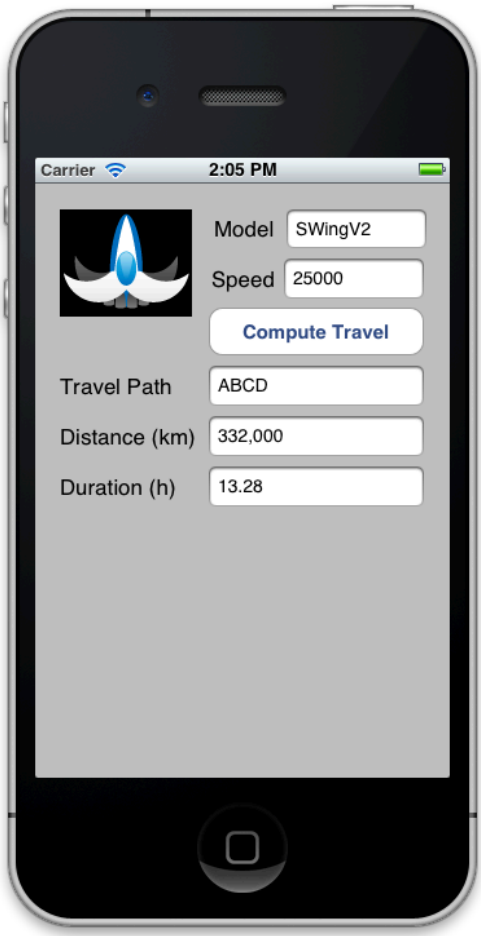
Assignment 2



Required Tasks

Design an app to allow starport operators to calculate the distances and durations of spaceflight between starports on different planets in the Andromeda solar system.

The operators need an app that allows them to quickly choose a starship and a travel path to calculate the total distance and total travel time between a series of starports. Due to ISF Starfleet Software Protocol you are required to provide modular methods and unit tests to ensure reliability of the calculations. Lives depend on accurate travel times for scheduling starships carrying food and supplies.

Ships travel at different speeds and have different docking times to unload cargo and refuel.

1. **Controller** - Create the glue between a model object and the view.
   a. The target object for actions from buttons.
   b. Contains outlets for view user interface controls (UITextField, UIImageView).
2. **View (UIView in .xib) -** Design and implement a view with the following view elements.
   a. *Model*
   b. *Speed* (No user input)
   c. *Compute Travel* button
   d. Travel Path
   e. *Distance (km)* - (No user input)
   f. *Duration (h)* - (No user input)
   g. Ship Image (UIImageView)
3. **Model (Ship Class)** - Create a class to store data for a model ship and to provide methods to calculate travel duration and distances for a given route.
   a. Speed - determined by starship model
      i. *SWingV1* - 20,000 km/h
      ii. *SWingV2* - 25,000 km/h
      iii. *SWingV3* - 34,000 km/h
      iv. *TearDrop* - 11,000 km/h
   b. Docking Time - A starship requires the following docking time at each station.
      i. *SWingV1* - 4 hours

ii. *SWingV2* - 2.5 hours
iii. *SWingV3* - 2 hours
iv. *TearDrop* - 1 hour

c. Required Methods - The following methods will help organize the calculation routines into testable chunks.
i. NSTimeInterval must be time in seconds
ii. CGFloat will represent distance in kilometers

```
- (id)initShipModel:(NSString *)model;
- (NSTimeInterval)dockingTime;
- (NSTimeInterval)timeToStation:(NSString *)toStation
                    fromStation:(NSString *)fromStation;
- (NSTimeInterval)timeForTravelPath:(NSString *)travelPath;
- (CGFloat)distanceToStation:(NSString *)toStation
                 fromStation:(NSString *)fromStation;
- (CGFloat)distanceForTravelPath:(NSString *)travelPath;
```

d. Unit Testing - You are required to implement unit tests for every method in the *Ship* class to verify that it meets specifications.
i. Test for edge cases or invalid input.
ii. Test each ship model to ensure max speed and docking times are correct.

