

Digital Solutions

Unit 1

Creating with Code

4. Generating user experiences and interfaces

4.1 Generating low-fidelity prototypes

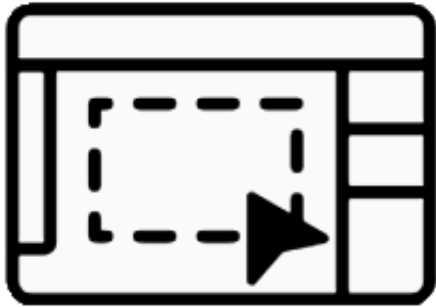
4.2 Generating coded components



LEARNING GOALS

WALT	WILF	TIB
<ul style="list-style-type: none">▪ Generate a low-fidelity prototype of the user interface.▪ Convert pseudocode into swift code components	<ul style="list-style-type: none">▪ Whole class will create a keynote project that uses links to navigate through slides.▪ Students will convert simple pseudocode into correct swift code.	<ul style="list-style-type: none">▪ Technologies have been an integral part of society for as long as humans have had the desire to create solutions to improve their own and others' quality of life.▪ Technologies have an impact on people and societies by transforming, restoring and sustaining the world in which we live.

4.1 Generating low-fidelity prototypes



- Keynote
Using Links



- PowerPoint



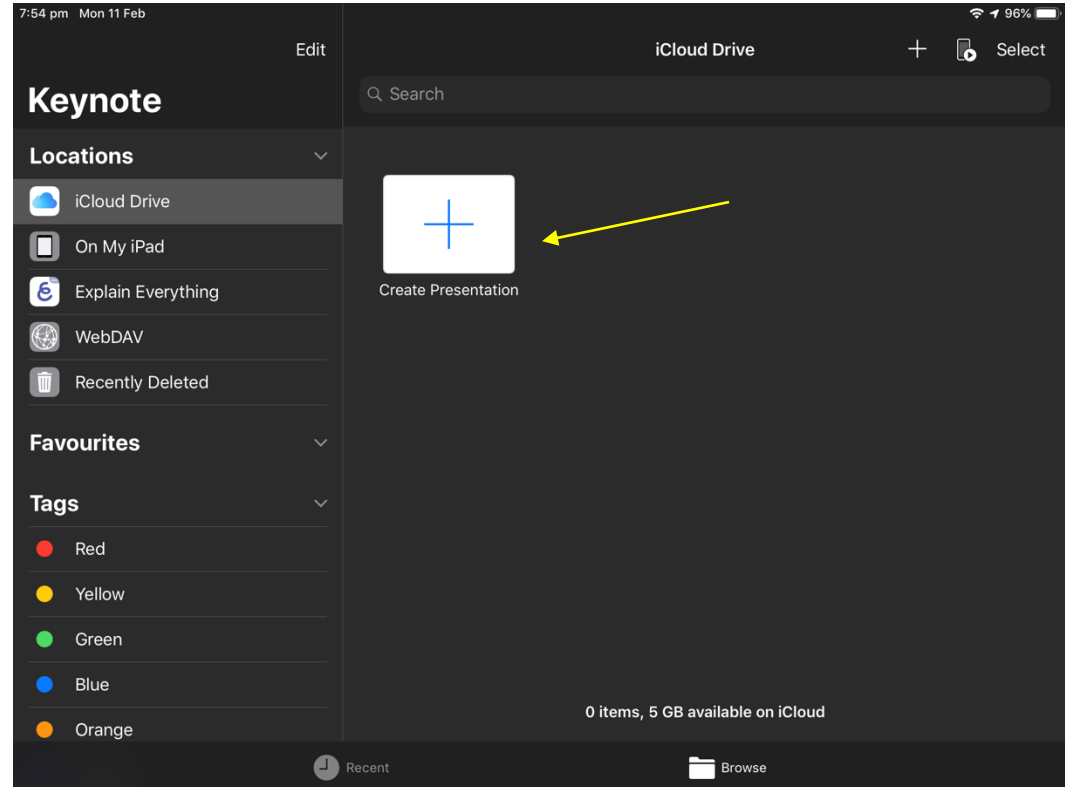
- Screen Dimensions

WALT

- Generate a low-fidelity prototype of the user interface.
- Convert pseudocode into swift code components



Keynote – Using Links



WILF

- Whole class will create a keynote project that uses links to navigate through slides.
- Students will convert simple pseudocode into correct swift code.

