

# Application GUI Design

Notes From a Tizen Toolkit Developer



**[stosb.com/talks](https://stosb.com/talks)**

Tom Hacohen  
Samsung Electronics Open Source Group  
[tom@osg.samsung.com](mailto:tom@osg.samsung.com)  
[@TomHacohen](https://twitter.com/TomHacohen)

# What does it do?

---

- ▶ Essential features

# What does it do?

---

- ▶ Essential features
- ▶ Nice to have features

# What does it do?

---

- ▶ Essential features
- ▶ Nice to have features
- ▶ Niche features ( $<1\%$  of the users)



# What does it do?

---

- ▶ Essential features
- ▶ Nice to have features
- ▶ Niche features ( $<1\%$  of the users)
- ▶ Remove all the non-essential features

# Who is it for?

---

- ▶ CLI power users? Designers?

# Who is it for?

---

- ▶ CLI power users? Designers?
- ▶ Target environment

# Who is it for?

---

- ▶ CLI power users? Designers?
- ▶ Target environment
- ▶ Common demographics

# Who is it for?

---

- ▶ CLI power users? Designers?
- ▶ Target environment
- ▶ Common demographics
- ▶ Application specific classifications

# Who is it for?

---

- ▶ CLI power users? Designers?
- ▶ Target environment
- ▶ Common demographics
- ▶ Application specific classifications
- ▶ Userbase  $\neq$  you

- ▶ Adapt feature list according to your audience



- ▶ Adapt feature list according to your audience
- ▶ Keep focus on the more important features





- ▶ Adapt feature list according to your audience
- ▶ Keep focus on the more important features
- ▶ Avoid creating complex UIs



- ▶ Adapt feature list according to your audience
- ▶ Keep focus on the more important features
- ▶ Avoid creating complex UIs
- ▶ Keep option lists (combo box) short, simple and in a sensible order



## Have a rough sketch

---

- ▶ Make a general storyboard sketch

## Have a rough sketch

---

- ▶ Make a general storyboard sketch
- ▶ Have all the major interactions and features there

## Have a rough sketch

---

- ▶ Make a general storyboard sketch
- ▶ Have all the major interactions and features there
- ▶ Can be really rough – it's more about the purpose of the “pages”

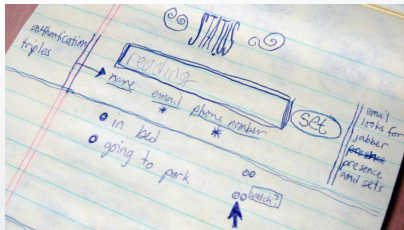
## Have a rough sketch

---

- ▶ Make a general storyboard sketch
- ▶ Have all the major interactions and features there
- ▶ Can be really rough – it's more about the purpose of the “pages”

# Have a rough sketch

- ▶ Make a general storyboard sketch
- ▶ Have all the major interactions and features there
- ▶ Can be really rough – it's more about the purpose of the “pages”



# Stick to the basics

---

- ▶ Don't bother with colour – harder and will be added later



## Stick to the basics

---

- ▶ Don't bother with colour – harder and will be added later
- ▶ Do not customise available widgets/patterns unless there really is no other way

## Stick to the basics

---

- ▶ Don't bother with colour – harder and will be added later
- ▶ Do not customise available widgets/patterns unless there really is no other way
- ▶ Develop it around the content – content is king

# I meant it! KISS...

---

- ▶ Be purposefully simple



# I meant it! KISS...

---

- ▶ Be purposefully simple
- ▶ Skeuomorphs are almost always bad



## I meant it! KISS...

---

- ▶ Be purposefully simple
- ▶ Skeuomorphs are almost always bad
- ▶ Be consistent (easier when simple)



Next



Next

## I meant it! KISS...

---

- ▶ Be purposefully simple
- ▶ Skeuomorphs are almost always bad
- ▶ Be consistent (easier when simple)
- ▶ Make it simple for your users – don't create new usage patterns, and adhere to common ones



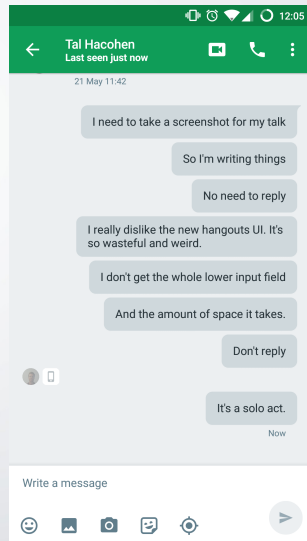
# I meant it! KISS...