4. Travel Path - An ordered array of letters representing different spaceport destinations.
a. Starport Stations (Linear distances)
b. **A** -> 24,000 km <- **B** -> 46,000 km <- **C** -> 225,000 km <- **D** -> 37,000 km <- **E**
c. AC = a travel path between Station A to Station C
i. Distance = 24,000 + 46,000 = 70,000 km
ii. Duration = 2.8 + 5 = 7.8 hours
1. AC in SWingV2 will take 70,000 / 25,000 = 2.8 hours
2. SWingV2 will take 2.5 hours to refuel/dock at each station = 5 hours
d. ABAB = a travel path between Station A -> B -> A -> B
i. Distance = 24,000 + 24,000 + 24,000 + 24,000 = 96,000 km
ii. Duration = 3.84 + 10 = 13.84 hours
1. ABAB in SWingV2 will take 96,000 / 25,000 = 3.84 hours
2. SWingV2 will take 2.5 hours to refuel/dock at each station = 10 hours

5. No Magic Numbers
a. All constants must be declared as a static constant in your .m files.
b. All constants must have a lowercase k as the first letter in the name.
c. Declare above the @implementation portion of a class

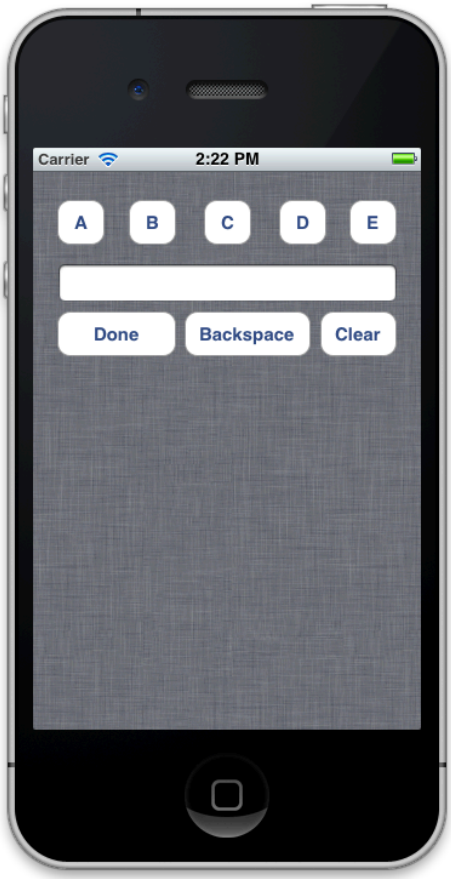
6. **Show Images**
a. A set of image resources will be provided to use in a UIImageView
b. Changing the ship's *model* should change the UIImage to the corresponding image file.

```
static const NSInteger kMaxValue = 2000;
static NSString * const kDefaultTitle = @"Default"; // different for pointers
```

7. **Format the output** - Limit floating point numbers to a max of two decimal places.

Hints

- Parse the travel path string using methods in *NSString Reference*
- Create ivars for speed, docking time, and the model.
- NSString properties use *copy*, not *strong*, to prevent issues with mutable strings.
- Use [UIImage imageNamed:@"TearDrop.png"] to load an image from your app bundle. The file must be added to your project's target.



Evaluation (100 points)

Assignments will lose points for the following:

- Build errors
- Build warnings.
- NSLog() print statements.
- Lack of Javadoc-style comments.
  - Assume the reader is unfamiliar with your code. Explain what methods do in your header files and any complex logic in your implementation files.
- Incomplete or missing required specifications.
- Code style does not match "Coding Guidelines for Cocoa."

Grading

- Style/Documentation 30
- Organization 20
- Functionality 50

Extra Credit

1. Implement a UIPicker to select the starship. Make sure to hide the picker when finished. (3 points)
2. Implement "Next textfield" switching between UITextFields, to allow seamless transitions between input areas. (2 points)
3. Implement a custom UIView/UIViewController, TravelPathViewController, to facilitate the creation of travel paths with using station name buttons. The UIViewController should be pushed modally. (5 points)
   a. Create a button in the standard interface to launch the picker.
   b.
```
[self presentModalViewController:travelPathViewController
                        animated:YES];
```