WALT

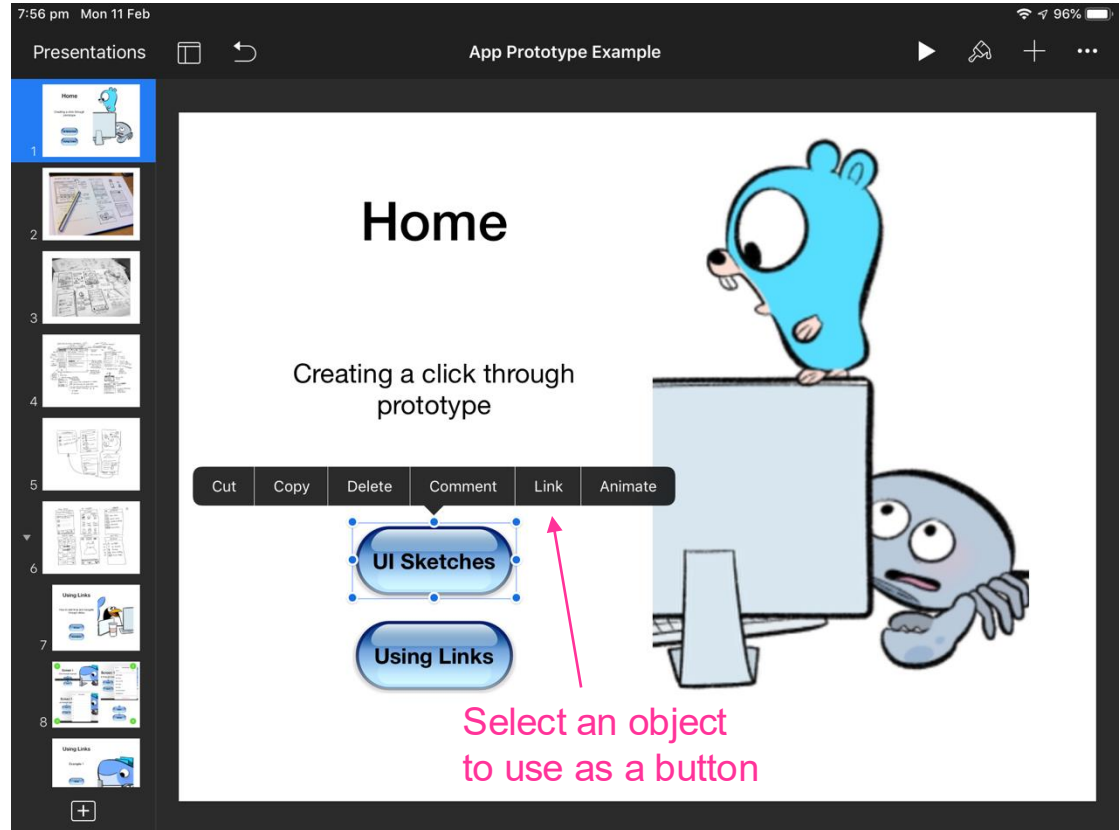
- Generate a low-fidelity prototype of the user interface.
- Convert pseudocode into swift code components

WILF

- Whole class will create a keynote project that uses links to navigate through slides.
- Students will convert simple pseudocode into correct swift code.



