

Battleship

Grupp 3

Kravdokument

V. 2.2

2023-04-13

Dokumenthistorik

Datum	Version	Beskrivning	Författare
2023-04-13	1.0	Påbörjade utvecklingen av kravdokumentet	Bilal Ayubi Firoz Akbari Oscar Svantesson Ömer Kolsuz Fatima Kadum Hossein Khavari Bashar Hassan Adam Lahbil
2023-04-17	1.1	Skrivit in kravnivå och några av kraven för spelkoden: Spel: FK 1-1.1 FK 2	Fatima Kadum
2023-05-03	1.2	Uppdaterat FK för server samt ordlistan. FK 1-1.2 , FK 2-2.1	Fatima Kadum
2023-05-08	1.3	Påbörjat FK för hårdvara: Fk 1-1.2 Fk 2-2.1	Adam Lahbil Bashar Hassan
2023-05-09	1.4	Uppdaterat FK för spel: FK 2.1-2.2, sever: FK 1.3 Inbyggt system: Data 1-1.1	Fatima Kadum

2023-05-10	1.5	<p>Uppdaterat FK för server med krav</p> <p>FK 1.1, 1.2, 1.3, 1.4, 1.5, 1.6,1.7.1.8,1.9</p> <p>Uppdaterat FK för Inbyggda system med krav</p> <p>FK 1.1, 1.2, 1.3,1.4,1.5</p> <p>Uppdaterad FK för Hårdvara</p> <p>FK 1.3 - 1.4</p> <p>Uppdatering av kvalitativa krav</p> <p>KK1 (KK 1.1, KK 1.2, KK 1,3) KK2 (KK 1.1, KK 1.2, KK1,3) KK3 (KK 1.1, KK1.2, KK 1.3)</p> <p>Uppdatering av Ordlista</p>	Bilal Ayubi
2023-05-10	1.6	<p>Uppdaterad FK för Hårdvara</p> <p>FK 1.5</p>	Bashar Hassan Hossen Khavari
2023-05-10	1.7	<p>Uppdaterad FK för Inbyggt System</p> <p>FK 2,3,4,5,6</p> <p>Uppdaterad FK för Spel</p> <p>FK 3</p>	Firoz Akbari
2023-05-10	1.8	<p>Uppdaterad FK för Spel</p> <p>FK 1.1</p>	Oscar Svantesson
2023-05-10	1.9	<p>Uppdaterat FK 4-4.1 för Spel samt ändrat och tagit bort FK 2-2.1 från server. Uppdaterat ordlistan.</p>	Fatima Kadum

2023-05-24	2.0	Fixat så att innehållet stämmer överens med alla "delkapitel", nu visas alla upp som de ska. Felsökt och kompletterade brister.	Fatima Kadum
2023-05-24	2.1	Skrivit in FK 5 för spel.	Ömer Kolsuz
2023-05-30	2.2	Uppdaterat FK för server FK 1.2 - 1.3 Skrivit in FK 1.7 - 1.9 för server Uppdaterat FK för inbyggt system FK 1.1 - 1.2 Skrivit in FK 1.4 - 1.5 för inbyggt system	Oscar Svantesson Bilal Ayubi

Innehåll

Battleship.....	0
Dokumenthistorik.....	1
Innehåll.....	4
Kravdokument.....	5
Syfte.....	6
Ordlista.....	6
Funktionella krav.....	8
Kravnivå.....	8
Must have (M).....	8
Should have (S).....	8
Could have (C).....	8
Won't have but would like (W).....	8
Server.....	8
FK 1: Ta emot och skicka begäran från och till hemsidan och inbyggda system.....	8
FK 1.1: Hantera flera samtidiga anslutningar från klienter.....	8
FK 1.2: HTTP ska användas för kommunikation.....	9
FK 1.3: Long polling.....	9
FK 1.4: Genom servern, få en alert på hemsidan att spelet har startats.....	9
FK 1.5: Genom servern kunna skicka meddelande till arduino från hemsida.....	9
FK 1.6: Genom server kunna skicka spelinformation till hemsida.....	9
Inbyggt System.....	10
FK 1: Kunna skicka data till och från servern samt hemsidan via backend.....	10
FK 1.1: Tar emot data i form av POST request.....	10
FK 1.2: skicka data i form av POST request.....	10
FK 1.3: Skicka lämplig data.....	10
FK 1.4: Kunna hantera anslutning med sockets.....	10
FK 1.5: Klient måste hantera spelkod och anslutning samtidigt samt skicka över information.....	11
FK 2: Kunna hantera knapptryck.....	11
FK 3: Läs på LCD-display.....	11
FK 4: Höra ljud från högtalare.....	11
FK 4.1: Musik från högtalare.....	11
FK 5: Ljus från LED-remsan.....	11
FK 5.1: Röra sig runt bland LED-dioderna.....	11
FK 5.2: Träff eller inte på LED-dioderna.....	12
FK 5.3: Intro och Avslut.....	12
FK 6: Starta om spelet.....	12
Hårdvara.....	12
FK 1: Vad man behöver för en funktionell spelplan.....	12

