Team MacroHard: The Perfect Selfie

Shreesha Suresha
Mary Anne Noskowski
Simranjit Singh Sekhon
Bragatheesh Sureshkumar
Beau Rampey

Intro:

The project is an integration of a drone, a video recording device, and a smartphone application that is capable of recognizing voice commands, using the drone to automatically orient a video recording device with respect to a subject, and uploading photos taken from the video device to a cloud storage media. It addresses the problem that arises when the subject is unable to take the ideal video recording or picture of him/herself due to the physical limitations posed by the subject having to physically hold the device. In order to accomplish this task, the project introduces two new innovations: interface between the drone's built in processor and the voice recognition software in the smartphone, and the use of the drone's processor to automate its movements.

Glossary of Terms:

- **Cortana** - A personal assistant created by Microsoft that mainly interacts with the user through voice commands..
- **AR Parrot Drone** - Quadcopter with inbuilt processor and smartphone communication infrastructure.
- **TCP** - Processes level communication protocol involving dedicated socket connection.
- **UDP** - Process level communication protocol without dedicated socket connection.
- **Windows 8.0 Silverlight SDK** - The software development kit develop apps on the windows phone.
- **Parrot AR Drone SDK** - used to interface the phone app with the drone. This allows us to send high level commands to control the drone.

System Architecture Overview:

There are three main components for this project: the windows phone 8.0 (and app), the AR Parrot Drone, and Cortana voice commands.

The phone is going to be the main component in this system because it serves as the interface between the Cortana voice commands and the Drone. We will be using the AR Parrot Drone SDK in order to interface with the windows 8.0 silverlight phone app. This comes with a library that allows us to implement higher level functions to command the drone. Basic commands include moving left/right/up/down, pan and turning, taking off and landing and transmitting live feed and photos. The phone app will have a live feed stream to the phone. On the live feed page we will also have two joysticks to control tilt left/right/up/down, forward/backward movement, pan left/right, and up down vertical movement. The Live feed page will feature the streaming image, two control joysticks, as well as a "capture" and "save photo" button to take

the picture and save it to a “perfect selfie” folder within the picture library. The settings page of the app will contain two sliders that set altitude and speed limits on the drone. In addition, there is a hyperlink to the phone’s setting page that allows the user to connect to the drone via wifi. The photos will be displayed on the main page in a gridded photo library. From there, users will be able to delete and share the photos on social media.

Phone/Cortana: The phone will process the voice commands, some of which will open and start running the app. The phone app will take these commands and translate them into various functions and actions. This is implemented using the Voice Command Definition file where we specify what actions correspond to specific voice commands. We will be using voice commands to navigate to different pages, tell the drone to take off, and start up the app. We are not implementing voice commands to control the direction of the drone.

Phone/Drone: The phone will translate the voice commands given into functions the drone can use. The Drone is only in charge of maintaining its flight and position, adjusting to commands given, taking a photo/video, and sending that photo/video back. Because of the small amount of processing power on the Drone, the phone will be doing all the processing. The two will communicate via a wifi connection, using both UDP and TCP. An AR Parrot Drone SDK is being used to interface with the drone. We will use it to set up the various controls (maximum height and speed, as well as setting up the wifi connections). The drone will be using its accelerometer and gyroscope to determine its relative position to the phone. This will be used to give the drone some level of autonomous positioning so it automatically faces the user and can adjust itself for the “perfect selfie.”

Requirements:

Use Case: Voice Command to Cortana
Actors: Photo Subject, Windows Phone
Precondition: Subject of photo is on the start page after starting app.
Flow of Events:

- Expected
    - Cortana indicates that it is accepting voice commands
    - Voice command is issued by subject
    - Cortana turns voice command into a specific action that app can execute
    - App executes the action specified
- Alternate
    - If voice command given is not an acceptable action that the app can execute:
        - App notifies the user that action cannot be executed
        - App prompts user for another voice command

Postcondition: Phone processes the subject’s voice command and performs the necessary action

Use Case: Tell drone To Take Off
Actors: Photo Subject, Drone, Cortana, Windows Phone
Precondition: Drone is positioned in a clear area and windows phone app is on the start page
Flow of Events:

- Expected
    - Subject issues command to drone to take off
    - Drone raises to a height previously specified
    - Drone starts transmitting live feed to photo user
- Alternate
    - Drone faces obstacles when taking off
    - Drone safely lands and notifies user that takeoff is not possible

Postcondition: Drone is in the air and transmitting live feed to photo subject

Use Case: Take Picture
Actors: Photo Subject, Drone, Windows Phone
Precondition: Drone is in the air and transmitting live feed
Flow of Events:

- Expected
    - Photo subject issues button command to take picture
    - Drone takes picture and transmits picture to phone
    - User is able to view pictures in phone picture library
- Alternate
    - Drone is unable to establish connection with phone
    - Phone returns a message saying that connection is unable to be established

Postcondition: Picture is taken and displayed on the phone picture library.