- ▶ Be purposefully simple
- ▶ Skeuomorphs are almost always bad
- ▶ Be consistent (easier when simple)
- ▶ Make it simple for your users – don't create new usage patterns, and adhere to common ones
- ▶ Common (all?) usage patterns should be easily available



# I meant it! KISS...

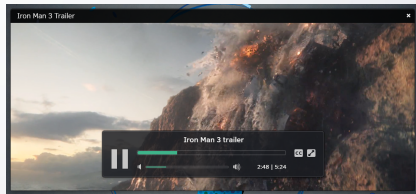
- ▶ Be purposefully simple
- ▶ Skeuomorphs are almost always bad
- ▶ Be consistent (easier when simple)
- ▶ Make it simple for your users – don't create new usage patterns, and adhere to common ones
- ▶ Common (all?) usage patterns should be easily available
  - ▶ Rest should be revealed as needed



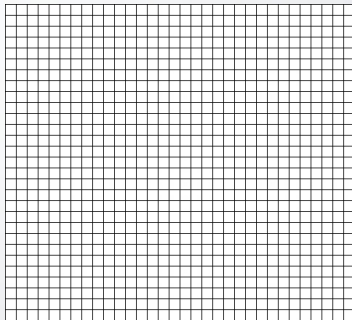


# I meant it! KISS...

- ▶ Be purposefully simple
- ▶ Skeuomorphs are almost always bad
- ▶ Be consistent (easier when simple)
- ▶ Make it simple for your users – don't create new usage patterns, and adhere to common ones
- ▶ Common (all?) usage patterns should be easily available
  - ▶ Rest should be revealed as needed
- ▶ Make self-documenting applications

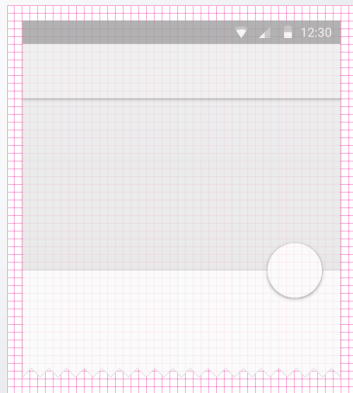


- ▶ Choose a baseline unit size (e.g. 8px)



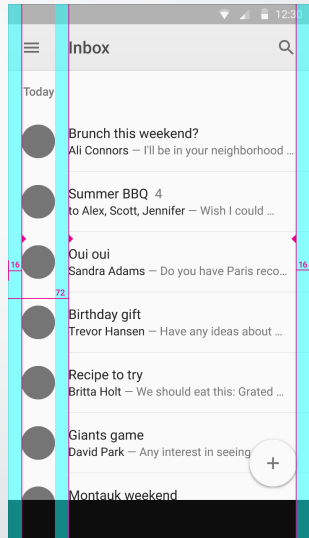
# Spacing

- ▶ Choose a baseline unit size (e.g. 8px)
- ▶ Align everything to that baseline size



# Spacing

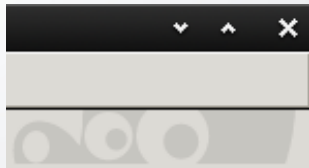
- ▶ Choose a baseline unit size (e.g. 8px)
- ▶ Align everything to that baseline size
- ▶ Use a small set of spacing alternatives (e.g. 2, 3 and 6 units)



# Spacing

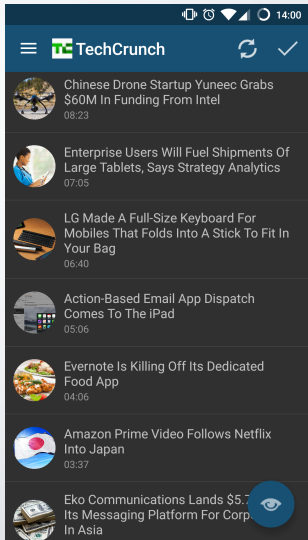
---

- ▶ Choose a baseline unit size (e.g. 8px)
- ▶ Align everything to that baseline size
- ▶ Use a small set of spacing alternatives (e.g. 2, 3 and 6 units)
- ▶ Make sure touch-targets are spaced enough



