Kravspecifikation

Peter Wickenberg, petwi187@student.liu.se

5 mars 2021

Version 1.2

Granskad	Namn:	Datum:
Godkänd	Namn	Datum

Dokumenthistorik

Version	Datum	Utförda ändringar	Författare	Granskare
0.1	2021-02-12	Påbörjat första utkast med vissa krav och beskrivning av system	Wickenberg, P.	
0.2	2021-02-14	Första utkast nästan klart, återkoppling från handledare behandlad	Wickenberg, P.	
1.0	2021-02-21	Första utkast klart	Wickenberg, P.	Boregrim, R., Wikström, C.
1.1	2021-03-03	Förtydligade krav 4	Wickenberg, P.	
1.2	2021-03-05	Gjorde ordval mer konsekvent och förtydligade meningar som syftade på produkten	Wickenberg, P.	

Innehåll

1 Inledning

Detta är en kravspecifikation för projektgrupp PUM7 i kursen TDDD96 VT1 2021.

1.1 Syfte

Syftet med denna kravspecifikation är att definiera krav för den produkt som projektgruppen ska skapa. Kraven är framställda av projektgruppen tillsammans med projektgruppens kund, Svenska Sjöräddningssällskapet. Kravspecifikationen är menad för kunden och projektmedlemmarna samt handledare och kursansvarig.

1.2 Omfattning

Produkten som ska skapas kallas Virtuell gimbal-kamera. Denna programvara ska kunna behandla en videoström tagen med en 180+ graders kamera. Videoströmmen ska komprimeras och skickas till en server. I detta projekt skapar projektgruppen en server i *debugging*-syfte. Servern ska visa videoströmmen samt kunna ta emot indata från en användare om hur videoströmmen ska beskäras. Det är tänkt att produkten ska vara installerad på en Raspberry Pi 3B+ som ska sitta på en drönare. Produktens värde ligger i att en liten 180+ grades kamera är lättare och kräver mindre energi än en kamera med en vanlig lins styrd av en *gimbal*. Båda dessa egenskaper är viktiga för en drönare då det tillåter drönaren att stanna i luften längre.

1.3 Definitioner

Fixhawk En Fixhawk är en hårdvarukomponent som sköter styrandet av drönaren. Fixhawken har drönarens *pitch*, *roll* och *yaw* samt drönarens koordinater.

Prioritet Varje krav har en prioritet där högsta prioritet är 1. För att ett krav med lägre prioritet ska arbetas på måste alla krav med högre prioritet vara avklarade. Krav av prioritet 1 beräknas vara klara senast den 15 maj 2021 och är projektets grundkrav.

Docker-kontainer En *docker*-kontainer är en packeterad miljö för mjukvara så att mjukvaran kan köras utan konflikter med annan mjukvara eller andra miljöer.

APStreamline Ett videoströmningsverktyg kompatibelt med Fixhawk som har öppen källkod. Version 2 av APStreamline används [1].

2 Översikt av system

Detta kapitel ger en översikt av hur systemet ska byggas och vad systemet ska vara.

2.1 Övergripande beskrivning

Produkten i detta projekt är ett program som agerar som en ersättning till en fysisk *gimbal*. Produkten ska både skapas och testas. Produkten ska installeras på en Raspberry Pi 3B+ som i sin tur ska sitta på en drönare. Produkten ska ta insignaler från en server med användargränssnitt samt en Fixhawk-komponent på drönaren. Utsignalen ska gå till servern med användargränsnittet där det ska visas en videoström.

2.2 Produktens perspektiv

Nedan sätts produkten Virtuell *gimbal*-kamera i perspektiv mot andra komponenter i systemet som helhet.

2.2.1 Användargränssnitt

Ett användargränsnitt ska finnas där användaren kan se en videoström från kameran på drönaren. I projektet skapas detta på en server som endast har felsökningssyfte. Servern ingår därför inte i produkten. Användaren ska även kunna bestämma i vilken riktning videoströmmen ska beskäras. Alternativen är mot ett par koordinater eller mot ett väderstreck.

2.2.2 Mjukvarugränssnitt

Mjukvarupaketet dronekit-python som använder mavlink protokollet ska användas för att kommunicera mellan Fixhawk och systemets Raspberry Pi 3B+. Version 2 av dronekit-python används. [2]

2.2.3 Kommunikationsgränssnitt

Representational state transfer API används för kommunikation mellan drönaren och servern. REST API används för att skicka data från klienten till dess server samt för servern att ge respons till klienten med *JSON*-objekt. I detta fall är produkten en klient på servern.

