Creative UIs with Compose

Chris Horner

Design Systems at Cash App

DroidKaigi 2024

Cash App

We’re going to analyse and build a complex UI

But first...

一緒に複雑なUIの調査と構築を行います。

まず。。。

14:00 3G

Your Library

Playlists | Audiobooks | Albums | Artists | Do

Recents

- Liked Songs
  Playlist • 2427 songs
- Charcoal Grace
  Album • Caligula's Horse
- It Leads To This
  Album • The Pineapple Thief
- Versions of the Truth
  Album • The Pineapple Thief
- Union
  Album • Ihlo
- Casualties of Cool
  Album • Devin Townsend
- Petals
- Discover Weekly
  Playlist • Made for [illegible]

The World Breathes with Me
Caligula's Horse

Home | Search | Your Library

14:00 3G

GDG Australia & New Ze...

Jump to or search...

Catch up
3 new

Threads
Caught up

Later
0 items

Drafts &
0 drafts

Channels

- general
- melbourne
- showcase
- Add channel

Channels

- android
- declarative-ui
- design
- google-io
- jobs-general

Home | DMs | Activity

14:00 3G

Google Photos

Trip to New Zealand
10 years ago

Video spotlight

Me
2 years ago

Sat, 20 July
Melbourne and Carlton

Wed, 17 July

Thu, 11 July
Melbourne and Melbourne

Photos | Memories | Collections | Search

14:00 3G

Search podcasts or add RSS URL

All categories | True Crime | Comedy

FEATURED

Here's Why

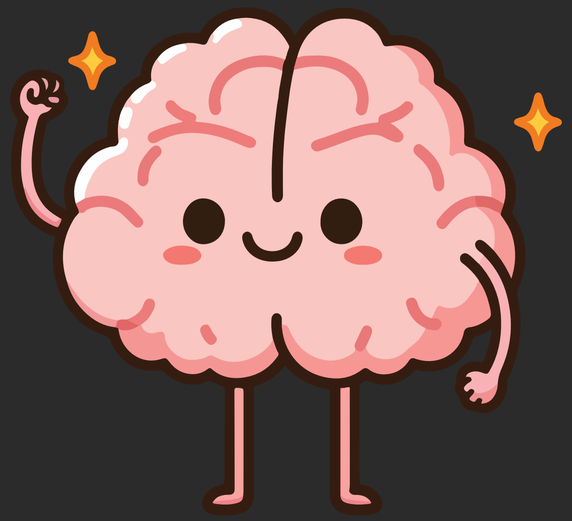
Bloomberg

Trending

SHOW ALL

- Hysterical
  Wondery | Pineapple Street Studios
- Assembly Required with Stacey Abrams
  Crooked Media
- The Weekly Show with Jon Stewart
  Comedy Central
- Where's Dia?
  Pushkin Industries

Podcasts | Filters | Discover | Up Next | Profile



14:00 3G

Your Library

Playlists | Audiobooks | Albums | Artists | Dc

Recents

Liked Songs
Playlist • 2427 songs

Charcoal Grace
Album • Caligula's Horse

It Leads To This
Album • The Pineapple Thief

Versions of the Truth
Album • The Pineapple Thief

Union
Album • Ihlo

Casualties of Cool
Album • Devin Townsend

Petals

The World Breathes with Me
Caligula's Horse

Discover Weekly

Home | Search | Your Library

14:00 3G

Search podcasts or add RSS URL

All categories | True Crime | Comedy

FEATURED

Here's Why
Bloomberg

Trending SHOW ALL

Hysterical
Wondery | Pineapple Street Studios

Assembly Required with Stacey Abrams
Crooked Media

The Weekly Show with Jon Stewart
Comedy Central

Where's Dia?
Pushkin Industries

Podcasts | Filters | Discover | Up Next | Profile

I arrange rectangles.