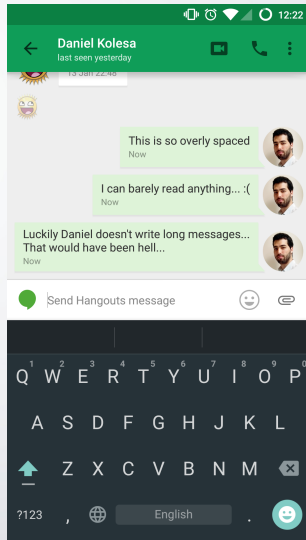
# Spacing

- ▶ Choose a baseline unit size (e.g. 8px)
- ▶ Align everything to that baseline size
- ▶ Use a small set of spacing alternatives (e.g. 2, 3 and 6 units)
- ▶ Make sure touch-targets are spaced enough
  - ▶ Counter example



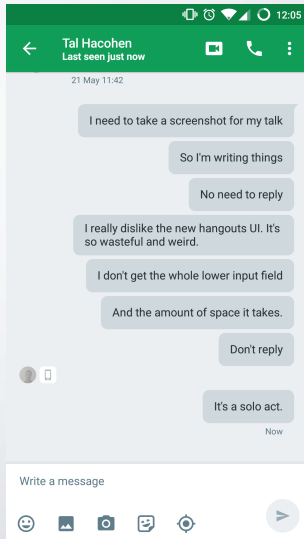
# Spacing

- ▶ Choose a baseline unit size (e.g. 8px)
- ▶ Align everything to that baseline size
- ▶ Use a small set of spacing alternatives (e.g. 2, 3 and 6 units)
- ▶ Make sure touch-targets are spaced enough
  - ▶ Counter example
- ▶ Be generous, but don't overdo it (don't waste my screen estate)



# Spacing

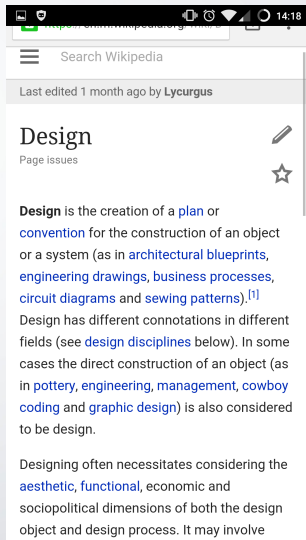
- ▶ Choose a baseline unit size (e.g. 8px)
- ▶ Align everything to that baseline size
- ▶ Use a small set of spacing alternatives (e.g. 2, 3 and 6 units)
- ▶ Make sure touch-targets are spaced enough
  - ▶ Counter example
- ▶ Be generous, but don't overdo it (don't waste my screen estate)
  - ▶ Counter example





# Spacing

- ▶ Choose a baseline unit size (e.g. 8px)
- ▶ Align everything to that baseline size
- ▶ Use a small set of spacing alternatives (e.g. 2, 3 and 6 units)
- ▶ Make sure touch-targets are spaced enough
  - ▶ Counter example
- ▶ Be generous, but don't overdo it (don't waste my screen estate)
  - ▶ Counter example
- ▶ Give back space when possible



# Organization

- ▶ Content should be at the front of the stage



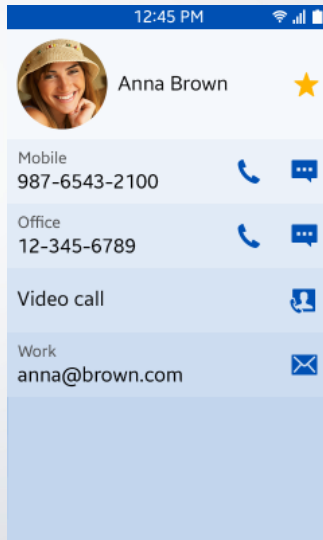
# Organization

- ▶ Content should be at the front of the stage
- ▶ Important functionality in key positions



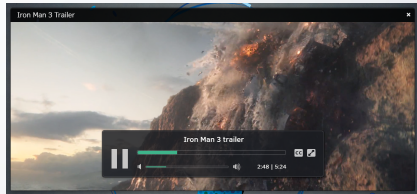
# Organization

- ▶ Content should be at the front of the stage
- ▶ Important functionality in key positions
- ▶ Make the hierarchy of information clear



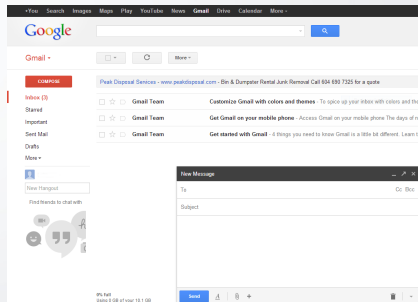
# Organization

- ▶ Content should be at the front of the stage
- ▶ Important functionality in key positions
- ▶ Make the hierarchy of information clear
- ▶ Associate related elements



# Organization

- ▶ Content should be at the front of the stage
- ▶ Important functionality in key positions
- ▶ Make the hierarchy of information clear
- ▶ Associate related elements
- ▶ Help directing the user's focus



- ▶ Use known icons on buttons



# Icons and images

---

- ▶ Use known icons on buttons
- ▶ Don't use a known icon for something other than intended





# Icons and images

- ▶ Use known icons on buttons
- ▶ Don't use a known icon for something other than intended
- ▶ Use rich graphics when appropriate (cover-art, mood graphics, etc.)



