

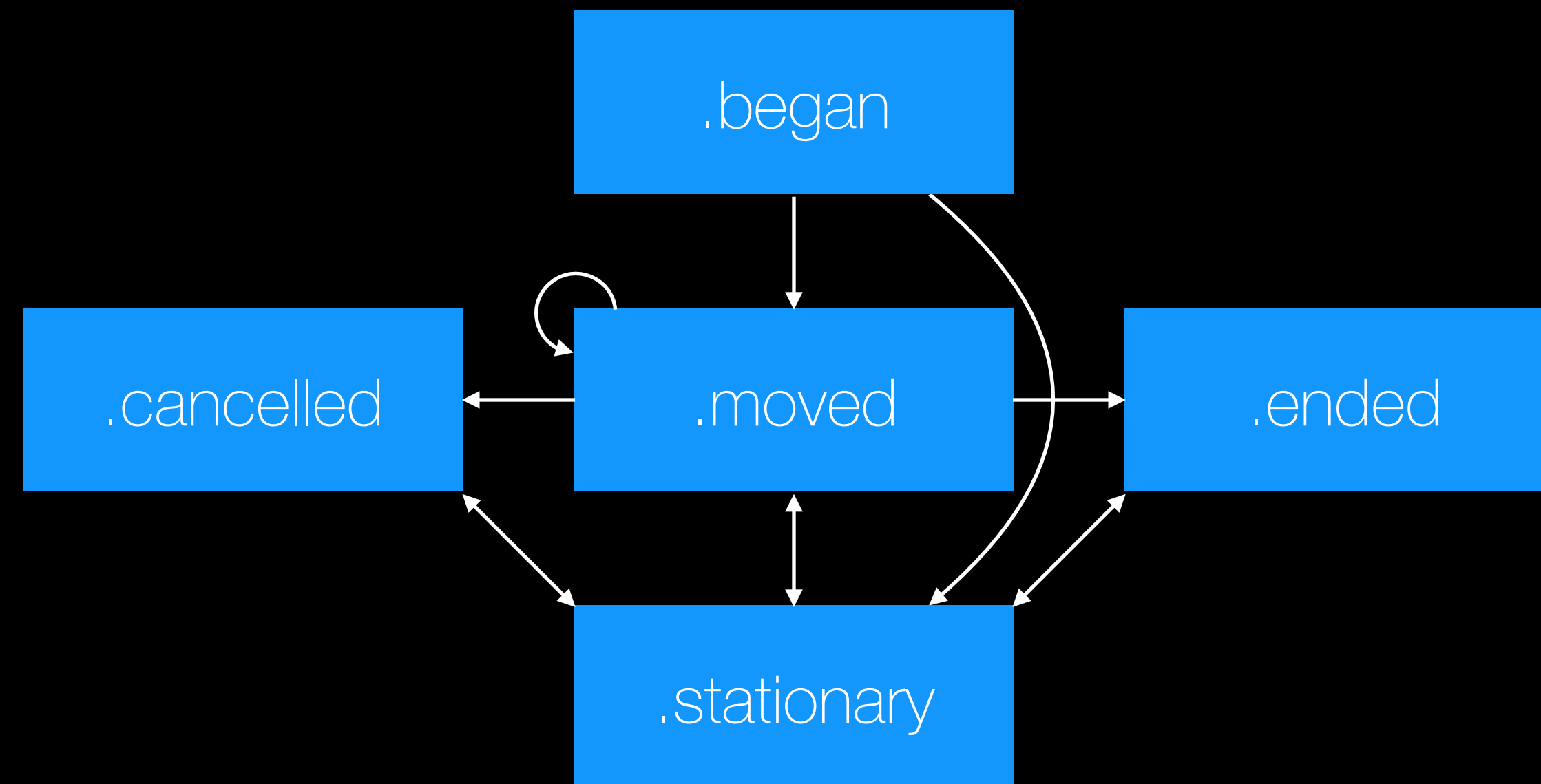
About touches, gestures and space flight