2.2.4 Minnesbegränsningar

Systemets minnesbegränsning ligger i Raspberry Pi 3B+ vars RAM-minne är 1GB LPDDR2 SDRAM vilket kan begränsa hastigheten av beräkningar.

2.2.5 Begränsningar

Systemet använder en 12-megapixel kamera som begränsar upplösningen av bilden. Eftersom bilden är tagen på distans från en drönare och mer än halva bilden ska bli beskärd försämras kvaliteten på bilden kraftigt. Systemet har även begränsad tillitlighet då 4G-upkopplingen som ska användas kan förlora signalen då drönaren flyger över områden med dålig täckning. Serverns nätverksuppkoppling begränsar även signalen till systemet samt till användaren.

3 Krav

Detta kapitel definierar krav för systemet. Varje krav har ett unikt nummer samt en prioritetsnivå.

3.1 Funktionella krav

Nedan beskrivs systemets funktionella krav.

3.1.1 Användargränssnitt

Användargränssnittet skapas endast ur felsökningssyfte.

Krav nr.	**Beskrivning**	**Prioritet**
Krav 1	Användargränssnittet ska visa en videoström från systemets kamera.	1
Krav 2	Användargränssnittet ska kunna skicka indata till produkten i form av ett *JSON*-objekt från användaren för att fästa kameran till en koordinat.	1
Krav 3	Användargränssnittet ska kunna skicka indata till produkten i form av ett *JSON*-objekt från användaren för att fästa kameran till ett väderstreck.	1
Krav 4	Användargränssnittet ska ha ett minimalt GUI, så att funktionaliteten som specificeras i krav 1, krav 2 och krav 3 kan upfyllas.	1
Krav 5	Användargränssnittet ska kunna visa en tvådimensionell karta med drönarens koordinater.	2
Krav 6	Användargränssnittet ska kunna välja storleken på hur mycket videoströmmen ska beskäras.	2

3.1.2 Mjukvarugränssnitt

Krav nr.	**Beskrivning**	**Prioritet**
Krav 7	Produkten ska kunna få systemets rymdkoordinater som indata från systemets Fixhawk. Rymdkoordinaterna ska bestå av: latitud, longitud och höjd.	1
Krav 8	Produkten ska kunna få systemets *roll*, *pitch* och *yaw* som indata från systemts Fixhawk.	1

3.1.3 Kommunikationsgränssnitt

Krav nr.	Beskrivning	Prioritet
Krav 9	Produkten ska kunna få indata i form av ett *JSON*-objekt från en server om hur videoströmmen ska beskäras.	1
Krav 10	Produkten ska kunna skicka en komprimerad videoström till en server. Videoströmmen ska vara kodad i formatet H.264.	1
Krav 11	Produkten ska kunna koppla upp sig till en server med ett 4G-modem.	1
Krav 12	Produkten ska kunna skicka systemets koordinater till en server.	2

3.1.4 Systemkrav

Krav nr.	Beskrivning	Prioritet
Krav 13	Systemet ska kunna beskära en 180+ graders videoström till en 640x480 pixlar stor videoström.	1
Krav 14	Produkten ska vara kompatibel med en Raspberry Pi 3B+ som kör Debian OS.	1
Krav 15	Produkten ska kunna beskära videoströmmen så att ett par koordinater som systemet får in alltid befinner sig vid mitten av videoströmmen.	1
Krav 16	Produkten ska kunna beskära videoströmmen så att ett väderstreck som systemet får in alltid befinner sig vid mitten av videoströmmen.	1
Krav 17	Servern ska kunna köras i en textitdocker-kontainer.	2
Krav 18	Systemet ska kunna beskära en 180+ graders videoström till en videoström som är beskärd med en upplösning enligt indata till systemet.	2
Krav 19	Produkten är kompatibel med programvaran APStreamline.	2

3.2 Prestandakrav

Krav nr.	Beskrivning	Prioritet
Krav 20	Videoströmmen får ha en maximal latens på 2.0 sekunder, mätt från att videoströmmen är tagen till att användaren ser videoströmmen.	1
Krav 21	Produkten ska inte förbruka så mycket ström att drönarens flygtid minskas med mer än 2/5 av flygtiden innan produkten integreras i systemet.	1

3.3 Systemattribut

Krav nr.	Beskrivning	Prioritet
Krav 22	Server-komponenten till systemet ska vara modulär så att produkten inte behöver ändras om servern till systemet byts ut. I detta fall behöver endast in- och utformaterings modulerna anpassas efter den nya servern.	1

Referenser

[1] A. Dhamija, Dec 2020. [Online]. Available: https://github.com/shortstheory/APStreamline

[2] T. Ryan, Feb 2021. [Online]. Available: https://github.com/dronekit/dronekit-python