FK 1.1: Inbyggt system.....	12
FK 1.2: En skärm med lysdioder.....	12
FK 1.3: Tryckknappar.....	12
FK 1.4: LCD-display.....	12
FK 1.5: Högtalare.....	13
Spel.....	13
FK 1: Singleplayer.....	13
FK 2: Multiplayer.....	13
FK 3: Regler.....	13
FK 3.1: Antal skepp.....	13
FK 3.2: Antal skott.....	13
FK 3.3: Skeppens placering.....	13
FK 3.4: Träff eller inte.....	13
FK 4: Resultat.....	14
Kvalitativa krav.....	14
KK 1: Prestanda.....	14
KK 2: Arduino uno minne begränsning.....	14
KK4: Server.....	14
KK5: GUI.....	15
KK6: Server.....	15

Kravdokument

Syfte

Syftet med detta dokument är att tydligt definiera de nödvändiga kapaciteterna och begränsningarna som krävs för vart och ett av delsystemen, såväl som för systemet som helhet. Genom att göra det kan vi säkerställa att systemet fungerar som avsett och uppfyller alla nödvändiga krav.

Ordlista

Arduino Ethernet Shield - är en tilläggsenhet som låter en Arduino ansluta till ett nätverk via Ethernet-kabel, vilket möjliggör internetanslutning och kommunikation med andra enheter på nätverket.

Backend - Den del av en programvara som hanterar logik, databaser och andra funktioner som inte syns för användaren.

CSS - Utvecklingsspråk som hanterar front end och design

HTML - Utvecklingsspråk som hanterar funktion av olika delar

HTTP - Är en protokoll som används för att överföra data fram och tillbaka.

HTTP Post request - ett sätt att skicka data från en källa till en mottagare över internet genom HTTP-protokollet. Den används ofta när användaren fyller i ett formulär på en webbsida eller när data behöver skickas till en server för att uppdatera information.

Inbyggda system - Inbyggda system är specialiserade datorer som är utformade för att utföra specifika uppgifter eller funktioner.

JavaScript - Utvecklingsspråk som hanterar backend för hemsida och funktion.

JSON fil - (JavaScript Object Notation) är en filformat som används för att lagra och överföra data mellan program och system.

Klient - Den del av en programvara eller ett system som interagerar med användaren eller som använder en tjänst eller en resurs som tillhandahålls av en server.

LCD skärm - En LCD-skärm (Liquid Crystal Display) är en typ av platt skärm som använder flytande kristaller för att producera bilder. Används i datorskärm, telefoner, tv skärmar etc.

Protokoll - En uppsättning regler och standarder som används för att möjliggöra kommunikation mellan olika enheter eller system

Server - En dator eller ett datorsystem som tillhandahåller resurser eller tjänster till andra datorer eller system, som klienterna.

Funktionella krav

Funktionella krav är en specifikation av de funktioner eller egenskaper som måste göras eller uppnå för att möta användarens behov. Vi har delat in kraven i server, inbyggt system, hårdvara och spel.

Kravnivå

Kraven delas upp enligt kravmodellen MoSCoW. MoSCoW-modellen är en prioriteringsmodell som används för att identifiera och hantera krav i en produkt eller ett projekt. Modellen delar upp kraven i fyra kategorier baserat på deras viktighet:

Must have (M)

Krav som är nödvändiga för att produkten eller projektet ska vara användbart och uppfylla dess syfte.

Should have (S)

Krav som är önskvärda, men inte lika nödvändiga som de som är kategoriserade som "must have".