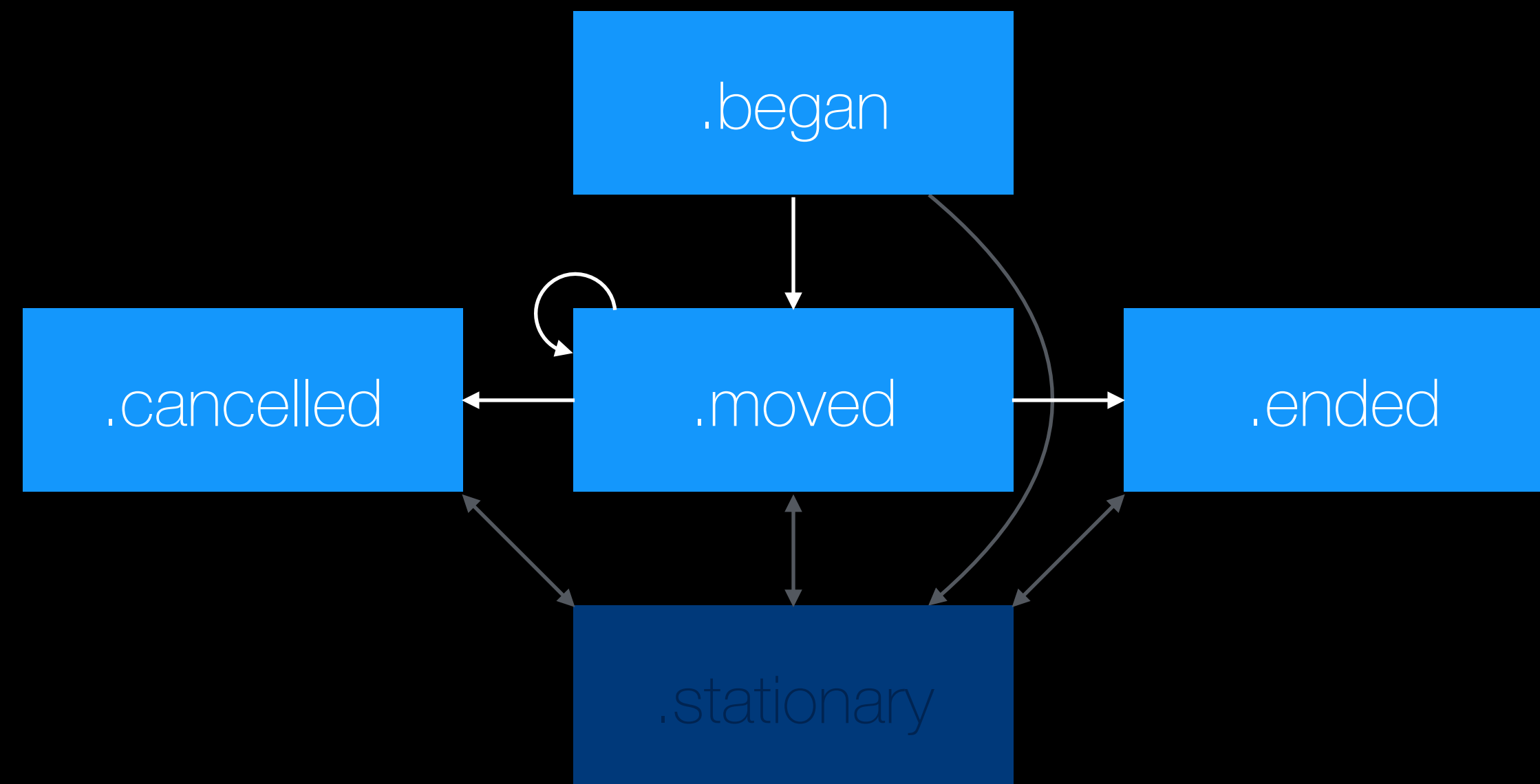
UITouch

- UITouch represents a single touch
- A touch can have multiple taps
- A touch can move around
- A touch has phases: began, moved, stationary, ended, cancelled

Phases of a touch



Phases of a touch



UIEvent

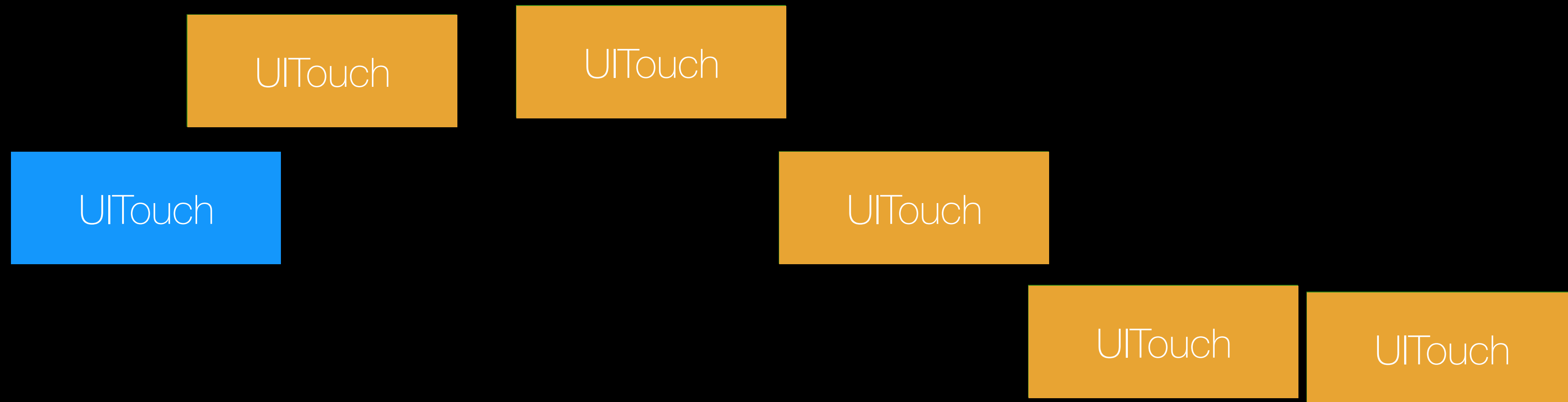
- Describes a user interaction with your app
- Has a type: touches, motion, remote control, presses
- Has a subtype: none, motion shake, remote control play, remote control pause, remote control stop, remote control toggle play pause,...