# Icons and images

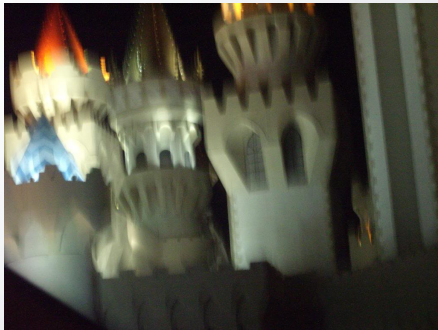
- ▶ Use known icons on buttons
- ▶ Don't use a known icon for something other than intended
- ▶ Use rich graphics when appropriate (cover-art, mood graphics, etc.)
- ▶ Don't overshadow content



# Icons and images

---

- ▶ Use known icons on buttons
- ▶ Don't use a known icon for something other than intended
- ▶ Use rich graphics when appropriate (cover-art, mood graphics, etc.)
- ▶ Don't overshadow content
- ▶ Don't use ugly graphics



# Icons and images

---

- ▶ Use known icons on buttons
- ▶ Don't use a known icon for something other than intended
- ▶ Use rich graphics when appropriate (cover-art, mood graphics, etc.)
- ▶ Don't overshadow content
- ▶ Don't use ugly graphics
- ▶ Have consistent sizing



- ▶ Use a readable font size

Text should be big and readable for everyone

- ▶ Use a readable font size
- ▶ Space up your text

Alice's Adventures in Wonderland (commonly shortened to Alice in Wonderland) is an 1865 novel written by English author Charles Lutwidge Dodgson under the pseudonym Lewis Carroll. It tells of a girl named Alice falling through a rabbit hole into a fantasy world populated by peculiar, anthropomorphic creatures. The tale plays with logic, giving the story lasting popularity with adults as well as with children.

Alice's Adventures in Wonderland (commonly shortened to Alice in Wonderland) is an 1865 novel written by English author Charles Lutwidge Dodgson under the pseudonym Lewis Carroll. It tells of a girl named Alice falling through a rabbit hole into a fantasy world populated by peculiar, anthropomorphic creatures. The tale plays with logic, giving the story lasting popularity with adults as well as with children.

- ▶ Use a readable font size
- ▶ Space up your text
- ▶ Use text attributes (e.g. bold and size) to make text more or less prominent

**Alice's** Adventures in **Wonderland**  
(commonly shortened to **Alice** in  
Wonderland) is an 1865 novel written by  
English author *Charles Lutwidge*  
*Dodgson* under the pseudonym Lewis  
Carroll. It tells of a girl named **Alice**  
falling through a rabbit hole into a  
fantasy world populated by peculiar,  
anthropomorphic creatures. The tale  
plays with logic, giving the story lasting  
popularity with adults as well as with  
children.

- ▶ Use a readable font size
- ▶ Space up your text
- ▶ Use text attributes (e.g. bold and size) to make text more or less prominent
- ▶ Use a small set of fonts (probably one)





# Overlaying text on images

---

- ▶ Just put it on (bad)



# Overlaying text on images

---

- ▶ Just put it on (bad)
- ▶ Black/colour/white-wash the whole image to make text readable



# Overlaying text on images

---

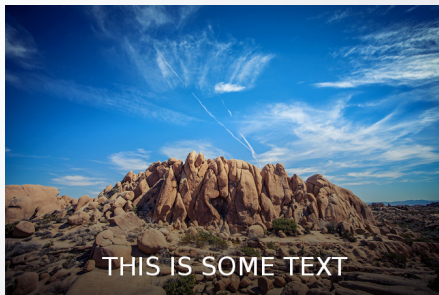
- ▶ Just put it on (bad)
- ▶ Black/colour/white-wash the whole image to make text readable
- ▶ Add translucent background to the text



# Overlaying text on images

---

- ▶ Just put it on (bad)
- ▶ Black/colour/white-wash the whole image to make text readable
- ▶ Add translucent background to the text
- ▶ Add a translucent gradient (i.e. partial black/white-wash) so your text area is handled