Keynote – Using Links

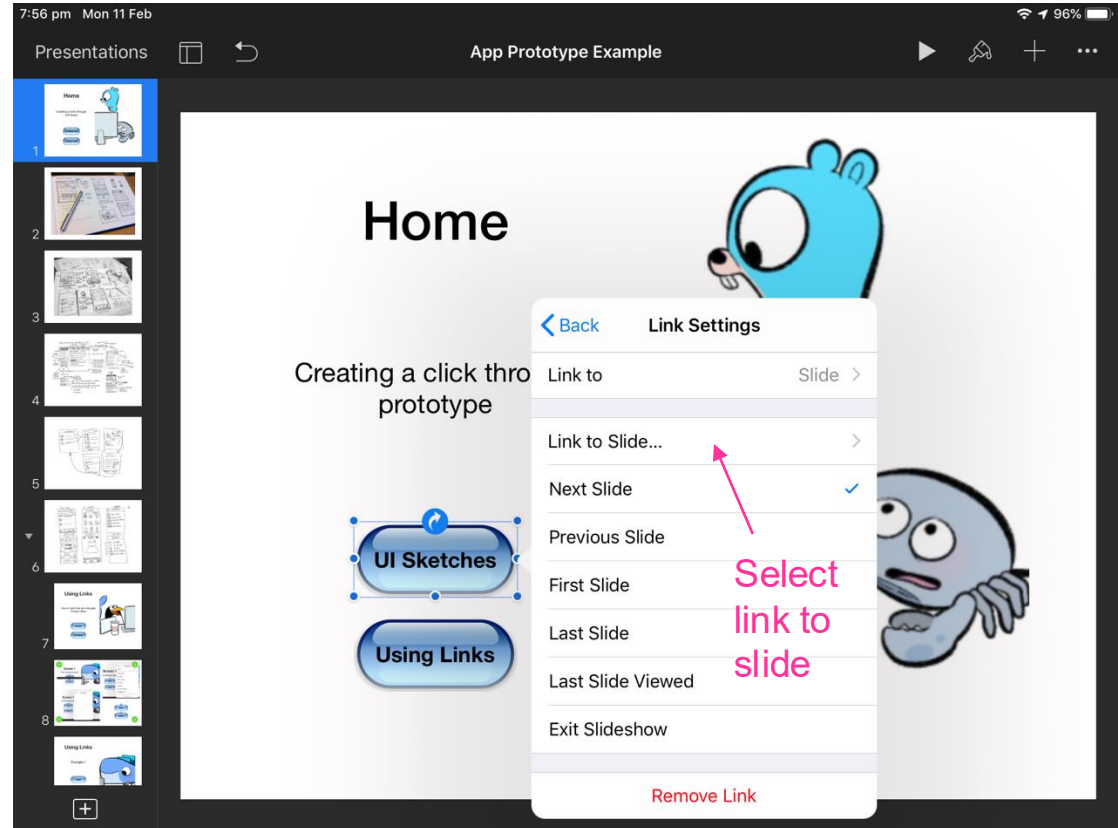


WALT

- Generate a low-fidelity prototype of the user interface.
- Convert pseudocode into swift code components



Keynote – Using Links



WILF

- Whole class will create a keynote project that uses links to navigate through slides.
- Students will convert simple pseudocode into correct swift code.

WALT

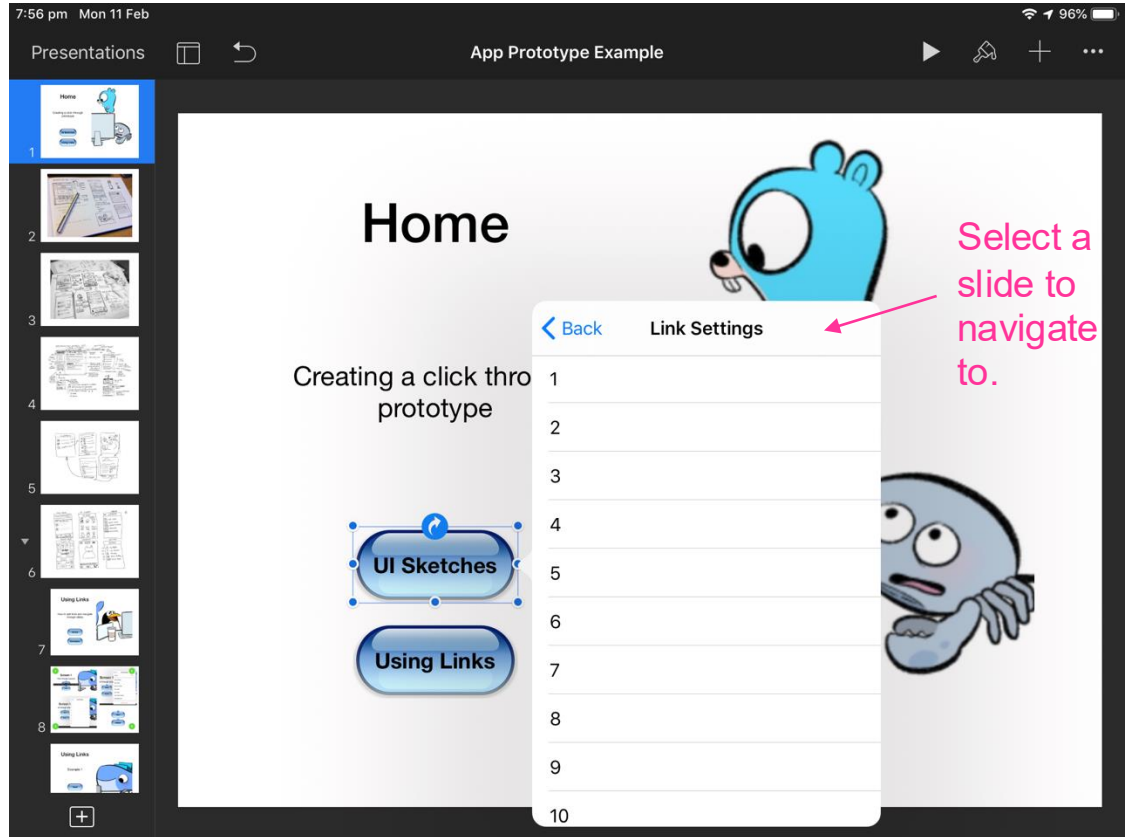
- Generate a low-fidelity prototype of the user interface.
- Convert pseudocode into swift code components