UIEvent

UIEventTypeTouches

- Describes a touch event
- Can have
 - multiple touches
 - previous touches
 - coalesced touches

UIEvent and UITouch



0x01

coalesced touches



predicted touches



UIEvent and UITouch

- The touch connected to the event remains the same
- There are other touches created for coalesced (i.e. previous) touches
- Algorithms are used to determine future touch locations

UITouch updates

- UIEvent gives you coalesced and predicted touches
- UITouch has estimated properties:
 - `.force`, `.azimuth`, `.altitude`, `.location`
- When an estimate is updated you receive a callback to `touchesEstimatedPropertiesUpdated`

You touch handling

- ... needs to be able to update touches
- ... needs to be able to move touches
- ... can not store UITouch objects



UIGestureRecognizer

- Adds meaning to a sequence of touches (or other input)
- Decouples (touch) input from app logic
- Is a replacement for view based touch handling

UIGestureRecognizer

- Useful predefined gestures:
tap, pinch, rotation, swipe, pan, screen edge, long press
- But also available for subclassing

UIGestureRecognizer

from scratch...

- (void)touchesBegan:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event;
- (void)touchesMoved:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event;
- (void)touchesEnded:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event;
- (void)touchesCancelled:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event;
- (void)touchesEstimatedPropertiesUpdated:(NSSet<UITouch *> *)touches;

UIGestureRecognizer

- (void)touchesBegan:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event;
- (void)touchesMoved:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event;
- (void)touchesEnded:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event;
- (void)touchesCancelled:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event;
- (void)touchesEstimatedPropertiesUpdated:(NSSet<UITouch *> *)touches;

UIGestureRecognizer

- (void)touchesBegan:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event;
- (void)touchesMoved:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event;
- (void)touchesEnded:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event;
- (void)touchesCancelled:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event;
- (void)touchesEstimatedPropertiesUpdated:(NSSet<UITouch *> *)touches;

UIGestureRecognizer

- (void)touchesBegan:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event;
- (void)touchesMoved:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event;
- (void)touchesEnded:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event;
- (void)touchesCancelled:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event;
- (void)touchesEstimatedPropertiesUpdated:(NSSet<UITouch *> *)touches;

UIGestureRecognizer

- (void)touchesBegan:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event;
- (void)touchesMoved:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event;
- (void)touchesEnded:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event;
- (void)touchesCancelled:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event;
- (void)touchesEstimatedPropertiesUpdated:(NSSet<UITouch *> *)touches;

UIGestureRecognizer

- (void)touchesBegan:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event;
- (void)touchesMoved:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event;
- (void)touchesEnded:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event;
- (void)touchesCancelled:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event;
- (void)touchesEstimatedPropertiesUpdated:(NSSet<UITouch *> *)touches;

UIGestureRecognizer

Touch handling

- Store properties you need in your own data structure
 - Remember UITouch is a mutating object!
- Make sure you can update them later one
 - Use **estimationUpdateIndex** as identifier

UIGestureRecognizer

Custom API

- Remember: A gesture recognizer adds meaning to touches
- So should your API
- Don't simply expose an array of touches but what the gesture represents
 - e.g. a tap gesture exposes taps, a swipe gesture swipes

Demo



Take off

- Apple Pencil gives you too additional values
 - altitude
 - azimuth

“Altitude (Alt), sometimes referred to as elevation, is the angle between the object and the observer's local horizon. For visible objects it is an angle between 0 degrees and 90 degrees.”

https://en.wikipedia.org/wiki/Horizontal_coordinate_system

“The azimuth is the angle formed between a reference direction [...] and a line from the observer to a point of interest projected on the same plane as the reference direction orthogonal to the zenith.”