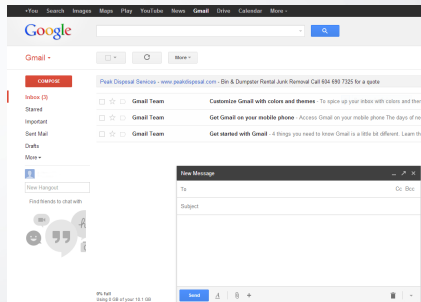
# Overlaying text on images

---

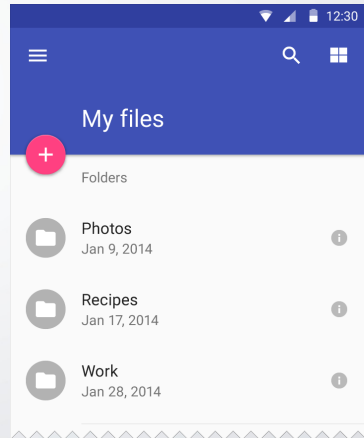
- ▶ Just put it on (bad)
- ▶ Black/colour/white-wash the whole image to make text readable
- ▶ Add translucent background to the text
- ▶ Add a translucent gradient (i.e. partial black/white-wash) so your text area is handled
- ▶ Use a big font with a shadow and on outline



- Option 1 (easier): keep the interface b&w and use colour to direct focus



- ▶ Option 1 (easier): keep the interface b&w and use colour to direct focus
- ▶ Option 2: Choose a palette of 2 different hues and use different shades



- ▶ Option 1 (easier): keep the interface b&w and use colour to direct focus
- ▶ Option 2: Choose a palette of 2 different hues and use different shades
- ▶ I almost always tint my greys (and not use black)

**Some text!**  
**Some text!**

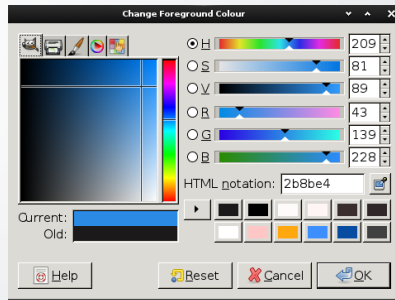


- ▶ Option 1 (easier): keep the interface b&w and use colour to direct focus
- ▶ Option 2: Choose a palette of 2 different hues and use different shades
- ▶ I almost always tint my greys (and not use black)
- ▶ Find a good palette online

Primary – Indigo	
500	#3F51B5
100	#C5CAE9
500	#3F51B5
700	#303F9F
Accent – Pink	
A200	#FF4081
Fallback	
A100	#FF80AB
A400	#F50057

# Colour

- ▶ Option 1 (easier): keep the interface b&w and use colour to direct focus
- ▶ Option 2: Choose a palette of 2 different hues and use different shades
- ▶ I almost always tint my greys (and not use black)
- ▶ Find a good palette online
- ▶ Use HSL/HSV rather RGB when choosing colours



# Colour

- ▶ Option 1 (easier): keep the interface b&w and use colour to direct focus
- ▶ Option 2: Choose a palette of 2 different hues and use different shades
- ▶ I almost always tint my greys (and not use black)
- ▶ Find a good palette online
- ▶ Use HSL/HSV rather RGB when choosing colours
- ▶ Be aware of cultural differences



# Mimic what works

---

- ▶ Compare to other applications you/users like better using what we've covered

# Mimic what works

---

- ▶ Compare to other applications you/users like better using what we've covered
- ▶ Mimic what's good there (e.g. do you need more spacing?)

# Mimic what works

---

- ▶ Compare to other applications you/users like better using what we've covered
- ▶ Mimic what's good there (e.g. do you need more spacing?)
- ▶ It's usually easier to mimic than to design from scratch

## Mimic what works

---

- ▶ Compare to other applications you/users like better using what we've covered
- ▶ Mimic what's good there (e.g. do you need more spacing?)
- ▶ It's usually easier to mimic than to design from scratch
- ▶ Don't mimic the bad things (i.e. use them as excuse)

# Mimic what works

---

- ▶ Compare to other applications you/users like better using what we've covered
- ▶ Mimic what's good there (e.g. do you need more spacing?)
- ▶ It's usually easier to mimic than to design from scratch
- ▶ Don't mimic the bad things (i.e. use them as excuse)
- ▶ Don't copy, learn. . .



