Robust Relational Layout Synthesis from Examples



Pavol Bielik, Marc Fischer, Martin Vechev

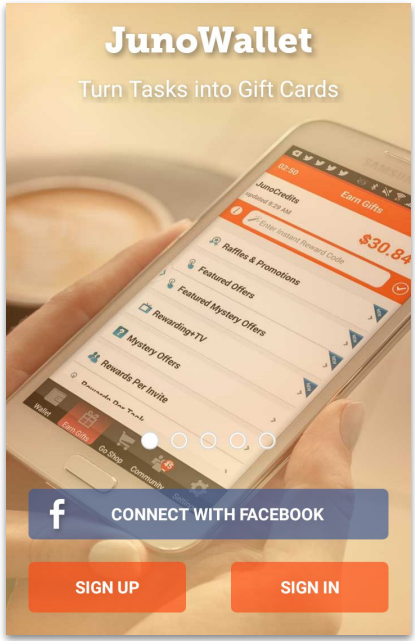
Department of Computer Science
ETH Zurich

User Interface Design Today

Developer →

← Designer

picture created by freepik
[Credits: Radu Cristian]



Figma

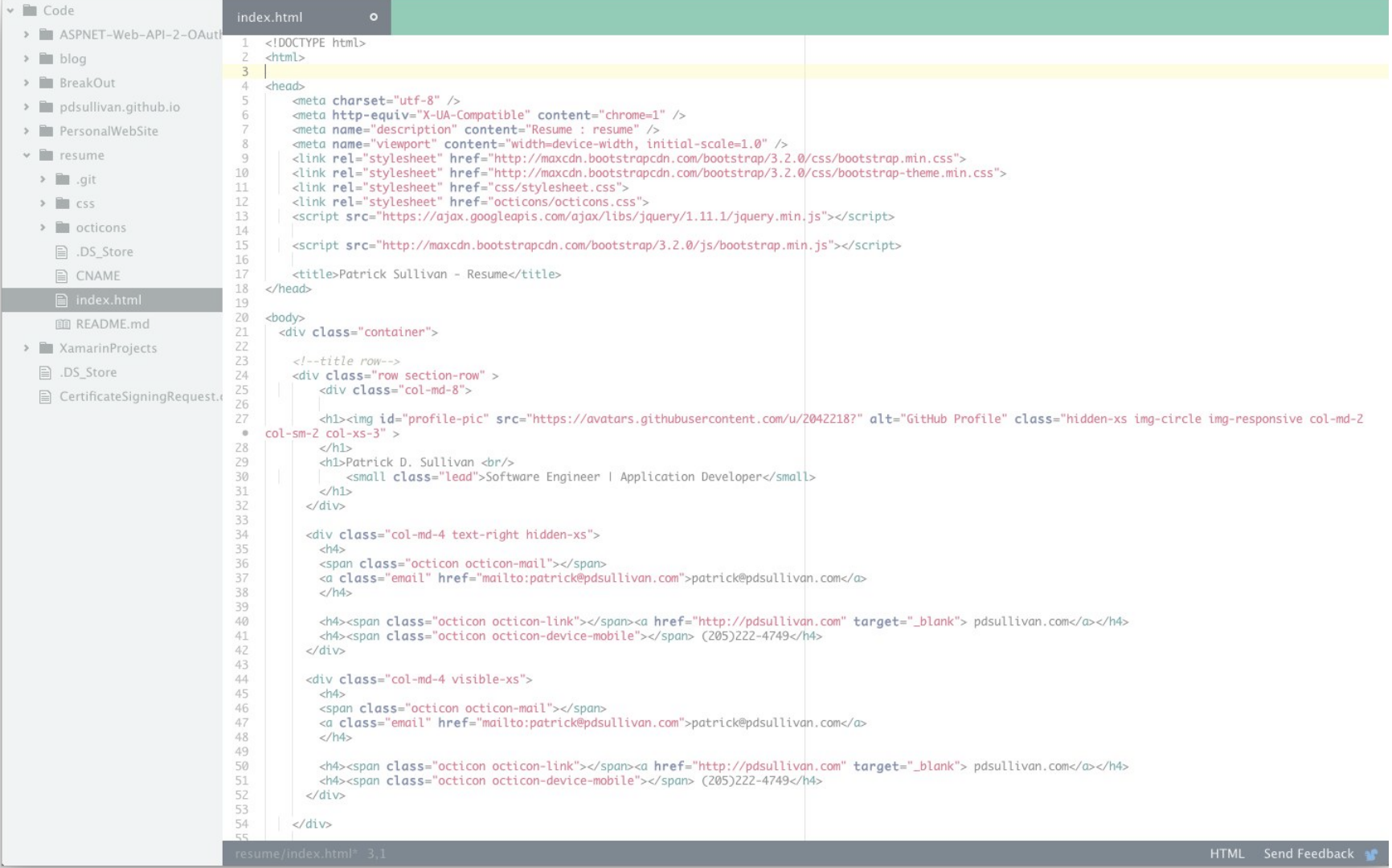
ZEPLIN

Sketch

Xd

Ps

Designer


Code
ASPNET-Web-API-2-OAuth
blog
BreakOut
pdsullivan.github.io
PersonalWebSite
resume
.git
css
octicons
.DS_Store
CNAME
index.html
README.md
XamarinProjects
.DS_Store
CertificateSigningRequest.
index.html
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<meta http-equiv="X-UA-Compatible" content="chrome=1" />
<meta name="description" content="Resume : resume" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
<link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap-theme.min.css">
<link rel="stylesheet" href="css/stylesheet.css">
<link rel="stylesheet" href="octicons/octicons.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
<script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script>
<title>Patrick Sullivan - Resume</title>
</head>
<body>

<!--title row-->

<h1><img id="profile-pic" src="https://avatars.githubusercontent.com/u/2042218?" alt="GitHub Profile" class="hidden-xs img-circle img-responsive col-md-2 col-sm-2 col-xs-3" >
</h1>
<h1>Patrick D. Sullivan
<small class="lead">Software Engineer | Application Developer</small>
</h1>