Could have (C)

Krav som är önskvärda, men kan vänta till senare faser eller versioner av produkten eller projektet.

Won't have but would like (W)

Krav som har identifierats, men inte kommer att implementeras i den aktuella versionen eller projektet.

Genom att använda MoSCoW-modellen kan teamet prioritera kraven och fokusera på att implementera de viktigaste kraven först.

Server

FK 1: Ta emot och skicka begäran från och till hemsidan och inbyggda system

Must have

Servern ska kunna skicka och ta emot begäran både från hemsidan och till det inbyggda systemet så att kommunikation sker åt båda hållen. Servern skall även hantera meddelanden från arduinon till hemsidan och från hemsidan till arduinon.

FK 1.1: Hantera flera simultana anslutningar från klienter.

Must have

Då vi ska ha inbyggda systemet och hemsidan som klienter, så ska servern kunna hantera flera anslutningar samtidigt, speciellt om vi ska ha multiplayer.

FK 1.2: HTTP ska användas för kommunikation

Should have

Med hjälp av detta protokoll så kan kommunikation ske. Servern kan då skicka request till inbyggda system, genom backend.

FK 1.3: Long polling

Should have

Detta protokoll gör det möjligt för oss att ha en fungerande kommunikation mellan server och webbplatsen där det möjliggör för klienten att berätta för servern att den inväntar meddelande.

FK 1.4: Genom servern, få en alert på hemsidan att spelet har startats

Should have

När en spelare trycker på en knapp för att börja spela (genom backend) så kommer det en notis på hemsidan att spelet har startats. Detta då ett knapptryck registreras i arduinon och skickar det till servern som vidare skickar det till hemsidan.

FK 1.5: Genom servern kunna skicka meddelande till arduino från hemsida

Should have

När en spelare vill registrera sig på hemsidan anger hen förnamn och efternamn. Vidare så dyker en flik upp som undrar ifall det är spelare 1 eller 2 och sedan så sparas denna information för framtida skäl (highscore list). Informationen skickas vidare till servern som skickar iväg det till arduino klienten som visar det på LCD skärmen.

FK 1.6: Genom server kunna skicka spelinformation till hemsida

Should have

Servern skickar information som är lämplig för highscore list så den kan uppdateras korrekt.

Must have

FK 1.7: Ansluta Java server och klienter genom sockets.

Med tanke på att första servern fallera så skall den andra servern kunna hantera anslutningar via sockets utan problem.

Must have**FK 1.8: Server måste ha en GUI**

Servern måste ha en GUI där spelarens namn och spelarens antal försök skall skrivas ut.

Must have**FK 1.9: Highscore måste skickas till servern och sparas i en textfil**

Highscore skickas från klient och ska sparas tillsammans med namn i en textfil som visas på highscorelistan.

Inbyggt System**FK 1: Kunna skicka data till och från servern samt hemsidan via backend.****Must have**

Kommunikation mellan alla enheter ska ske via backend. Det ska kunna skickas data till och från det inbyggda systemet.

FK 1.1: Tar emot data i form av POST request**Should have**

Inbyggt system i form av arduino och ethernet shield behöver kunna ha kommunikation servern i form av HTTP post request så att data från hemsida kan skickas.

FK 1.2: skicka data i form av POST request**Should have**

Inbyggt system i form av arduino och ethernet shield behöver kunna ha kommunikation servern i form av HTTP post request för att skicka data så som knapptryckning till java servern så att det sedan vidare används.

FK 1.3: Skicka lämplig data.**Should have**

När en spelare exempelvis anger sitt namn och efternamn eller ifall ett knapptryck registreras så skall lämplig data skickas till och från inbyggda system.

FK 1.4: Kunna hantera anslutning med sockets**Must have**

Anslutning med server och klient sker genom en socket anslutning och klient skall kunna hantera det.

FK 1.5: Klient måste hantera spelkod och anslutning samtidigt samt skicka över information

Must have

Klient skall ansluta upp sig till server och hantera spelkoden samtidigt, skicka över information till java servern gällande tries och servern skall visa det i en GUI.

FK 2: Kunna hantera knapptryck

Must have

Det inbyggda systemet ska kunna hantera knapptryck, det vill säga, om en knapp trycks ned ska det registreras.