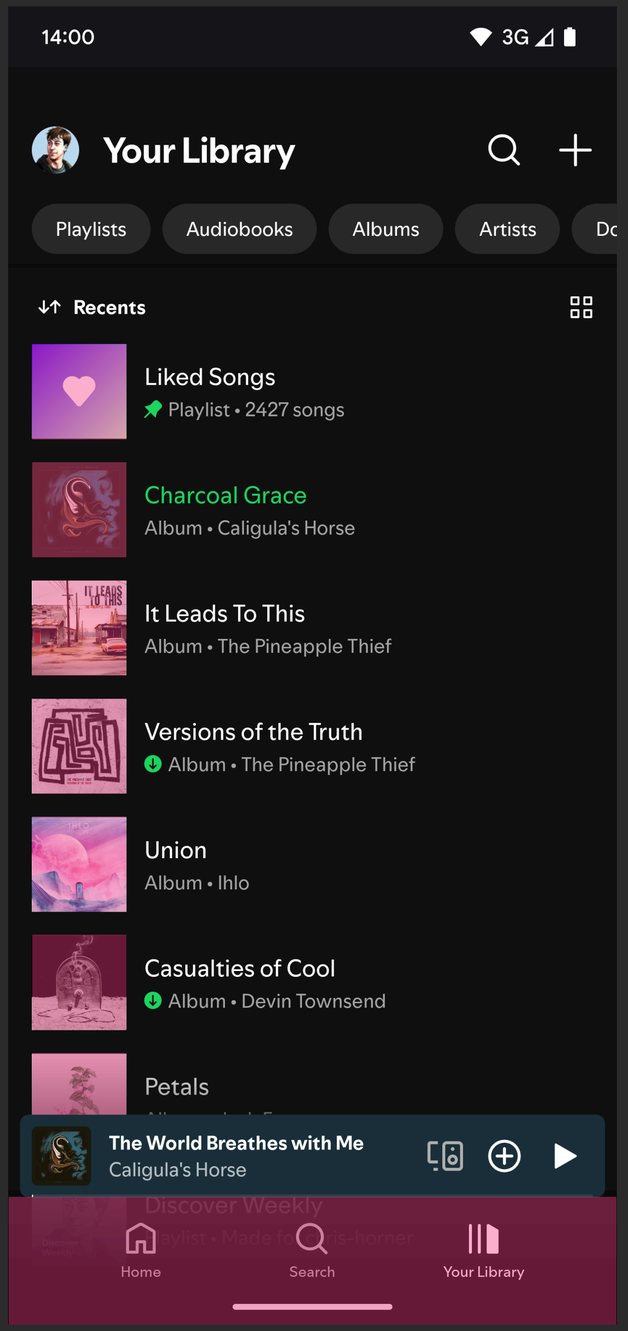
私は、四角形を作成しました。

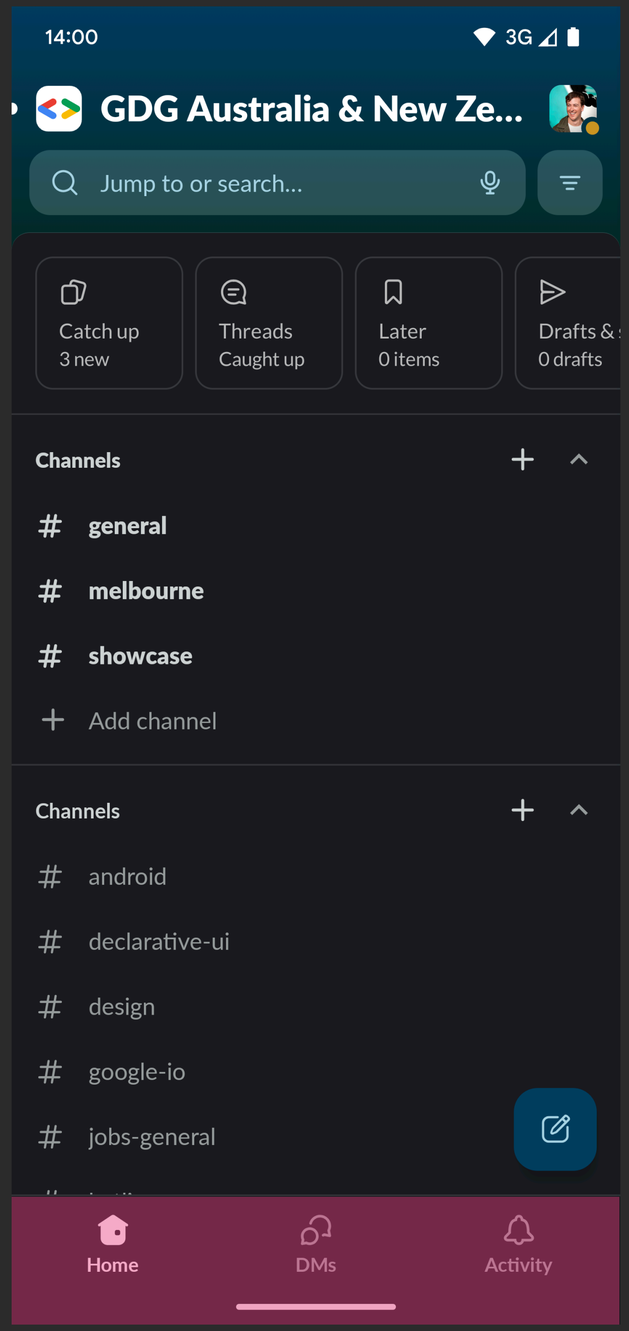
- A designer I worked with