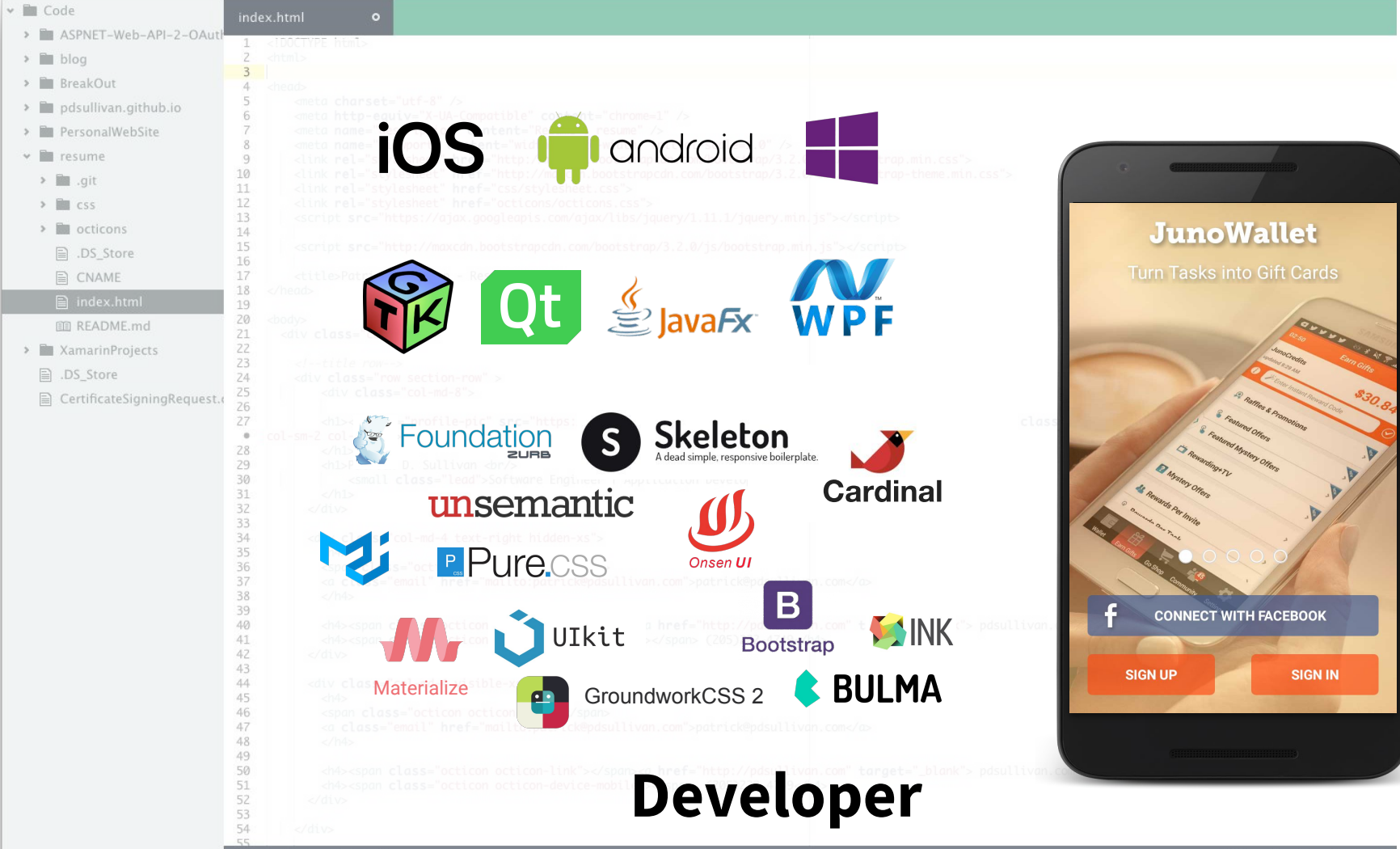
<h4>

patrick@pdsullivan.com
</h4>
<h4> pdsullivan.com</h4>
<h4> (205)222-4749</h4>

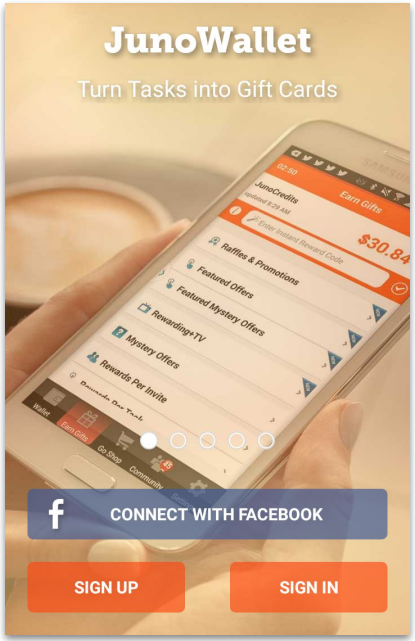
<h4>

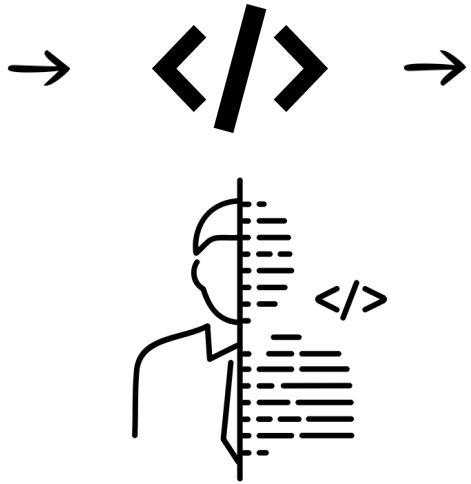
patrick@pdsullivan.com
</h4>
<h4> pdsullivan.com</h4>
<h4> (205)222-4749</h4>

resume/index.html* 3,1
HTML
Send Feedback

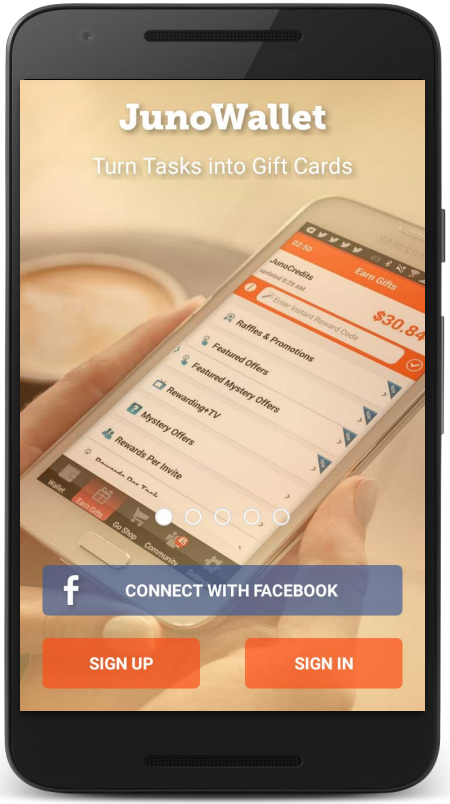

Code
ASPNET-Web-API-2-OAuth
blog
BreakOut
pdsullivan.github.io
PersonalWebSite
resume
.git
css
octicons
.DS_Store
CNAME
index.html
README.md
XamarinProjects
.DS_Store
CertificateSigningRequest
index.html
iOS
android
GTK
Qt
JavaFx
WPF
Foundation
ZURB
Skeleton
A dead simple, responsive boilerplate.
Cardinal
unsemantic
Pure.css
Onsen UI
Materialize
UIkit
Bootstrap
INK
GroundworkCSS 2
BULMA
JunoWallet
Turn Tasks into Gift Cards
CONNECT WITH FACEBOOK
SIGN UP
SIGN IN
Developer
resume/index.html* 3,1
HTML
Send Feedback