Keynote – Using Links

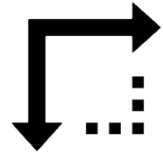
WILF

- Whole class will create a keynote project that uses links to navigate through slides.
- Students will convert simple pseudocode into correct swift code.



WALT

- Generate a low-fidelity prototype of the user interface.
- Convert pseudocode into swift code components



Screen Dimensions

Unless your app is for iPad a low-fidelity prototype generated in keynote will not use the entire screen. Try to use the correct dimensions of the device the app is intended for. Use a simplistic black or white background.

WILF

- Whole class will create a keynote project that uses links to navigate through slides.
- Students will convert simple pseudocode into correct swift code.



WALT

- Generate a low-fidelity prototype of the user interface.
- Convert pseudocode into swift code components



Activity 4.1 A – Café Loco App

Create a low fidelity prototype of the Café Loco app using Keynote or PowerPoint. Aim to include two or three key screens based on your annotated sketches.

A low fidelity prototype is usually not coded, the idea is to *showcase* your concept for the app and to simulate the expected user experience.

WILF

- Whole class will create a keynote project that uses links to navigate through slides.
- Students will convert simple pseudocode into correct swift code.

4.2 Generating coded components



- Converting pseudocode to Swift code

WALT

- Generate a low-fidelity prototype of the user interface.
- Convert pseudocode into swift code components



Converting Pseudocode to Swift

WILF

- Whole class will create a keynote project that uses links to navigate through slides.
- Students will convert simple pseudocode into correct swift code.

WALT

- Generate a low-fidelity prototype of the user interface.
- Convert pseudocode into swift code components



Pseudocode -> Swift

Example 1

Pseudocode

```
IF grade >= 50 THEN
    OUTPUT pass
ELSE
    OUTPUT fail
END IF
```

Swift Code

```
if grade >= 50 {
    print("pass")
} else {
    print("fail")
}
```

WILF

- Whole class will create a keynote project that uses links to navigate through slides.
- Students will convert simple pseudocode into correct swift code.

WALT

- Generate a low-fidelity prototype of the user interface.
- Convert pseudocode into swift code components



Pseudocode -> Swift

Example 2

Pseudocode

```
WHILE userNum not equal 0
    OUTPUT userNum
    userNum = userNum - 1
END WHILE
```

Swift Code

```
while userNum != 0 {
    print(userNum)
    userNum = userNum - 1
}
```

WILF

- Whole class will create a keynote project that uses links to navigate through slides.
- Students will convert simple pseudocode into correct swift code.

WALT

- Generate a low-fidelity prototype of the user interface.
- Convert pseudocode into swift code components



Pseudocode -> Swift

Example 3

Pseudocode

area = width x length

perimeter = 2 x (width + length)

REPEAT X 5

OUTPUT area

OUTPUT perimeter

END REPEAT

Swift Code

area = width * length

perimeter = 2 * (width + length)

for i in 1...5 {

print(area)

print(perimeter)

}

WILF

- Whole class will create a keynote project that uses links to navigate through slides.
- Students will convert simple pseudocode into correct swift code.

WALT

- Generate a low-fidelity prototype of the user interface.
- Convert pseudocode into swift code components



Activity 4.2 A – Sample Swift Code

In some assignments you will be asked to provide a small sample of Swift code for an app concept and at other times you will provide a lot more. As a low-fidelity prototype is not generally coded (FIA1, IA1) you will need a tool to allow you to generate a few algorithms. The Playgrounds app on your iPad will allow you to do this.

WILF

- Whole class will create a keynote project that uses links to navigate through slides.
- Students will convert simple pseudocode into correct swift code.
 - Choose one of the algorithms from the examples on the previous slides
 - Generate the Swift code shown in playground
 - You will need to declare variables and initialise them with a value to make sure the code runs
 - Add a few annotations to your code that explain key parts