- デザイナー友達

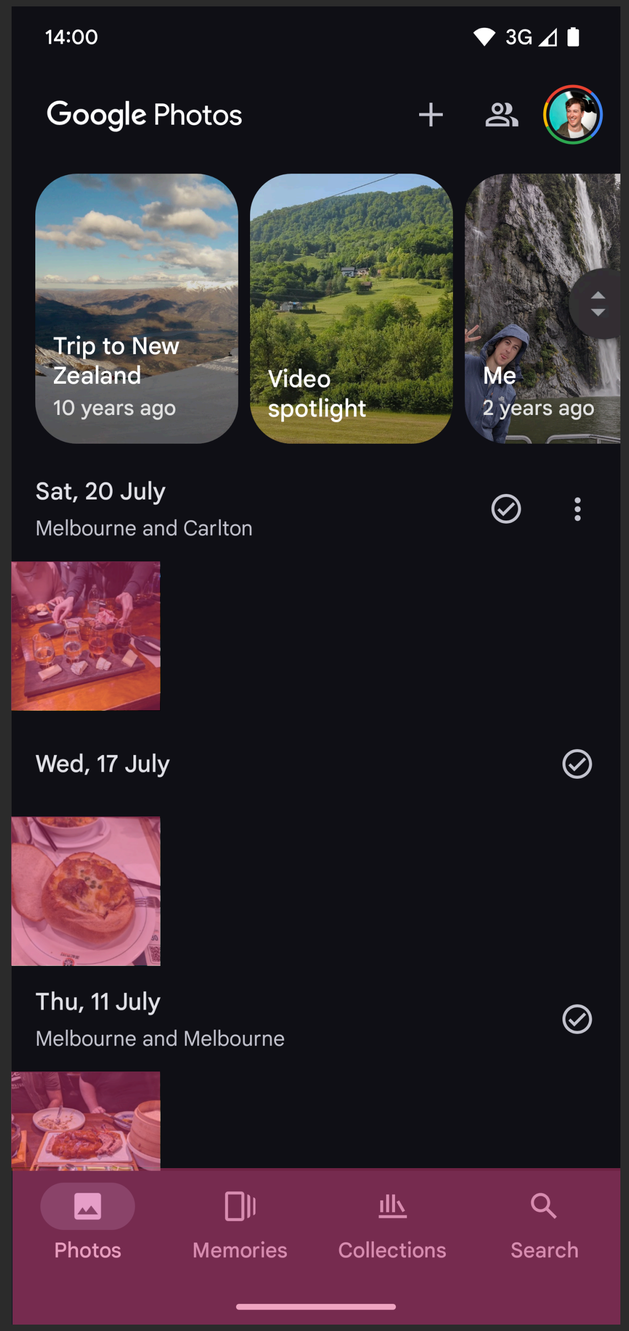
I arrange rectangles. Sometimes, if I want to get fancy, I round the corners.