User Interface Design Today

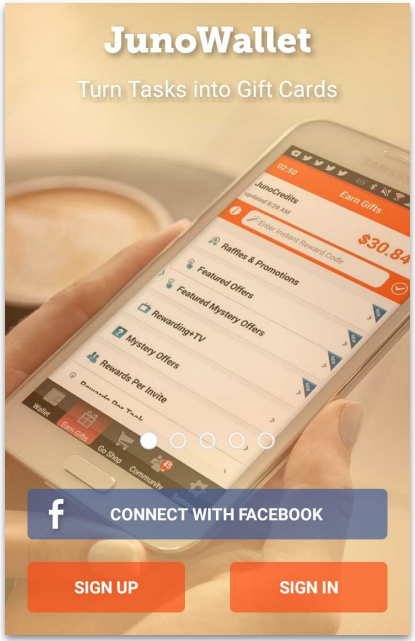


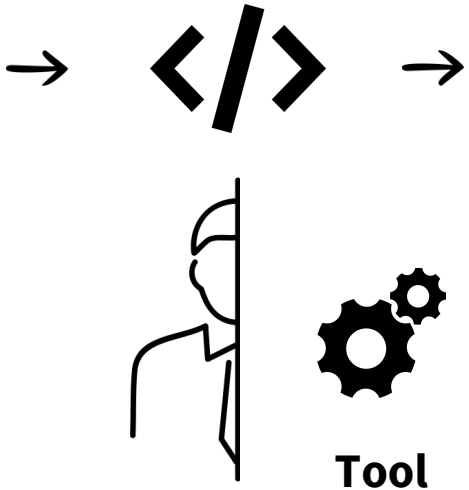


Designer Developer



Our Goal

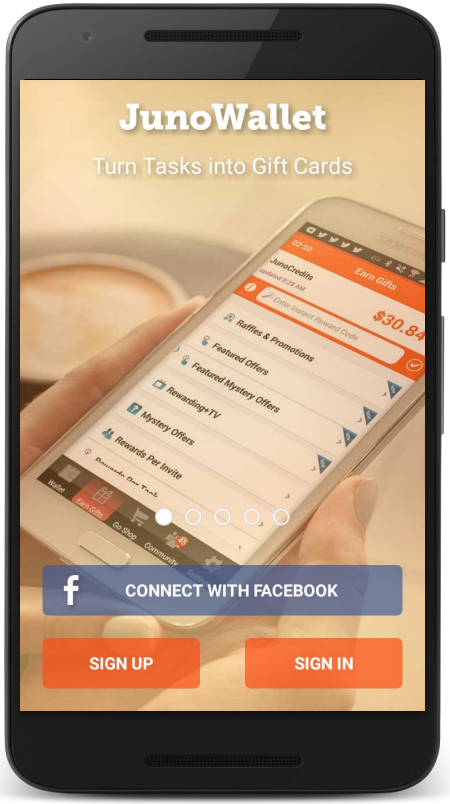




Designer

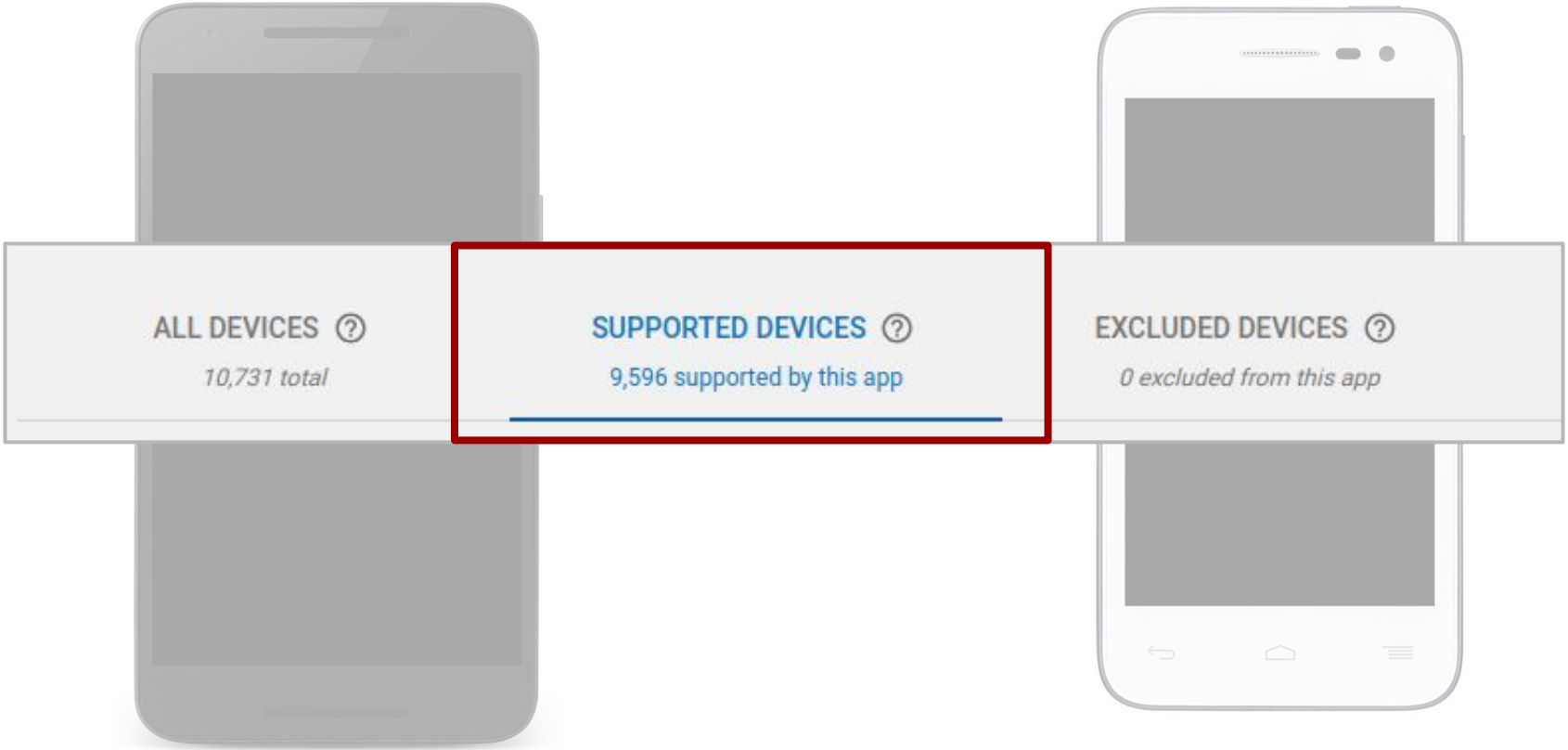
Tool

~~Developer~~



Preventing Layout Errors

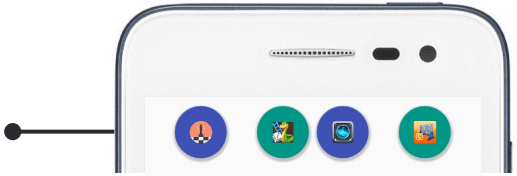
Preventing Layout Errors



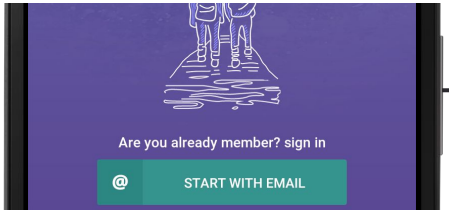
Preventing Layout Errors

Asobimasu

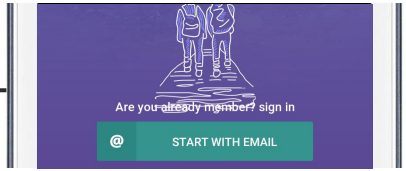
Centering



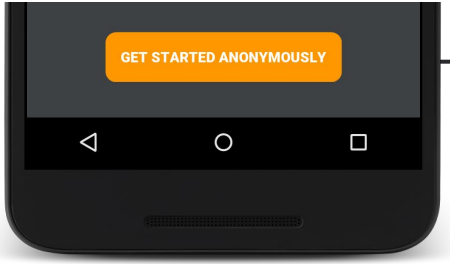
Amigo



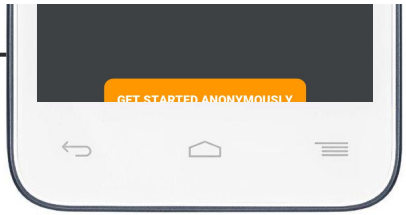
Overlaying Views



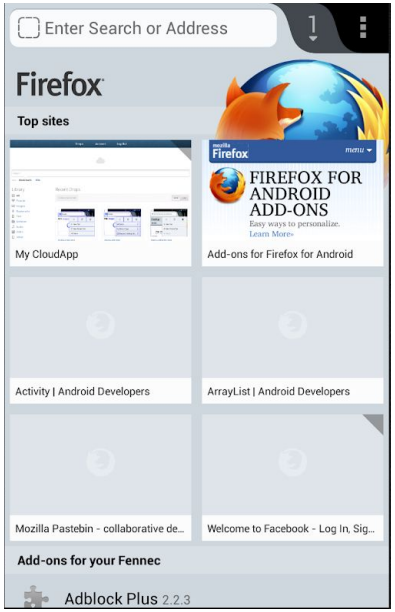
Candid

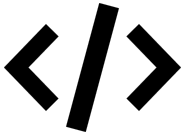


Out of Screen

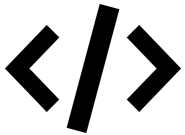


Improving Performance

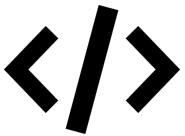




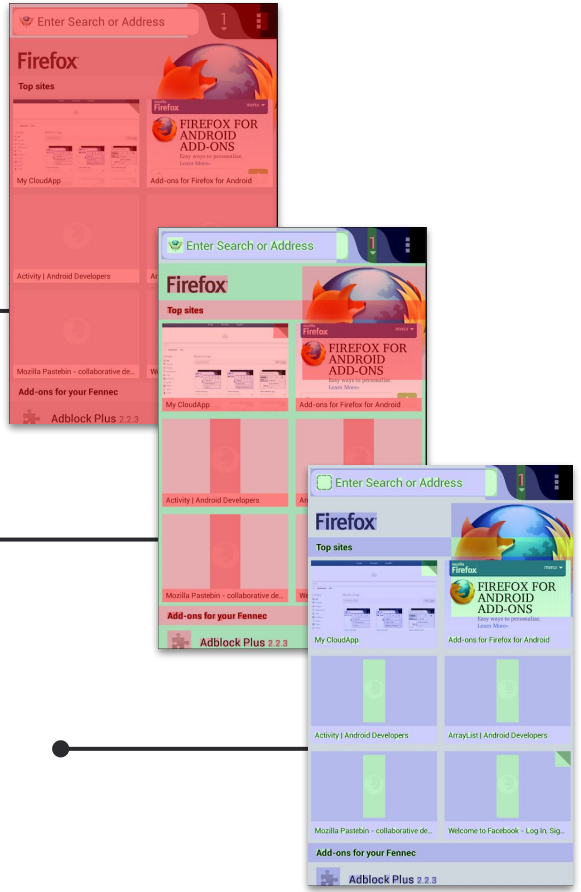
Layout #1



Layout #2



Layout #3



Slow

Fast

[Credits: Sriram Ramani]

Code Porting

Linear Layout (horizontal)

Linear Layout (vertical)

Relative Layout

android

Constraint Layout

>98%

Old / Deprecated

<2%

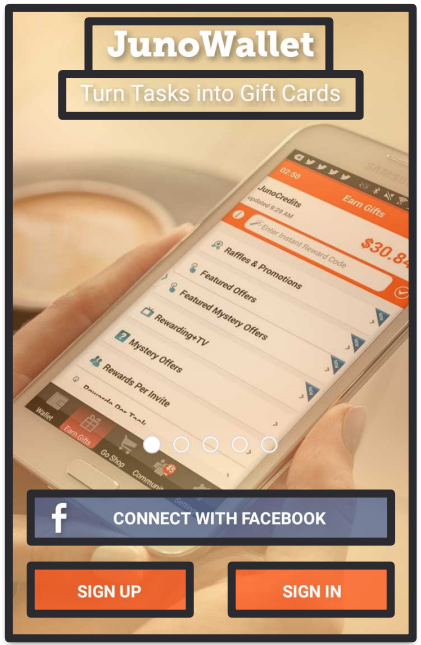
Automating End-User Design

Developer →

← Designer

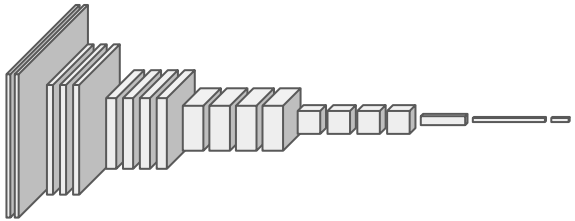
picture created by freepik
[Credits: Radu Cristian]

Machine Learning Approach



Design Specification

Training

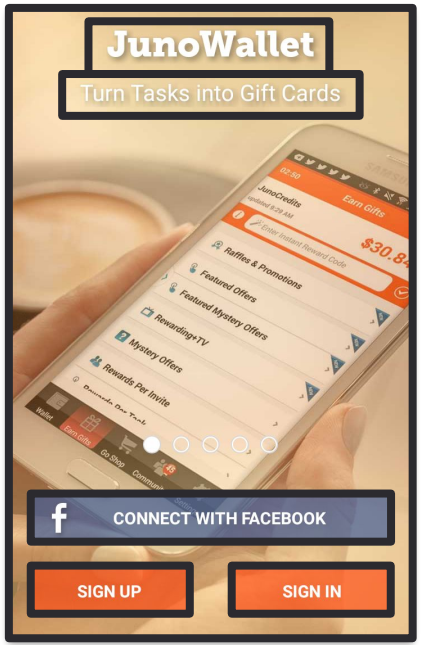


- ✗ Syntactically Incorrect
- ✗ No Formal Guarantees
- ✗ Robustness
- ✗ User Feedback

GitHub

Google Play

Programming Languages Approach



Design Specification

Layout Formalization

Robustness Properties

User Feedback

Layout Synthesis

List of Devices

Probabilistic Model

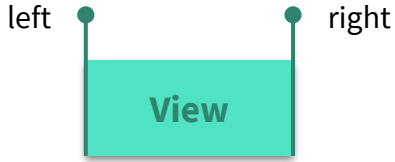
Layout

GitHub

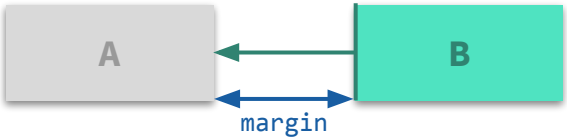
Google Play

InferUI

Relational Layout Constraints

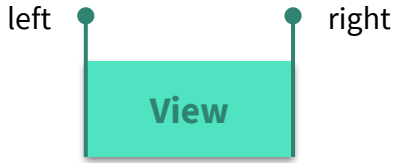


Align left of B to right of A

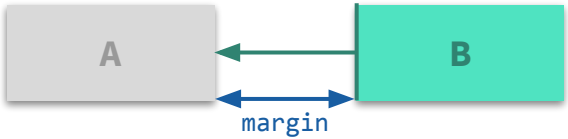



```
B.left = A.right + margin
```

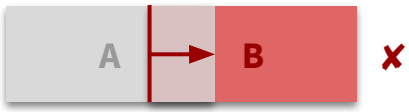
Relational Layout Constraints



Align left of B to right of A

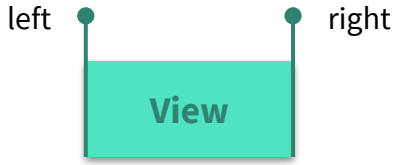



```
B.left = A.right + 0
```

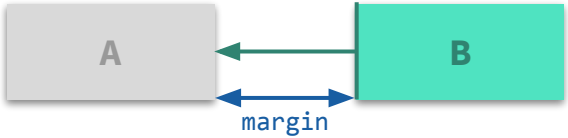


✘

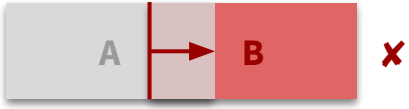
Relational Layout Constraints



Align left of B to right of A

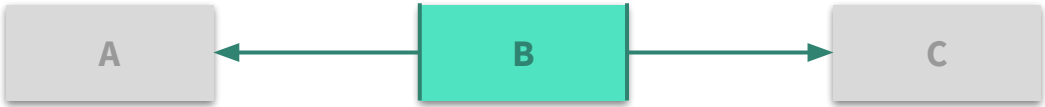



```
B.left = A.right + margin
```

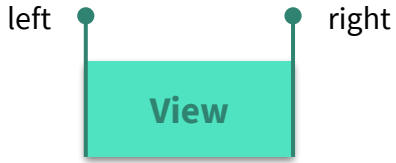


✘

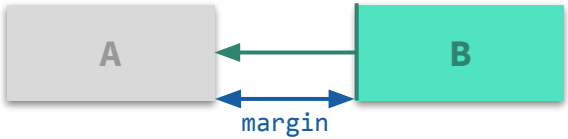
Center B in between right of A and left of C