Use Case: Drone Actions in the Air
Actors: User, Windows Phone, Cortana, Drone
Precondition: Drone is in the air and transmitting live feed
Flow of Events:

- Expected
    - User issues voice command for drone to reposition itself (pan right, pan left etc.)
    - Drone repositions itself and transmits new live feed to user
- Alternate
    - Drone faces obstacle in repositioning itself
    - Drone notifies Windows Phone app that action requested is not possible.

Postcondition: Drone performs requested action in the air and is transmitting new live feed.

Use Case: Drone Landing
Actors: User, Windows Phone, Cortana, Drone
Precondition: Drone is in the air and is transmitting live feed
Flow of Events:

- Expected

        - User issues voice command for drone to land
        - Drone stops transmitting live feed
        - Drone descends until it safely lands
- Alternate
    - Drone is unable to establish connection with user
        - Windows app notifies user that connection cannot be established
    - Drone is unable to land due to obstacles
        - Drone notifies windows app that landing is not possible

Postcondition: Drone is no longer transmitting live feed and is on the ground.

Use Case: Change Speed Settings
Actors: User, Settings user interface, Drone
Precondition: User is on the settings page and sees the speed slider
Flow of Events:

- Expected
    - User slides slidebar to expected speed setting
    - Slidebar moves to requested speed setting and displays new speed on the right
    - Drone receives new speed setting
    - Drone sends confirmation of new speed setting
- Alternate
    - No alternate this should work in all cases unless connection is lost

Postcondition: Drone is unable to move faster than specified speed setting.

Use Case: Change altitude settings
Actors:User, Settings Page Interface, Drone
Precondition: User is on the phone settings page and is able to see altitude slider
Flow of Events:

- Expected
    - User moves altitude slidebar to desired altitude
    - Slidebar moves to requested altitude and displays new altitude on the right
    - Drone receives new altitude setting
    - Drone sends a confirmation to the user that altitude setting has been changed
- Alternate
    - No alternate this should work in all cases unless connection is lost

Postcondition: Drone is unable to fly higher than specified altitude

Use Case: Change Photo Distance
Actors: Photo Subject, Settings Page Interface, Drone
Precondition: User is on the phone settings page and is able to see Photo Distance level
Flow of Events:

- Expected
    - Photo Subject selects desired photo distance

        - Selected photo distance button is highlighted
        - Drone receives new photo distance setting
        - Drone sends a confirmation to the user of new photo distance setting
- Alternate
    - No alternate, this should work in all cases unless connection is lost

Postcondition: Drone will now move at the desired distance away from the photo subject before taking picture.

Use Case: Record Video
Actors:Photo Subject, Windows Phone app to Drone interface, Drone
Precondition:Drone is in the air transmitting live feed
Flow of Events:

- Expected
    - User gives drone a recording command through the button interface (click recording icon)
    - Drone starts recording and windows phone app displays recording icon
    - User gives drone a stop recording command through button interface (click recording icon)
    - Video recording stops and is stored on phone picture library
- Alternate
    - Connection is lost in between the recording
        - Recording is stopped and drone sends signal that this is the case
    - Windows app stores recording in phone picture library

Postcondition: Video recording is stored in phone picture library.

Use Case: Upload photo/video recording to Cloud
Actors: User, Windows Phone application, Microsoft OneDrive
Precondition:Photo/Video Recording is in phone picture library
Flow of Events:

- Expected
    - User tells clicks on upload picture on cloud upload page on app
    - User selects the picture/video recording that he/she wishes to be recorded
    - Photo is uploaded to the cloud storage
- Alternate
    - Wifi connection is lost
        - Phone returns message indicating that wifi connection is lost.

Postcondition: Picture/Video Recording is uploaded to Cloud

Use Case: Upload Photo/Video to Social Media
Actors: User, Windows Phone Application, Facebook or Some Other Social Media
Precondition: Photo/video is saved in the app's folder in the picture library
Flow of Events:

- Expected

        - user selects a photo on the main page of the app
        - user selects the “share photo” option
        - user chooses which channel of social media is preferred
        - If a login is required, the user logs into their account through a pop-up
        - the photo is uploaded to social media
- Wifi connection is lost, account does not exist, or not logged in
    - error message with the appropriate warning is displayed
- Postcondition: Photo/video is uploaded and posted on selected form of social media

Use Case: Drone crashes/runs into object
Actors: user, app, drone
Precondition: User is connected to drone and controlling drone
Flow of Events:

- Drone crashes or runs into something
- Drone automatically turns off the engines and stops the propellers from spinning
- Drone is still on and connected
- Warning is Displayed on the phone app
- emergency button on app is triggered

Postcondition: drone remains on but stagnant, an error message is sent and displayed on the phone and the emergency button is triggered

Use Case: Trigger Emergency By Button
Actors: User, Phone
Precondition: User is on the live feed page taking a picture and is able to click the “Emergency” button which toggles an emergency event and the drone will land.
Flow of Events:

- Expected
    - User clicks “Emergency” button and drone drops from preprogrammed height and makes a fast landing.
- Alternate
    - There is no alternate event in the case of an emergency triggered by the emergency button.

