen. Den data som skickas mellan servern och klienten är information om spelarnas val under matchens gång. Servern tillhandahåller även olika decks som spelaren kan välja mellan att spela med. När spelaren har valt en deck så skickas den till klienten.

# Användningsfallsdiagram

[Infoga bild på användnignsfallsdiagram. En handritad.]

## Scenarion/Användningsfallsbeskrivningar

<Introducerande text som beskriver hur användningsfallen förtydligas i text. Justera rubriken ovan för den typ som används. För exemepl på hur scenarion och användningsfallsbeskrivningar utformas se föreläsning F5.>

### <Namn på användnignsfall>

<beskrivning av något slag>

### <Namn på användnignsfall>

<beskrivning av något slag>

# Användargränssnitt

Hur användargränssnittet ska se ut för brädan bestämdes i tidigt stadie när gruppen designade kravdokumentet. Gruppen kom snabbt överrens att spelaren ska ha två fält att lägga ut sina unitkort på och ett särskilt fält för heroicsupport kort.

Brädan ska även innehålla plats för spelarens samt motståndarens hjälte och och kort-på-handen panel. Brädan ska även innehålla en info-panel där man visar det senast musmarkerade kortet unitkortet som befinner sig ute på spelplanen och en textruta som meddellar spelaren med olika händelser.

För övrigt ska brädan innehålla en skrot-panel där de 5 senast förbrukade korten för varje spelare ska visas.

Användargränssnittet för själva kortobjekten var inspirerad från kortspelet *Magic the Gathering* och gruppen använde deras kort som mall för att designa kortobjekten. Det som skiljer sig här är att om kortet är ett unitkort måste det ändra sitt utseende för att få plats på den begärnsade skärmytan med tanke på att det finns flera spelfält att placera korten på.

BoardPrototypeV1 visar hur spelbrädan ska se ut samt hur korten placeras på olika fält. Korten är tillfälligt placerade i JFrame behållare för att smidigt kunna flytta på objekten och anpassa prototypen.



BoardPrototypeV3 visar hur brädan ser ut när man startar en match. Knapparna i den högra panelen är tillfälliga för att man ska kunna testa funktionaliteten och kommer ersättas med en meny där användaren ska kunna avsluta rundan, ge upp m.m.

Spelfältens namn och kant synliggörs tillfälligt när spelaren behöver inteagera med fältet (vid spelande av unitkort eller användning av area-of-effect förmågor).

Textpanelen används för tillfället för debugging men där ska händelser uppdateras när systemet är färdig. Färgen och typsnittet på texten kommer att ändras beroende på händelsens natur.

Hjälteobjekt har tre progressbars som representerar hjältens liv, energisköld samt resurser. Användargränssnittet för hjälten är fortfarande ”work in progress” och kommer att få en visuell uppgradering.

Användargränssnittet för kortobjekten är också ”work in progress” och kommer få en visuell uppgradering. De kortobjekt som används för tillfället representerar inte kortens design (attribut, förmågor) utan används mer för testsyfte.

BoarPrototypeV3 kan testköras genom att starta programmet med klassen StartGameWindow i paketet guiPacket.



# Klassdiagram

Klassdiagram utvecklades med verktyget *Visual Paradigm* och är alldeles för stort för att kunna infoga som bild i detta dokument utan att påverka läsbarheten. Klassdiagrammet över klientsidan bifogas som en separat fil. Klassdiagrammet över serversidan är inte komplett ännu och bifogas till nästa tillfälle.

Om man inte har Visual Paradigm installerad finns det en gratisversion att hämta på länken: <https://www.visual-paradigm.com/download/community.jsp>

Se bilaga: GalaxyCardsClientV2.vpp

Obs: Läs den rödmarkerade inforutan för att lättare förstå associationerna mellan kortobjekten och de olika muslyssnare. Klassdiagrammet är inte helt färdig och där måste man jobba vidare med bl.a. Controller och Client klassen.