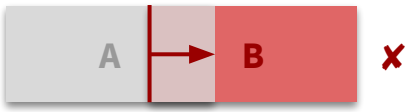
Relational Layout Constraints



Align left of B to right of A



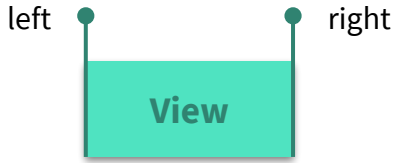

```
B.left = A.right + margin
```



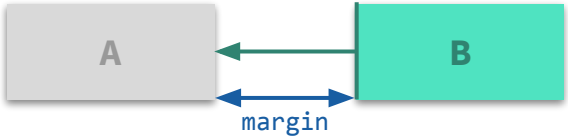
✘

Center B in between right of A and left of C

Relational Layout Constraints

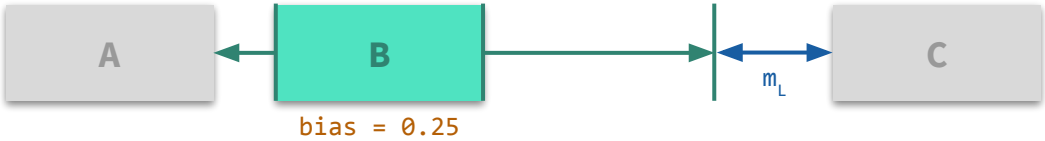


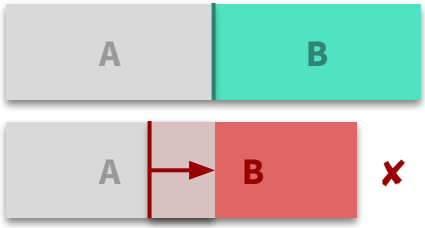
Align left of B to right of A



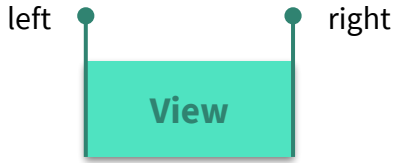

```
B.left = A.right + margin
```

Center B in between right of A and left of C

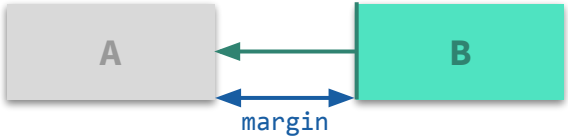




Relational Layout Constraints

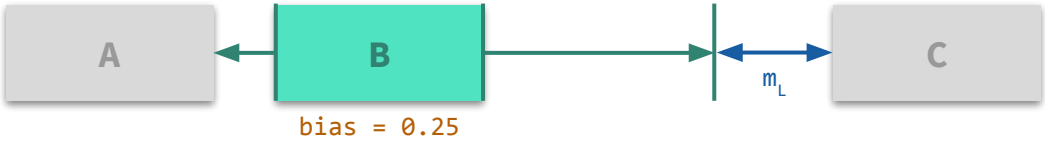


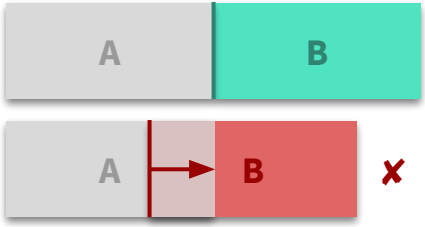
Align left of B to right of A



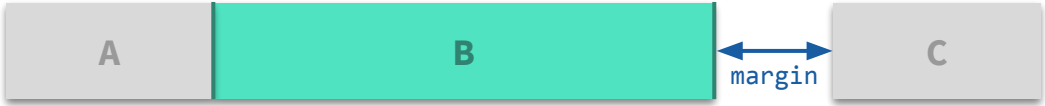

```
B.left = A.right + margin
```

Center B in between right of A and left of C

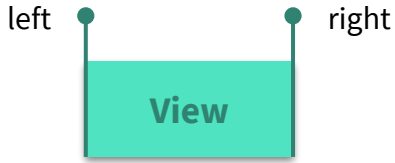




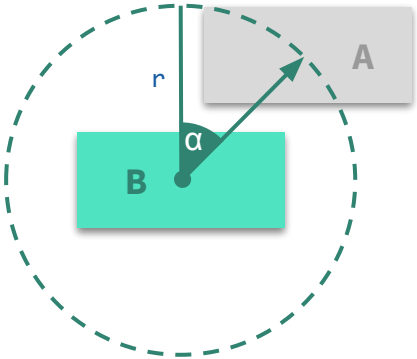
Center B in between right of A and left of C (Dynamic Size)



Relational Layout Constraints

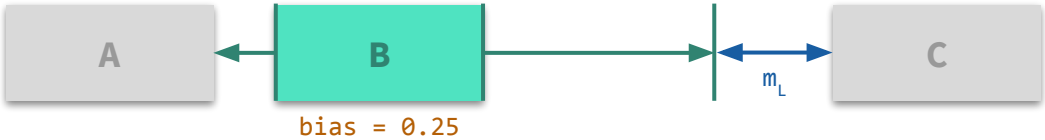


Align Center of A to Center of B at an Angle + Distance



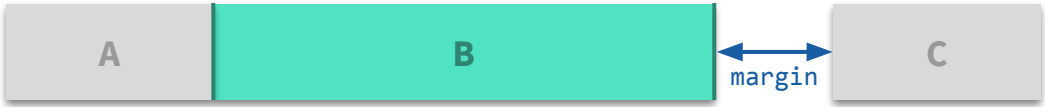
$$B.left + B.right = 2r*\sin(\alpha) + A.left + A.right$$

Center B in between right of A and left of C



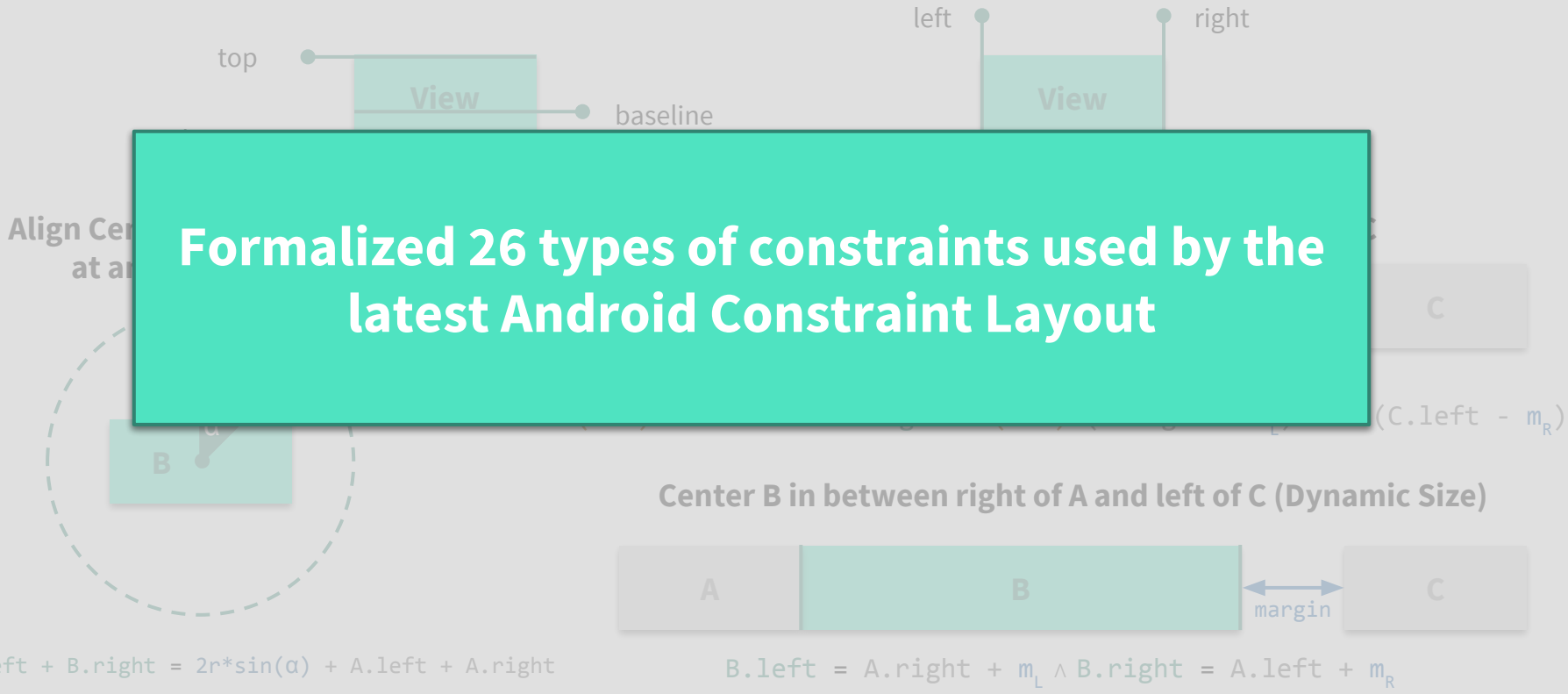
$$(1-b)*B.left + b*B.right = (1-b)*(A.right + m_L) + b*(C.left - m_R)$$

Center B in between right of A and left of C (Dynamic Size)