Postcondition: **The drone lands.**

Use Case: Trigger Emergency By Dropping Phone
Actors: User, Phone
Precondition: User is on the live feed page taking a picture and drops the phone.
Flow of Events:

- Expected
    - Accelerometer reads the user dropping the phone and the drone descends from it’s pre programmed height and makes a fast landing.
- Alternate

        - There is no alternate event in the case of an emergency triggered by the accelerometer sensor.

Postcondition: **The drone lands.**

Use Case: Drone starts to run low on battery
Actors: Drone, Windows Smartphone, User
Precondition: Drone is in the air and is low on battery
Flow of Events:

- Expected
    - Drone senses that it is low on battery
    - Drone transmits message to phone indicating its low on battery
    - Drone transmits battery level

- Alternate
    - Drone loses connection with phone.
        - Drone attempts landing

Postcondition: User is notified of low battery levels

Use Case: Drone runs out of battery
Actors: Drone, Windows Smartphone, User
Precondition: Drone is in the air and is at the point where battery runs out
Flow of Events:

- Expected
    - Drone sends message indicating that it is low on battery.
    - Drone starts descending
    - Phone switches to joystick screen and user is given control of drone.

- Alternate
    - Drone loses connection with phone
        - Drone automatically lands.

Postcondition: Drone is on the ground.

Use Case: Phone runs out of battery
Actors: Drone
Precondition: Drone is in the air and the phone runs out of battery while/after sending a command to the drone
Flow of Events:

- Expected
    - Drone waits for a new command from phone for up to 2 minutes
    - Upon not receiving a new command, drone lands
- Alternate

        - None, Drone will always land after 2 minutes of inactivity

Postcondition: Drone is on the ground

Use Case: Another device connects to the drone
Actors: Drone, Windows Phone. Another device with networking capabilities
Precondition: Drone is in the air and receives command from another compatible device
Flow of Events:

- Expected
    - Windows smartphone goes to emergency land button
    - User presses emergency land button
    - User sends button indicating that the drone should turn off

- Alternate
    - Drone is unable to receive user emergency land request
        - Control of drone is transferred to other user

Postcondition: Drone has landed and is off.

Use Case: Drone is taken above max altitude limit
Actors: Drone, Windows Phone
Precondition: Drone is in the air and some external event causes it to exceed its max set altitude limit
Flow of Events:

- Expected
    - Phone executes automatic landing procedure to prevent Drone from flying far away

- Alternate
    - Automatic landing procedure command fails to go through. Drone lands after 2 minutes of inactivity

Postcondition: Drone is on the ground

Use Case: Drone moves at speed greater than max speed
Actors: Drone, Windows Phone
Precondition: Drone is moving at or below a set speed
Flow of Events:

- Expected
    - Drone starts moving faster than max speed due to physical conditions
    - Drone senses that it's moving faster than max speed
    - Drone slows down and messages phone that it is moving faster than max speed

- Alternate
    - Drone loses connection with phone
        - Drone slows down to max speed but doesn't send message to phone

Postcondition: Drone slows down to max speed.

Use Case: Live feed is not transmitted
Actors: Drone, Windows Phone
Precondition: Drone is in the air and attempting live feed transmission
Flow of Events:

- Expected
    - Drone can't transmit live feed
    - User lands drone through land command
    - User resets drone
    - User commands drone to take off

- Alternate
    - Resetting drone doesn't work
        - Adverse environmental conditions are blocking wifi connection
        - User takes drone somewhere else

Postcondition: The Drone is now able to transmit live feed and drone is in the air

Prototyping:

- Prototype 1 - Windows 8.1:

  Initial prototype built using Windows 8.1, capable of Cortana voice commands and maneuvering the drone. Since there was no SDK support for the drone on Windows 8.1 it had to be completed from scratch by sending network commands. This allowed affective launch and flight, but in order to get live feed would require implementation of complex and time consuming video wrapper.
- Prototype 2 - Windows 8.0 Silverlight with AR Parrot Drone SDK support:

  In order to avoid implementing a new SDK for Windows 8.1 we chose to switch down to Windows 8.0 Silverlight. The AR Parrot Drone SDK compatible with 8.0 supports all hardware on drone. This allocates more time to make a more polished application interface and implement augmented reality, geolocation, and other advance features.

Test Cases:

- Connection to Drone:

  Send commands to test to ensure proper functionality
- Page Navigation Tests:

  Compare current state of page to expected conditions

System Models:

Technologies used:

- **Microsoft Windows Phone 8.0 SDK**
- **ARDrone2Windows SDK**
- **C#**
- **Parrot AR Drone**