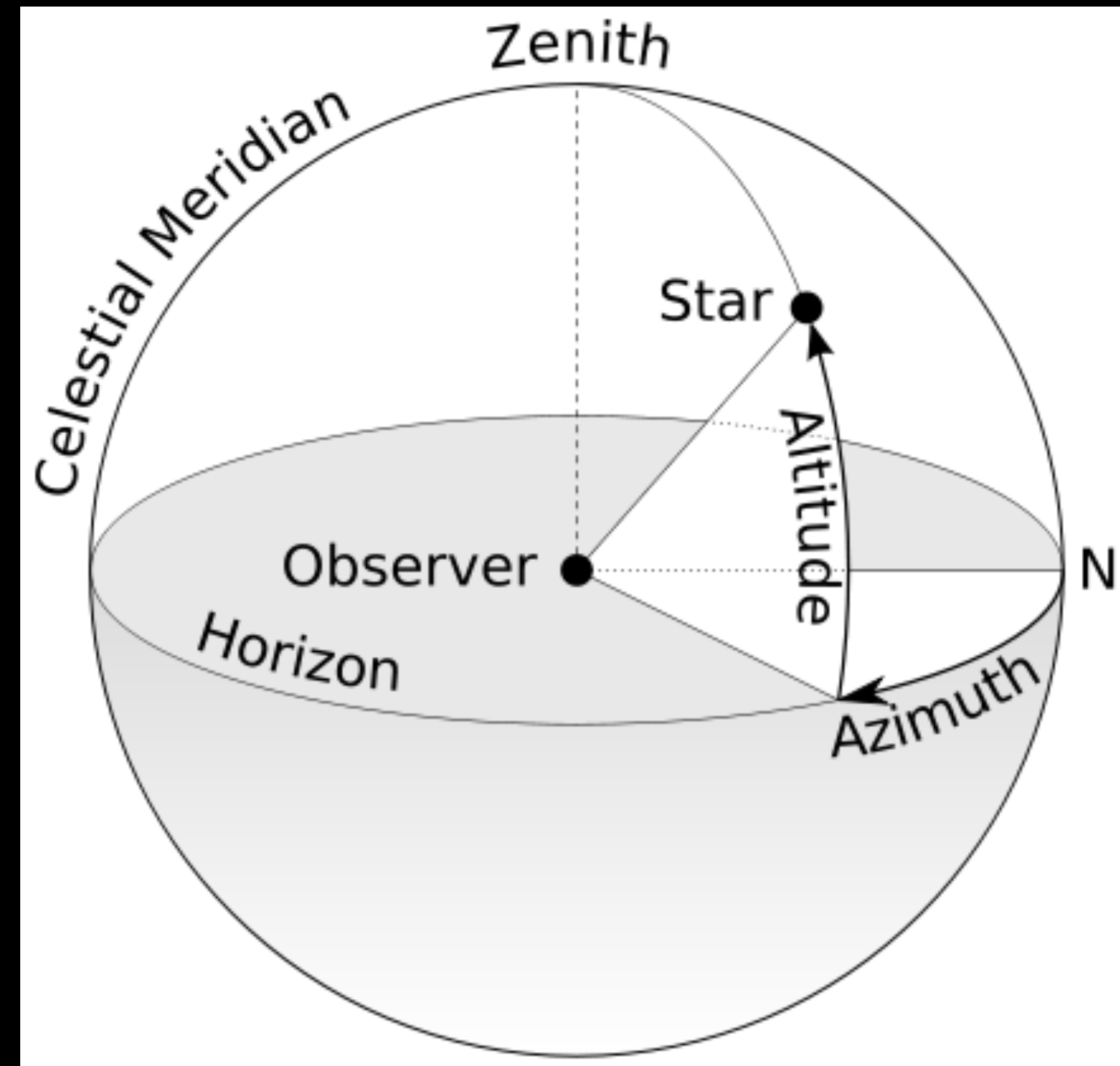
<https://en.wikipedia.org/wiki/Azimuth>

“The zenith is an imaginary point directly ‘above’ a particular location, on the imaginary celestial sphere. ‘Above’ means in the vertical direction opposite to the apparent gravitational force at that location.”

<https://en.wikipedia.org/wiki/Zenith>



Altitude & Azimuth



Altitude & Azimuth

```
// Zero radians indicates that the stylus is parallel to the screen surface,  
// while M_PI/2 radians indicates that it is normal to the screen surface.  
var altitudeAngle: CGFloat { get }  
  
// Zero radians points along the positive X axis.  
func azimuthAngle(in view: UIView?) → CGFloat  
  
func azimuthUnitVector(in view: UIView?) → CGVector // 🚗
```

Demo

Thank you

Michael Ochs

 @_mochs

<https://pspdfkit.com/blog/>