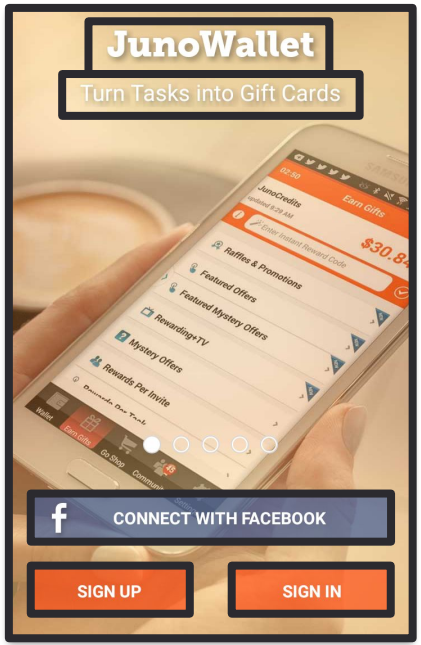
$$B.left = A.right + m_L \wedge B.right = A.left + m_R$$

Relational Layout Constraints



Formalized 26 types of constraints used by the latest Android Constraint Layout

Overview



Design Specification

Layout Formalization

Robustness Properties

User Feedback

Layout Synthesis

List of Devices

Probabilistic Model

Layout

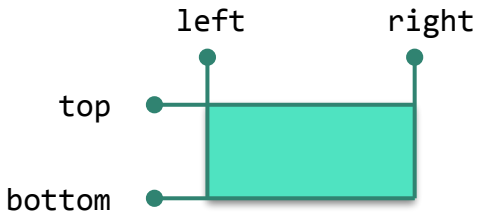
GitHub

Google Play

InferUI

Layout Solving

Goal
Compute absolute position of all the views

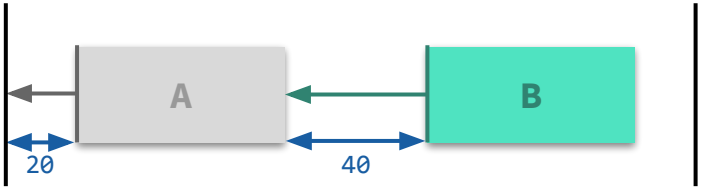


Layout Solving

Goal
Compute absolute position of all the views

```
A.left  = ?        B.left  = ?
A.right = ?        B.right = ?
```

} **v**: Absolute Horizontal Positions




```
B.left = A.right + 40
A.left = screen.left + 20
```

} **c**: Horizontal Position Constraints

```
screen.left = 0 ∧ screen.right = 240
```

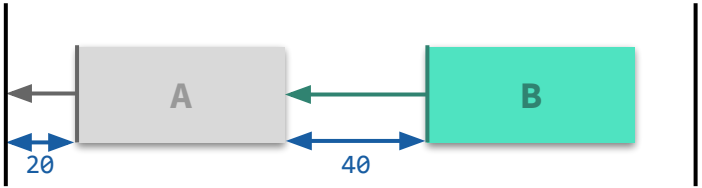
} screen: Target Device

Layout Solving

Goal
Compute absolute position of all the views

```
A.left  = 20        B.left  = ?
A.right = ?         B.right = ?
```

} **v**: Absolute Horizontal Positions




```
B.left = A.right + 40
A.left =      0      + 20
```

} **c**: Horizontal Position Constraints

```
screen.left = 0 ∧ screen.right = 240
```

} screen: Target Device

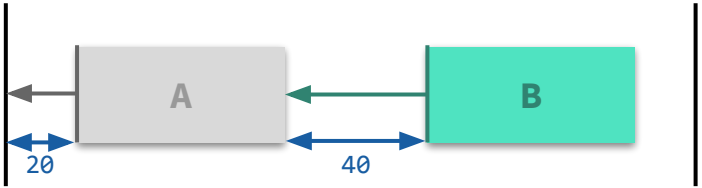
Layout Solving

Goal

Compute absolute position of all the views

```
A.left  = 20        B.left  = 140
A.right = 100       B.right = 220
```

v: Absolute Horizontal Positions




```
B.left = A.right + 40
A.left =      0      + 20
```

c: Horizontal Position Constraints

```
screen.left = 0 ∧ screen.right = 240
```

screen: Target Device

```
A.right - A.left = 80
B.right - B.left = 80
```

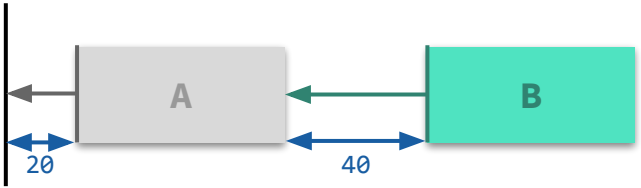
s: Horizontal Size Constraints

$$\mathbf{v} \models \Psi_{layout}(screen, \mathbf{c}, \mathbf{s})$$

Layout Solving vs Layout Synthesis

Goal
Compute absolute position of all the views

```
A.left  = ?        B.left  = ?
A.right = ?        B.right = ?
```

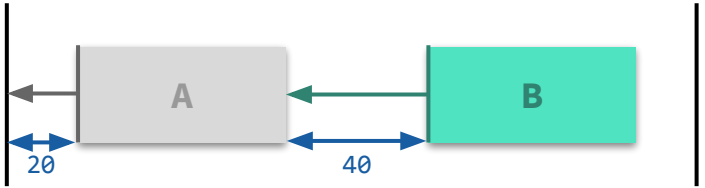



```
B.left = A.right + 40
A.left = screen.left + 20
screen.left = 0 ∧ screen.right = 240
A.right - A.left = 80
B.right - B.left = 80
```

$\mathbf{v} \models \Psi_{layout}(screen, \mathbf{c}, \mathbf{s})$

Goal
Synthesize position and size constraints

```
A.left  = 0        B.left  = 120
A.right = 80       B.right = 200
```




```
?
?
screen.left = 0 ∧ screen.right = 240
?
?
```

$\mathbf{c} \wedge \mathbf{s} \models \Psi_{syn}(screen, \mathbf{v})$

Layout Solving vs Layout Synthesis

Approach

1. **Encode the synthesis program as a logical formula** $\psi_{syn}(\texttt{screen}, \texttt{v})$
2. **Find a satisfying assignment to** $\texttt{c} \wedge \texttt{s} \models \psi_{syn}(\texttt{screen}, \texttt{v})$

$\texttt{v} \models \psi_{layout}(\texttt{screen}, \texttt{c}, \texttt{s})$

$\texttt{c} \wedge \texttt{s} \models \psi_{syn}(\texttt{screen}, \texttt{v})$

Layout Synthesis

$$\psi_{syn}(\text{screen}, \mathbf{v}) = \underbrace{\varphi_{position}(\text{screen}, \mathbf{v})}_{\text{Input Specification}} \wedge \underbrace{\varphi_{constraints}(\mathbf{v})}_{\text{Constraints Encoding}} \wedge \underbrace{\varphi_{valid}(\mathbf{v}) \wedge \varphi_{acyclic}(\mathbf{v})}_{\text{Ensure Constraints are Valid}}$$

Layout Synthesis

$$\psi_{syn}(\text{screen}, \mathbf{v}) = \underbrace{\varphi_{position}(\text{screen}, \mathbf{v})}_{\text{Input Specification}} \wedge \underbrace{\varphi_{constraints}(\mathbf{v})}_{\text{Constraints Encoding}} \wedge \underbrace{\varphi_{valid}(\mathbf{v}) \wedge \varphi_{acyclic}(\mathbf{v})}_{\text{Ensure Constraints are Valid}}$$

$$\varphi_{position} \stackrel{\text{def}}{=} \begin{aligned} &\text{screen.left} = 0 \wedge \\ &\text{screen.right} = 240 \wedge \\ &\quad A.left = 0 \wedge \\ &\quad A.right = 80 \wedge \\ &\quad B.left = 120 \wedge \\ &\quad B.right = 200 \end{aligned}$$

Layout Synthesis

$$\psi_{syn}(screen, v) = \underbrace{\varphi_{position}(screen, v)}_{\text{Input Specification}} \wedge \underbrace{\varphi_{constraints}(v)}_{\textbf{Constraints Encoding}} \wedge \underbrace{\varphi_{valid}(v) \wedge \varphi_{acyclic}(v)}_{\text{Ensure Constraints are Valid}}$$

Layout Synthesis

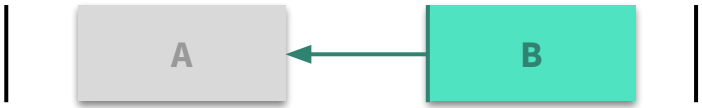
What are the possible constraints for view B?

$$C(\mathtt{B},\ \mathbf{v},\ \mathtt{screen}) = \{c_1^B,\ c_2^B,\ c_3^B,\ \ldots\ ,\ c_n^B\}$$

Layout Synthesis

What are the possible constraints for view B?

$$C(\mathrm{B},\ \mathbf{v},\ \mathrm{screen}) = \{c_1^B,\ c_2^B,\ c_3^B,\ \ldots\ ,\ c_n^B\}$$



$$\mathrm{B.left} = \mathrm{A.right} + m_L \quad (c_1)$$

Layout Synthesis

What are the possible constraints for view B?

$$C(\mathtt{B},\ \mathtt{v},\ \mathtt{screen}) = \{c_1^B,\ c_2^B,\ c_3^B,\ \ldots\ ,\ c_n^B\}$$

$B.left = A.right + m_L \quad (c_1)$

$B.left = screen.right + m_L \quad (c_2)$

Layout Synthesis

What are the possible constraints for view B?

$C(B, v, \text{screen}) = \{c_1^B, c_2^B, c_3^B, \ldots, c_n^B\}$

$B.left = A.right + m_L \quad (c_1)$

$B.left = screen.right + m_L \quad (c_2)$

$(1-b)*B.left + b*B.right = (1-b)*(A.right + m_L) + b*(screen.right - m_R) \quad (c_3)$

Layout Synthesis

What are the possible constraints for view B?

$C(B, v, screen) = \{c_1^B, c_2^B, c_3^B, \ldots, c_n^B\}$ ← **O(|v|²)**

$B.left = A.right + m_L \quad (c_1)$

$B.left = screen.right + m_L \quad (c_2)$

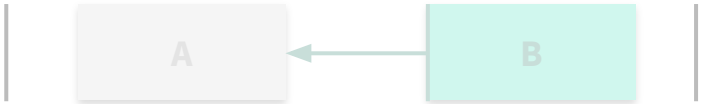
$(1-b)*B.left + b*B.right = (1-b)*(A.right + m_L) + b*(screen.right - m_R) \quad (c_3)$

Layout Synthesis

What are the possible constraints for view B?