- ▶ Consistent behaviour (with the platform and within the application)

- ▶ Consistent behaviour (with the platform and within the application)
- ▶ Start instantly and lazy load in the background

- ▶ Consistent behaviour (with the platform and within the application)
- ▶ Start instantly and lazy load in the background
- ▶ Everything should take a small amount of clicks

- ▶ Consistent behaviour (with the platform and within the application)
- ▶ Start instantly and lazy load in the background
- ▶ Everything should take a small amount of clicks
- ▶ Discoverable UI (easy to figure out how to do things)

# More user experience tips

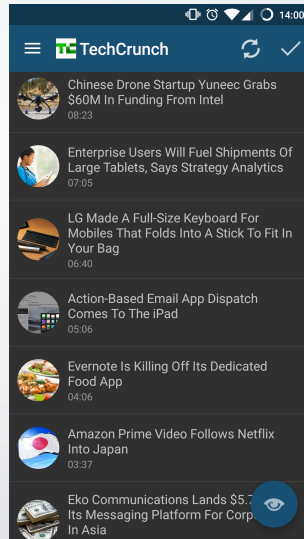
---

- ▶ Make it hard to make mistakes



# More user experience tips

- ▶ Make it hard to make mistakes
  - ▶ Counter example



# More user experience tips

- ▶ Make it hard to make mistakes
  - ▶ Counter example
- ▶ Avoid interruptions



## More user experience tips

---

- ▶ Make it hard to make mistakes
  - ▶ Counter example
- ▶ Avoid interruptions
- ▶ Sort long lists in a predictable, sensible order



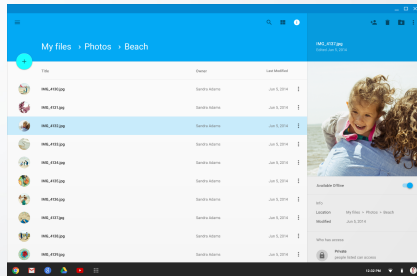
## More user experience tips

---

- ▶ Make it hard to make mistakes
  - ▶ Counter example
- ▶ Avoid interruptions
- ▶ Sort long lists in a predictable, sensible order
- ▶ If your users do something and expect something to happen, it probably should happen

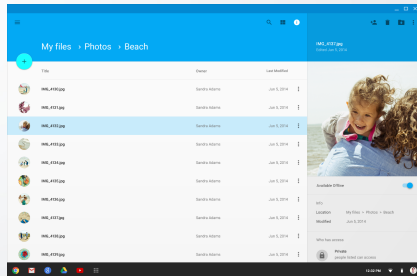
# Material design (Google)

- Highly talked about cross device design guidelines



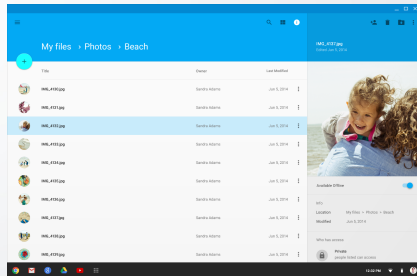
# Material design (Google)

- ▶ Highly talked about cross device design guidelines
- ▶ A fairly good Android adoption rate



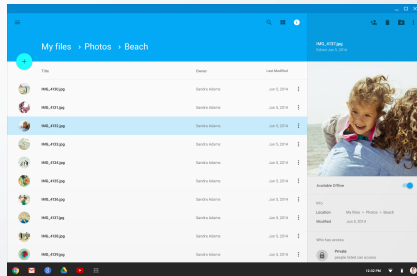
# Material design (Google)

- ▶ Highly talked about cross device design guidelines
- ▶ A fairly good Android adoption rate
- ▶ I personally don't like animations that slow users down



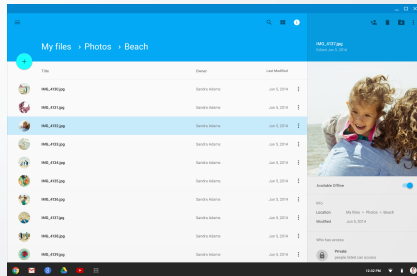
# Material design (Google)

- ▶ Highly talked about cross device design guidelines
- ▶ A fairly good Android adoption rate
- ▶ I personally don't like animations that slow users down
- ▶ Concepts are good, it's implementations that are not always good



# Material design (Google)

- ▶ Highly talked about cross device design guidelines
- ▶ A fairly good Android adoption rate
- ▶ I personally don't like animations that slow users down
- ▶ Concepts are good, it's implementations that are not always good
- ▶ It doesn't feel like they care about low-end and power consumption



# iOS (Apple)

- Old style: horrible skeuomorphism



# iOS (Apple)

- ▶ Old style: horrible skeuomorphism
- ▶ New style: very simple and clean, sometimes confusing