FK 3: Läsa på LCD-display

Must have

Nödvändig information till spelaren ska kunna skrivas ut på LCD-displayen och även läsas av spelaren.

FK 4: Höra ljud från högtalare

Must have

Högtalaren som sitter på spelet ska spela ljudeffekter när man rör på pekaren samt att det ska vara ljudeffekter när man skjuter ner ett skepp, olika ljud beroende på om man missar eller träffar.

FK 4.1: Musik från högtalare

Should have

Innan och efter ett spel börjar/avslutas ska man höra musik från högtalaren och när man startar spelet ska musiken sluta höras.

FK 5: Ljus från LED-remsan

Must have

När spelet startar ska det lysa blå på led-dioderna som ska motsvara ett hav av gömda skepp.

FK 5.1: Röra sig runt bland LED-dioderna

Must have

När man förflyttar sig till en annan position ska det vara tydligt vart man har rört sig och vart man står.

FK 5.2: Träff eller inte på LED-dioderna

Must have

När man skjuter ett skott på någon position ska det visas på LED dioderna om man träffade eller missade genom att ändra färg.

FK 5.3: Intro och Avslut

Should have

Det kan visas fina mönster på LED dioderna innan ett spel och efter ett avslutat spel.

FK 6: Starta om spelet

Must have

Efter ett avslutat spel ska det kunna gå för en ny eller samma spelare att vilja spela igen.

Hårdvara

FK 1: Vad man behöver för en funktionell spelplan.

Must have

För att ha något att kunna spela på krävs följande komponenter.

FK 1.1: Inbyggt system.

Must have

Arduino Uno, Ethernet Shield krävs för att vi skall ha en fungerande inbyggt system.

FK 1.2: En skärm med lysdioder.

Must have

Det måste finnas en skärm med lysdioder som visualiserar spelplanen.

FK 1.3: Tryckknappar.

Must have

Det måste finnas tryckknappar för att spelaren ska kunna skifta position på spelplanen och för att kunna skjuta vald position.

FK 1.4: LCD-display.

Must have

Det måste finnas en LCD display för att printa ut nödvändig information som ska visas för spelaren.

FK 1.5: Högtalare.**Should have**

Det borde finnas en högtalare som ska spela musik innan, samtidigt eller efter att spelet är klart då man vinner eller förlorar.

Spel

FK 1: Singleplayer**Must have**

Spelet ska kunna fungera med minst en spelare.

FK 2: Multiplayer**Won't have but would like to have**

Man ska kunna spela mot en annan spelare på den andra boxen.

FK 3: Regler**Must have**

Följande regler ska tillämpas.

FK 3.1: Antal skepp**Must have**

Det ska finnas 5 olika skepp på spelplanen. Där respektive skepp har 2-5 antal liv. Efter att ett skepp har förlorat alla sina liv ska det sänkas och antal skepp dekrementeras.

FK 3.2: Antal skott**Must have**

Den spelare som har skjutit minst antal skott och sänkt alla skepp vinner. Om det är två spelare som möter varandra har de ett skott att skjuta per omgång.

FK 3.3: Skeppens placering**Must have**

Skeppens placering ska vara randomiserad genom att använda `placeShipsRandomly()` metoden.

FK 3.4: Träff eller inte**Must have**

Om man missar sitt skott ska det visas en röd färg på LED-dioden. Om man träffar sitt skott ska det visas en grön färg på LED-dioden.

FK 4: Resultat

FK 4.1: Förlora

Must have

När man gör 45 skott förlorar man automatiskt och antalet skott och visar ett X på skärmen.

FK 4.2: Vinna

Must have

När man hittar alla 5 skepp under 45 skott då vinner man spelet. Det visas också en smile emoji på skärmen.

Kvalitativa krav

KK 1: Prestanda

Spelet ska vara responsiv och snabb på 1 ms eller mindre (med delays), och ge en smidig spelupplevelse för användaren.

KK 2: Arduino uno minne begränsning

Arduino uno minnet bör inte överstiga mer än 32 KB.

KK3: Server

Server skall kunna hantera anslutning med klient och GUI.

KK4: GUI

All data gällande hur en match gick skall visualiseras på ett GUI.

KK5: Server

Server skall kunna hantera en anslutning mellan två arduino klienter samtidigt.