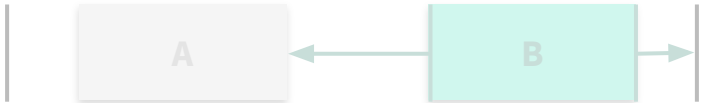
$C(B, v, screen) = \{c_1^B, c_2^B, c_3^B, \ldots, c_n^B\}$ ← **O(|v|²)**

Select exactly one constraint for each view



$B.left = A.right + m_L \quad (c_1)$

$B.left = screen.right + m_L \quad (c_2)$



$(1-b)*B.left + b*B.right = (1-b)*(A.right + m_L) + b*(screen.right - m_R) \quad (c_3)$

Layout Synthesis

What are the possible constraints for view B?

$C(\texttt{B, v, screen}) = \{c_1^B, c_2^B, c_3^B, \ldots, c_n^B\}$ ← **O(|v|²)**

Select exactly one constraint for each view

Boolean variables denoting whether constraint was selected

$$
\begin{cases}
g_1 \Rightarrow (\texttt{B.left} = \texttt{A.right} + m_L) & (c_1) \\
g_2 \Rightarrow (\texttt{B.left} = \texttt{A.left} + m_L) & (c_2) \\
\qquad\qquad \ldots & \\
g_n \Rightarrow (\texttt{((1-b)*B.left + b*B.right} = \ldots) & (c_n)
\end{cases}
$$

Layout Synthesis

What are the possible constraints for view B?

$C(\texttt{B}, \texttt{v}, \texttt{screen}) = \{c_1^B, c_2^B, c_3^B, \ldots, c_n^B\}$ ← **O(|v|²)**

$$\varphi_{\texttt{constraints}}(\texttt{screen}, \texttt{v}) \stackrel{\text{def}}{=} \underbrace{\left(\bigwedge_{k=0}^{C(\texttt{B}, \texttt{v}, \texttt{screen})} g_k \Rightarrow [\![c_k]\!] \right)}_{\text{View constraints}} \wedge \underbrace{g_0 + \ldots + g_{|C(\texttt{B}, \texttt{v}, \texttt{screen})|} = 1}_{\text{Select exactly one constraint for each view}}$$

Boolean variables denoting whether constraint was selected

$$\begin{cases} g_1 \Rightarrow (\texttt{B.left} = \texttt{A.right} + m_L) & (c_1) \\ g_2 \Rightarrow (\texttt{B.left} = \texttt{A.left} + m_L) & (c_2) \\ \ldots \\ g_n \Rightarrow (\texttt{(1-b)*B.left} + \texttt{b*B.right} = \ldots) & (c_n) \end{cases}$$

Layout Synthesis

What are the possible constraints for view B?

$$C(\texttt{B}, \texttt{v}, \texttt{screen}) = \{c_1^B, c_2^B, c_3^B, \ldots, c_n^B\} \quad \leftarrow \mathcal{O}(|\texttt{v}|^2)$$

$$\varphi_{\texttt{constraints}}(\texttt{screen}, \texttt{v}) \stackrel{\text{def}}{=} \bigwedge_{i=0}^{|\texttt{v}|} \Big(\bigwedge_{k=0}^{C(v_i, \texttt{v}, \texttt{screen})} g_k^i \Rightarrow [\![c_k^i]\!] \Big) \wedge g_0^i + \ldots + g_{|C(v_i, \texttt{v}, \texttt{screen})|}^i = 1$$

View constraints

Select exactly one constraint for each view

Boolean variables denoting whether constraint was selected

$$\begin{cases} g_1 \Rightarrow (\texttt{B.left} = \texttt{A.right} + m_L) & (c_1) \\ g_2 \Rightarrow (\texttt{B.left} = \texttt{A.left} + m_L) & (c_2) \\ \ldots \\ g_n \Rightarrow (\texttt{((1-b)*B.left + b*B.right} = \ldots) & (c_n) \end{cases}$$

Layout Synthesis

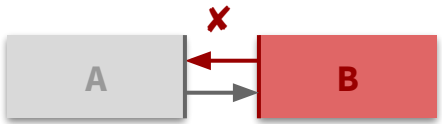
$$\psi_{syn}(screen, v) = \underbrace{\varphi_{position}(screen, v)}_{\text{Input Specification}} \wedge \underbrace{\varphi_{constraints}(v)}_{\textbf{Constraints Encoding}} \wedge \underbrace{\varphi_{valid}(v) \wedge \varphi_{acyclic}(v)}_{\text{Ensure Constraints are Valid}}$$

Layout Synthesis

$$\psi_{syn}(screen, v) = \underbrace{\varphi_{position}(screen, v)}_{\text{Input Specification}} \wedge \underbrace{\varphi_{constraints}(v)}_{\text{Constraints Encoding}} \wedge \underbrace{\varphi_{valid}(v) \wedge \varphi_{acyclic}(v)}_{\text{Ensure Constraints are Valid}}$$

$$\varphi_{valid}(v) \stackrel{\text{def}}{=} (m_L \geq 0) \wedge (m_R \geq 0) \wedge (0 \leq b \leq 1) \wedge (0 \leq \alpha < 360) \wedge (r \geq 0)$$

$\varphi_{acyclic}(v)$ $\stackrel{\text{def}}{=}$



Layout Synthesis

$$\psi_{syn}(screen, \mathbf{v}) = \underbrace{\varphi_{position}(screen, \mathbf{v})}_{\text{Input Specification}} \wedge \underbrace{\varphi_{constraints}(\mathbf{v})}_{\text{Constraints Encoding}} \wedge \underbrace{\varphi_{valid}(\mathbf{v}) \wedge \varphi_{acyclic}(\mathbf{v})}_{\text{Ensure Constraints are Valid}}$$

$$\psi_{syn}(screen, \mathbf{v}) \rightarrow \textbf{Solver} \rightarrow \mathbf{c} \wedge \mathbf{s}$$

$$\mathbf{c} \wedge \mathbf{s} \models \psi_{syn}(screen, \mathbf{v})$$

Layout Synthesis

$$\psi_{syn}(screen, \mathbf{v}) = \underbrace{\varphi_{position}(screen, \mathbf{v})}_{\text{Input Specification}} \wedge \underbrace{\varphi_{constraints}(\mathbf{v})}_{\text{Constraints Encoding}} \wedge \underbrace{\varphi_{valid}(\mathbf{v}) \wedge \varphi_{acyclic}(\mathbf{v})}_{\text{Ensure Constraints are Valid}}$$

$$\psi_{syn}(screen, \mathbf{v}) \rightarrow \textbf{Solver} \rightarrow \mathbf{c} \wedge \mathbf{s}$$

$$\mathbf{c} \wedge \mathbf{s} \models \psi_{syn}(screen, \mathbf{v})$$

Are We Done?

Layout Synthesis

$$\psi_{syn}(screen, \mathbf{v}) = \underbrace{\varphi_{position}(screen, \mathbf{v})}_{\text{Input Specification}} \wedge \underbrace{\varphi_{constraints}(\mathbf{v})}_{\text{Constraints Encoding}} \wedge \underbrace{\varphi_{valid}(\mathbf{v}) \wedge \varphi_{acyclic}(\mathbf{v})}_{\text{Ensure Constraints are Valid}}$$

$$\psi_{syn}(screen, \mathbf{v}) \rightarrow \textbf{Solver} \rightarrow \mathbf{c} \wedge \mathbf{s}$$

$$\mathbf{c} \wedge \mathbf{s} \models \psi_{syn}(screen, \mathbf{v})$$

Are We Done? **No**

Layout Synthesis

Considers only Single Device

$$\psi_{syn}(screen, \mathbf{v}) = \underbrace{\varphi_{position}(screen, \mathbf{v})}_{\text{Input Specification}} \wedge \underbrace{\varphi_{constraints}(\mathbf{v})}_{\text{Constraints Encoding}} \wedge \underbrace{\varphi_{valid}(\mathbf{v}) \wedge \varphi_{acyclic}(\mathbf{v})}_{\text{Ensure Constraints are Valid}}$$

max **4** views

Scalability

13%

Generalization

Are We Done? **No**

Layout Synthesis

Considers only Single Device

$$\psi_{syn}(screen, v) = \underbrace{\varphi_{position}(screen, v)}_{\text{Input Specification}} \wedge \underbrace{\varphi_{constraints}(v)}_{\text{Constraints Encoding}} \wedge \underbrace{\varphi_{valid}(v) \wedge \varphi_{acyclic}(v)}_{\text{Ensure Constraints are Valid}}$$

Google Play

GitHub

Probabilistic Model

User Feedback

Generalize to Multiple Devices

Are We Done? No

Natural Layouts

$$\psi_{syn}(screen, \mathbf{v}) = \varphi_{position}(screen, \mathbf{v}) \wedge \varphi_{constraints}(\mathbf{v}) \wedge \varphi_{valid}(\mathbf{v}) \wedge \varphi_{acyclic}(\mathbf{v})$$

$$\max \sum_{i=0}^{|c|} \log P(c_i \mid screen, \mathbf{v})$$

Probability of each selected constraint according to learned probabilistic model

$$\psi_{syn}(screen, \mathbf{v}) \rightarrow \textbf{Solver} \rightarrow \mathbf{c} \wedge \mathbf{s}$$

the most likely

synthesize ~~any~~ layout

that satisfies the specification

Natural Layouts

$$\psi_{syn}(screen, \mathbf{v}) = \varphi_{position}(screen, \mathbf{v}) \wedge \varphi_{constraints}(\mathbf{v}) \wedge \varphi_{valid}(\mathbf{v}) \wedge \varphi_{acyclic}(\mathbf{v})$$

$$\max \sum_{i=0}^{|c|} \log P(c_i \mid screen, \mathbf{v})$$

Probability of each selected constraint according to learned probabilistic model

max **4** views → **12** views

Scalability

13% → **75%**

Generalization

Probabilistic Model

Learn a probabilistic model that predicts likelihood that a constraint is used

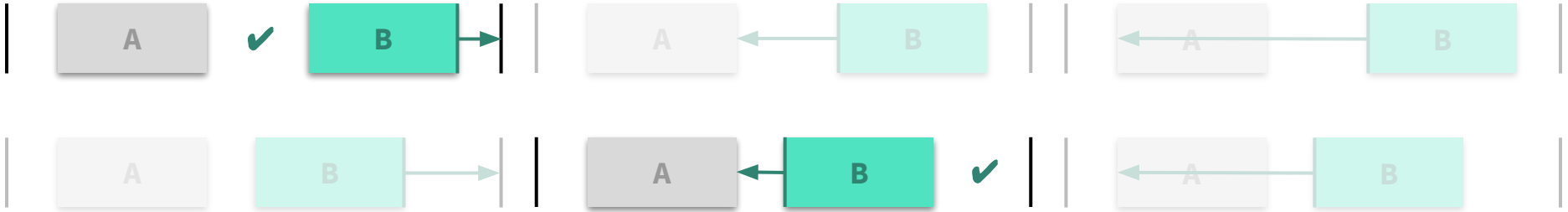
$$P(c \mid \text{screen}, \mathbf{v}) \rightarrow \mathbb{R}^{\langle 0,\ 1\rangle}$$

Which constraint is most likely?