# iOS (Apple)

- ▶ Old style: horrible skeuomorphism
- ▶ New style: very simple and clean, sometimes confusing
- ▶ Flat and easy to theme



# iOS (Apple)

- ▶ Old style: horrible skeuomorphism
- ▶ New style: very simple and clean, sometimes confusing
- ▶ Flat and easy to theme
- ▶ Usually not very discoverable

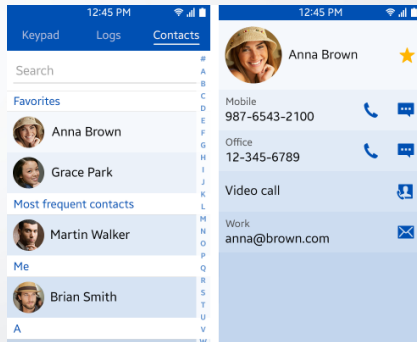


# iOS (Apple)

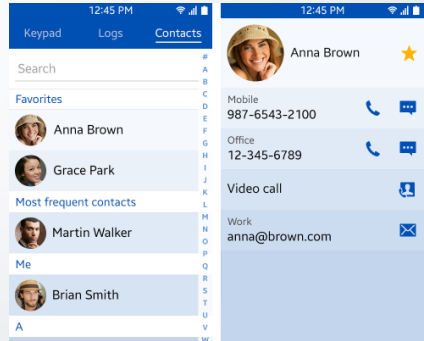
- ▶ Old style: horrible skeuomorphism
- ▶ New style: very simple and clean, sometimes confusing
- ▶ Flat and easy to theme
- ▶ Usually not very discoverable
- ▶ Good general guidelines



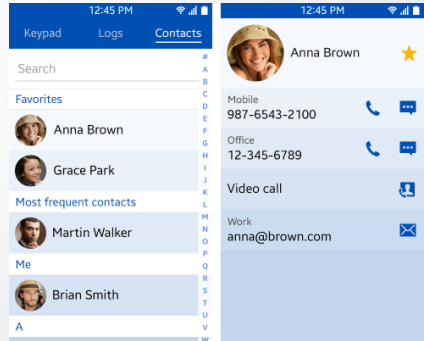
- Designed for low resolution devices



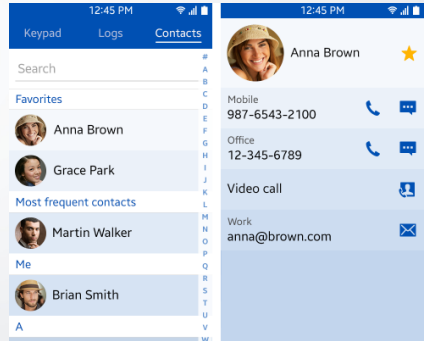
- ▶ Designed for low resolution devices
- ▶ Focus on content and user needs



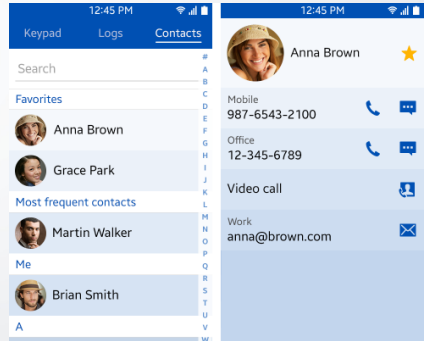
- ▶ Designed for low resolution devices
- ▶ Focus on content and user needs
- ▶ Frugal with screen estate



- ▶ Designed for low resolution devices
- ▶ Focus on content and user needs
- ▶ Frugal with screen estate
- ▶ Tries to make applications consistent (encourages a fixed colour theme)

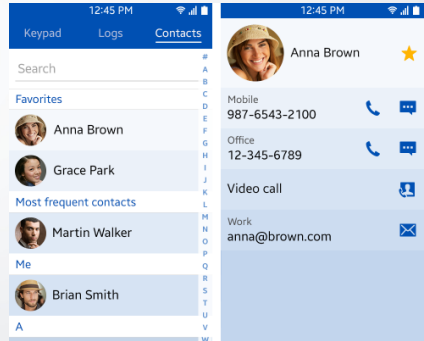


- ▶ Designed for low resolution devices
- ▶ Focus on content and user needs
- ▶ Frugal with screen estate
- ▶ Tries to make applications consistent (encourages a fixed colour theme)
- ▶ Make interaction simple and easy