Probabilistic Model

Learn a probabilistic model that predicts likelihood that a constraint is used

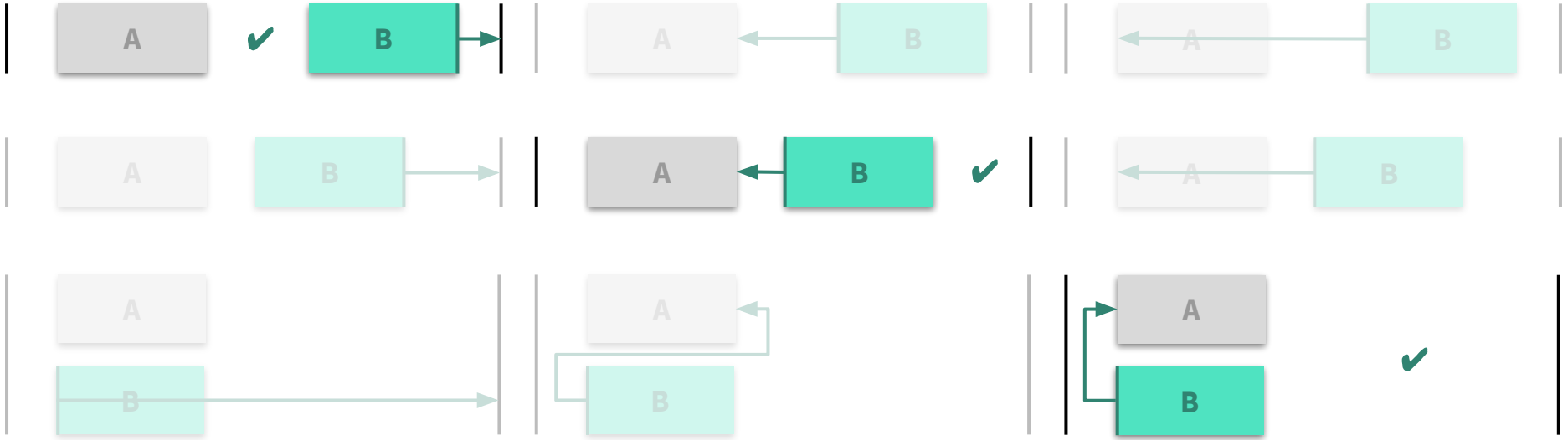
$$P(c \mid \text{screen}, \mathbf{v}) \rightarrow \mathbb{R}^{\langle 0,\ 1\rangle}$$



Probabilistic Model

Learn a probabilistic model that predicts likelihood that a constraint is used

$$P(c \mid \text{screen}, \mathbf{v}) \rightarrow \mathbb{R}^{\langle 0,\ 1\rangle}$$



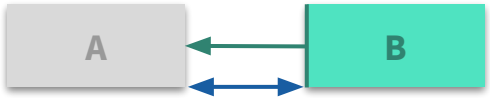
Probabilistic Model

$$P(c \mid \text{screen}, \mathbf{v}) = \frac{1}{Z(\text{screen}, \mathbf{v})} \prod P_f(c \mid f_k(c, \text{screen}, \mathbf{v}))^{w_k}$$

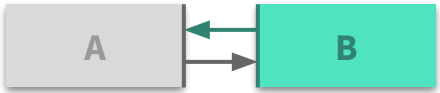
f_b Bias

b

f_d Distance



f_t Constraint Type

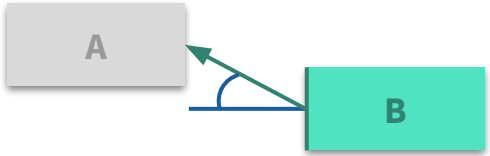


f_i Intersection

f_m Margins

$\langle m_L, m_R \rangle$

f_o Orientation



f_s Size

f_c Complexity

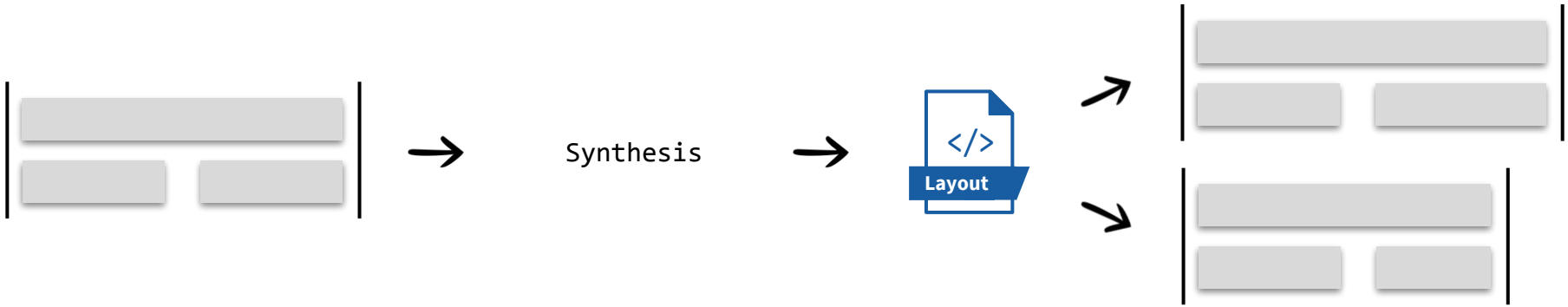
$b == 0$

$m_L == 0, m_R == 0$

All Trained using Maximum Likelihood Estimation

Generalize to Multiple Devices

1. Synthesize Layout Using Input Specification



2. Render on Multiple Devices

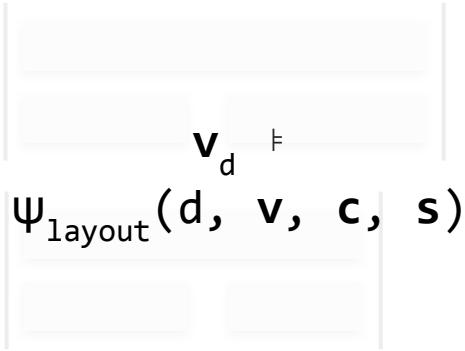
3. Check that Synthesized Layout Generalizes

Generalize to Multiple Devices

1. Synthesize Layout Using Input Specification

$$c \wedge s \models \psi_{syn}(screen, v)$$

2. Render on Multiple Devices



$$v_d \models \psi_{layout}(d, v, c, s)$$

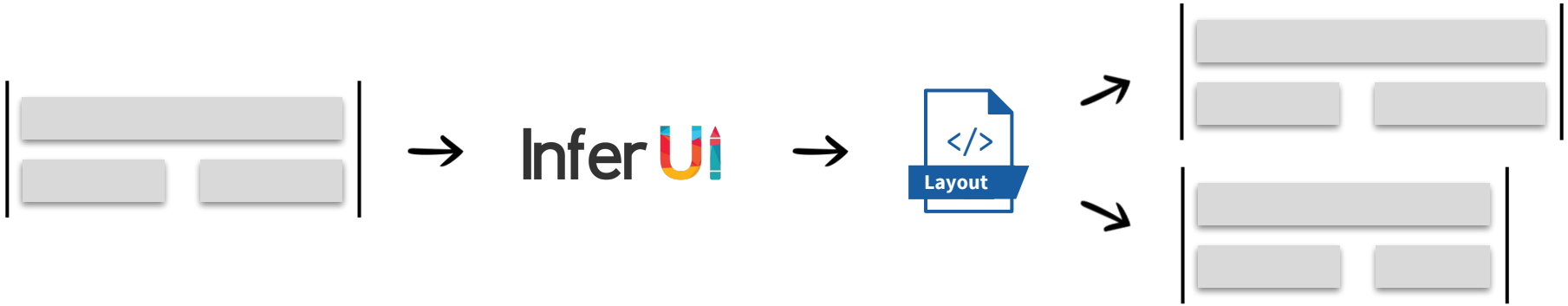
3. Check that Synthesized Layout Generalizes

$$\varphi_{preserve\_aspect\_ratio}(v, v_d) \wedge \varphi_{inside\_screen}(d, v_d) \wedge \varphi_{pixel\_perfect}(v_d) \wedge \varphi_{preserve\_order}(v, v_d) \wedge \varphi_{preserve\_centering}(v, v_d) \wedge \varphi_{preserve\_margins}(v, v_d)$$

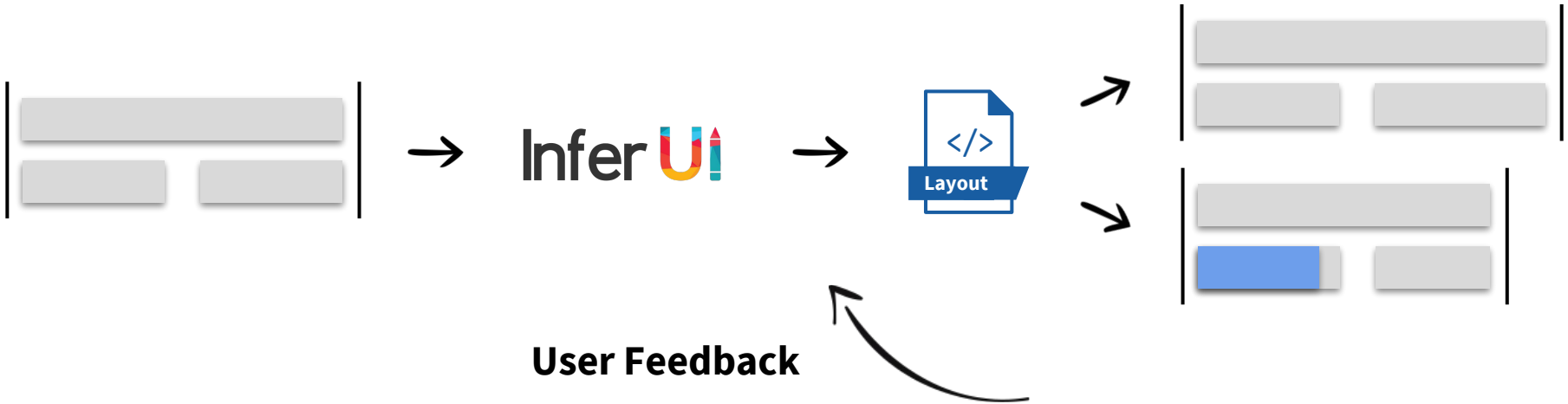
Set of Robustness Properties

All Steps Encoded as a Single Formula!

User Feedback



User Feedback



$$\varphi_{\texttt{user\_feedback}} \stackrel{\text{def}}{=} B_k.\texttt{left} = 120 \wedge B_k.\texttt{right} = 200$$

Evaluation

Infer UI

Top 500

Google Play

Top 500

GitHub

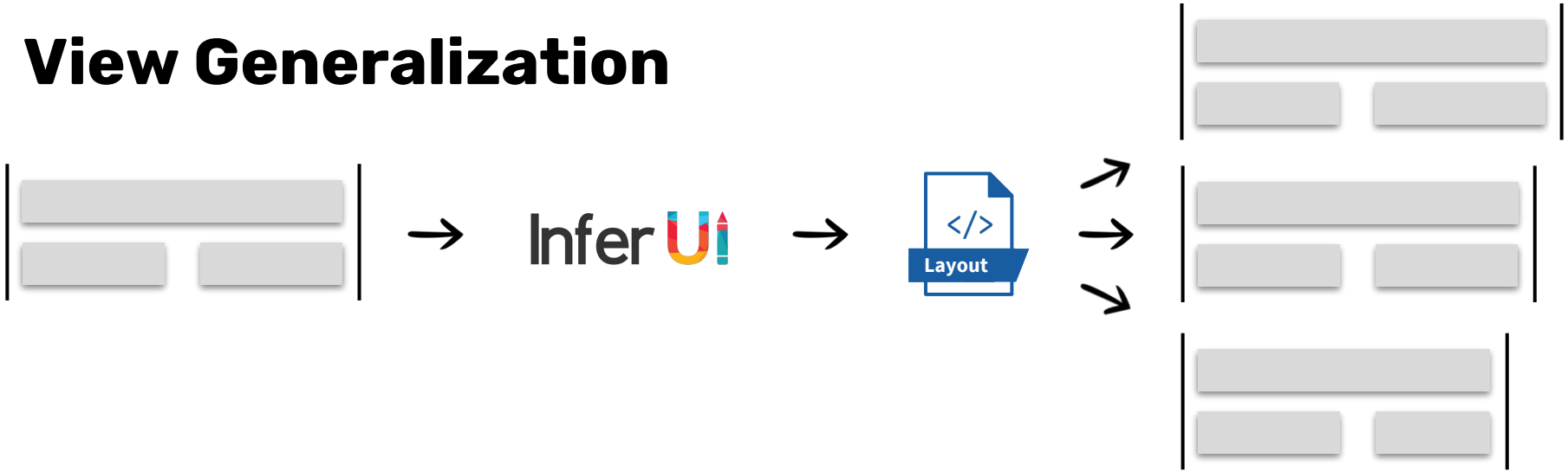
Evaluation

Synthesis Success Ratio / Runtime

	[2, 4)	[4, 8)	[8, 12)	[12, 16)	
Single Device	**100%** 37 ms	**100%** 59 ms	**100%** 129 ms	**100%** 519 ms	Success Ratio Average Synthesis Time
Multi Device	**99%** 44 ms	**93%** 95 ms	**73%** 314 ms	**56%** 3 s	

Number of Views

View Generalization

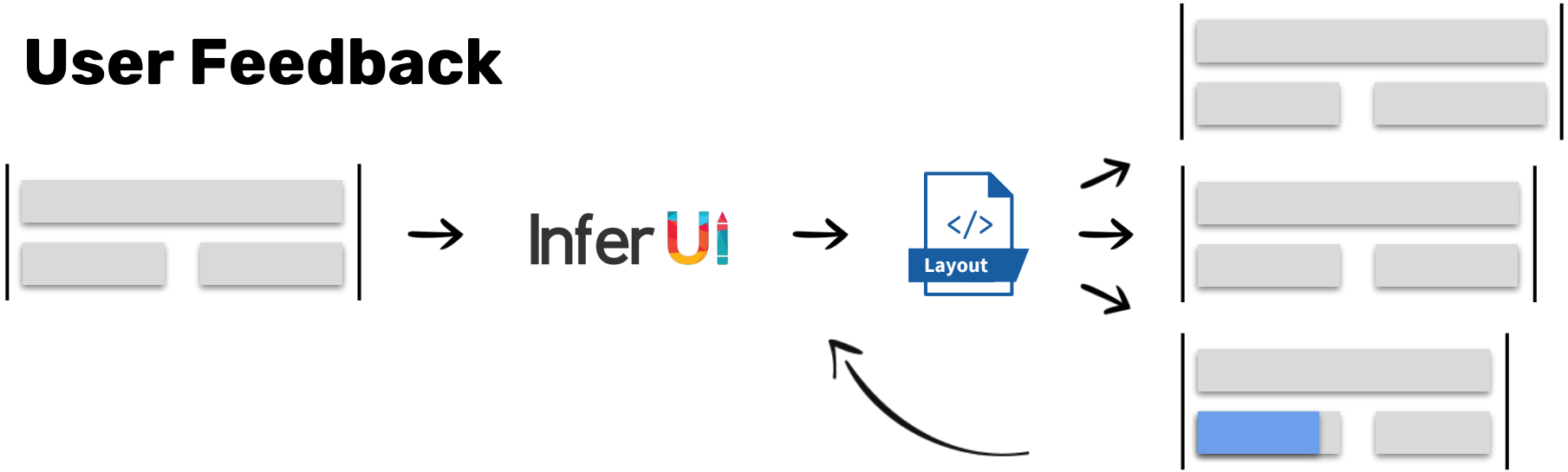


Percentage of Views That Correctly Generalize

13% → 75% → **92%**

$\Psi_{\texttt{single\_syn}}(\texttt{screen, v})$ + Probabilistic Model + Robustness Properties

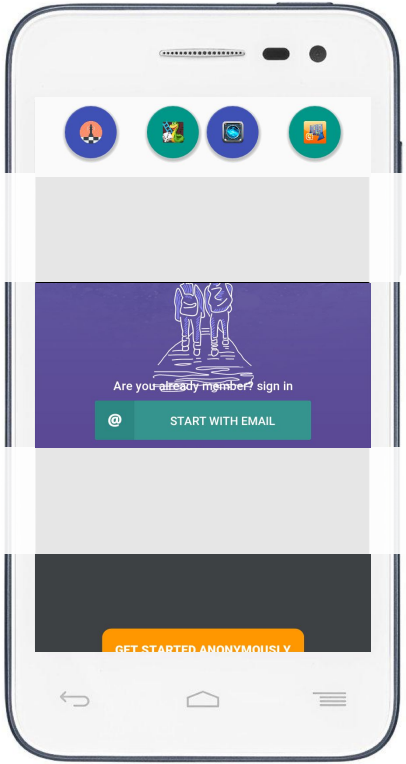
User Feedback



Number of User Feedbacks Required

63%	25%	8%	4%
no feedback	1 feedback	2 feedbacks	3+ feedbacks

Layout Errors



24
(5%)
$\neg\varphi_{\texttt{preserve\_margins}}$

4
(0.8%)
$\neg\varphi_{\texttt{preserve\_aspect\_ratio}}$

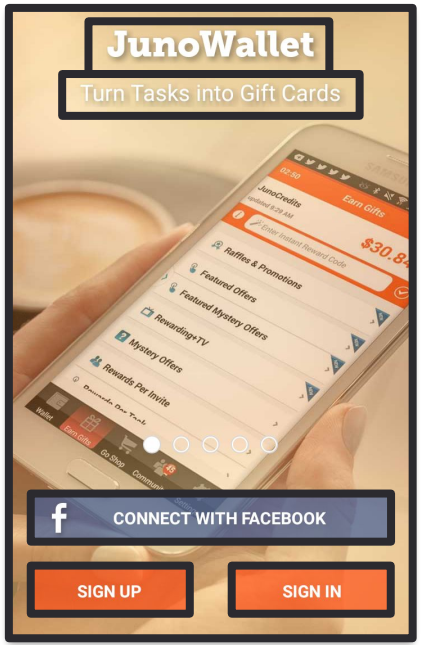
19
(4%)
$\neg\varphi_{\texttt{preserve\_order}}$

41
(8.5%)
$\neg\varphi_{\texttt{pixel\_perfect}}$

21
(4%)
$\neg\varphi_{\texttt{inside\_screen}}$

16
(3.3%)
$\neg\varphi_{\texttt{preserve\_centering}}$

InferUI



Design Specification

Layout Formalization

Robustness Properties

User Feedback

Layout Synthesis

List of Devices

Probabilistic Model

Layout

GitHub

Google Play