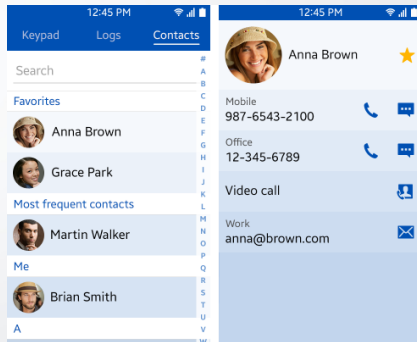




- ▶ Designed for low resolution devices
- ▶ Focus on content and user needs
- ▶ Frugal with screen estate
- ▶ Tries to make applications consistent (encourages a fixed colour theme)
- ▶ Make interaction simple and easy
- ▶ Aims to support user customisation



- ▶ Designed for low resolution devices
- ▶ Focus on content and user needs
- ▶ Frugal with screen estate
- ▶ Tries to make applications consistent (encourages a fixed colour theme)
- ▶ Make interaction simple and easy
- ▶ Aims to support user customisation
- ▶ Flat and “fun” design



- ▶ Trying to pixel match a design across toolkits is a bad idea

# Know your toolkit

---

- ▶ Trying to pixel match a design across toolkits is a bad idea
- ▶ Know what takes a performance hit on your toolkit

- ▶ Trying to pixel match a design across toolkits is a bad idea
- ▶ Know what takes a performance hit on your toolkit
- ▶ Check out the toolkit's common patterns, those are usually best for performance and users

## Easy takeaways

---

- ▶ It's like API design, it should be simple, self-documenting, consistent and have a familiar feeling

## Easy takeaways

---

- ▶ It's like API design, it should be simple, self-documenting, consistent and have a familiar feeling
- ▶ Start with black and white, don't colour unless you need it

## Easy takeaways

---

- ▶ It's like API design, it should be simple, self-documenting, consistent and have a familiar feeling
- ▶ Start with black and white, don't colour unless you need it
- ▶ Saturate your greys and almost never use black



## Easy takeaways

---

- ▶ It's like API design, it should be simple, self-documenting, consistent and have a familiar feeling
- ▶ Start with black and white, don't colour unless you need it
- ▶ Saturate your greys and almost never use black
- ▶ Add spacing and make your text bigger

## Easy takeaways

---

- ▶ It's like API design, it should be simple, self-documenting, consistent and have a familiar feeling
- ▶ Start with black and white, don't colour unless you need it
- ▶ Saturate your greys and almost never use black
- ▶ Add spacing and make your text bigger
- ▶ Use common patterns (and take inspiration from the best)

## Easy takeaways

---

- ▶ It's like API design, it should be simple, self-documenting, consistent and have a familiar feeling
- ▶ Start with black and white, don't colour unless you need it
- ▶ Saturate your greys and almost never use black
- ▶ Add spacing and make your text bigger
- ▶ Use common patterns (and take inspiration from the best)
- ▶ Pay the design some thought (from the start)

## Easy takeaways

---

- ▶ It's like API design, it should be simple, self-documenting, consistent and have a familiar feeling
- ▶ Start with black and white, don't colour unless you need it
- ▶ Saturate your greys and almost never use black
- ▶ Add spacing and make your text bigger
- ▶ Use common patterns (and take inspiration from the best)
- ▶ Pay the design some thought (from the start)
- ▶ KISS

---

Questions?

Tom Hacoen

[tom@osg.samsung.com](mailto:tom@osg.samsung.com)

<http://stosb.com>

@TomHacoen

- 
- ▶ Page 11, flight-deck.jpg
  - ▶ Page 15, twitter-sketch.jpg
  - ▶ Page 23, purposefully-simple.png
  - ▶ Page ??, skeuomorph.jpg
  - ▶ Page ??, self-documenting.png
  - ▶ Page ??, layout-baseline-align.png
  - ▶ Page ??, layout-spacing-alternatives.png
  - ▶ Page 58, direct-attention.png

- 
- ▶ Page ??, imagery-mood.png
  - ▶ Page ??, bad-blurry.png
  - ▶ Page ??, multiple-fonts.png
  - ▶ Page 53, text-overlay-bad.jpg
  - ▶ Page ??, two-colours.png
  - ▶ Page ??, palette.png
  - ▶ Page ??, colour-culture.jpg
  - ▶ Page ??, colour-culture.jpg
  - ▶ Page 73, repo-delete-confirmation.png
  - ▶ Page ??, interrupt-programmer.jpg
  - ▶ Page 78, material.png
  - ▶ Page 83, ios6v7.jpg
  - ▶ Page 88, tizen.png
  - ▶ Page ??, info-hierarchy.png