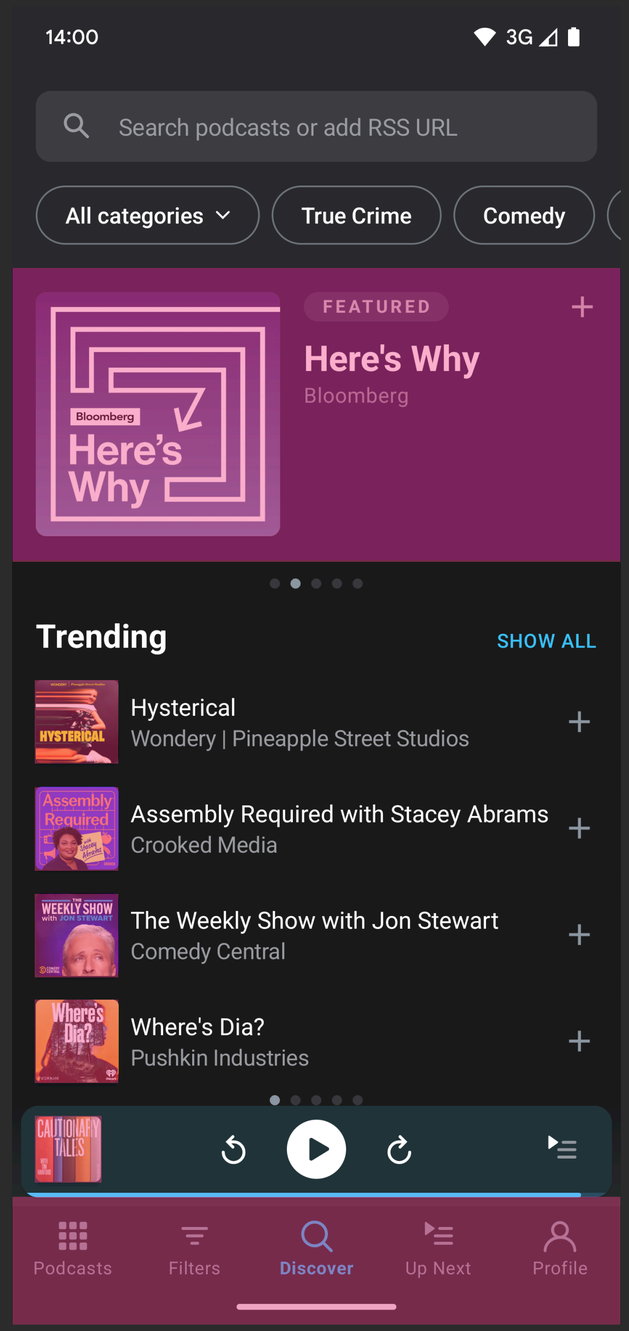
私は、四角形を作成しました。
角を丸めて、おしゃれにしたいです。

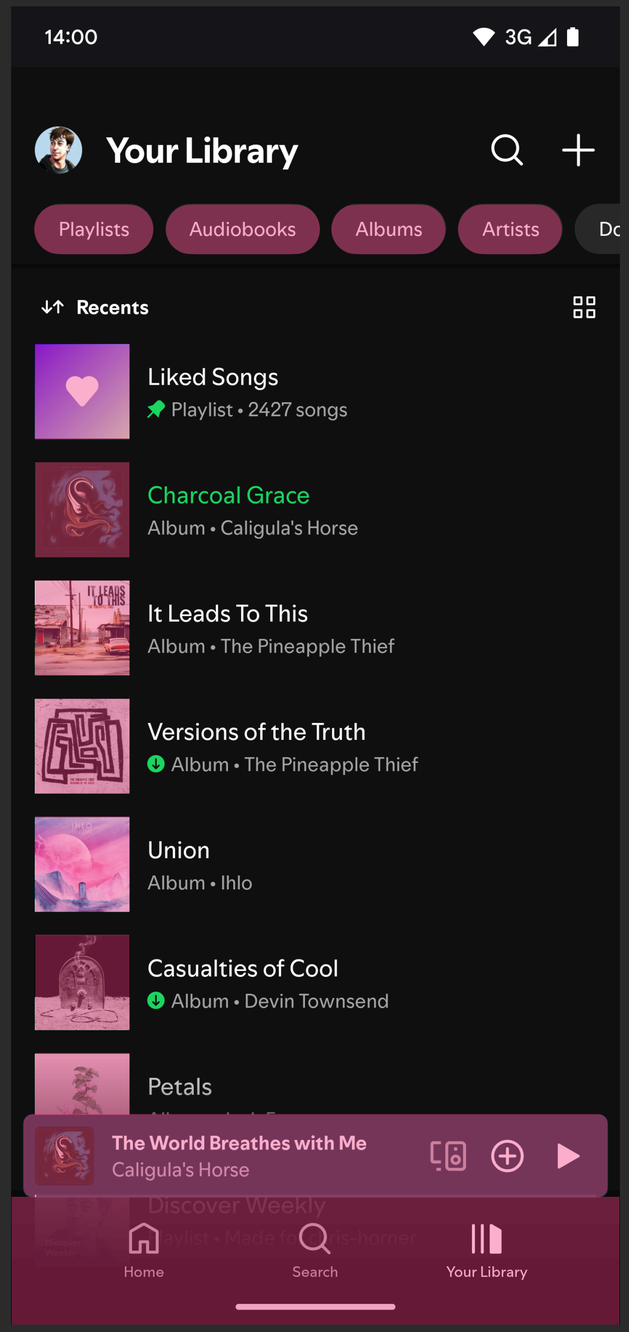
- A designer I worked with
- デザイナー友達

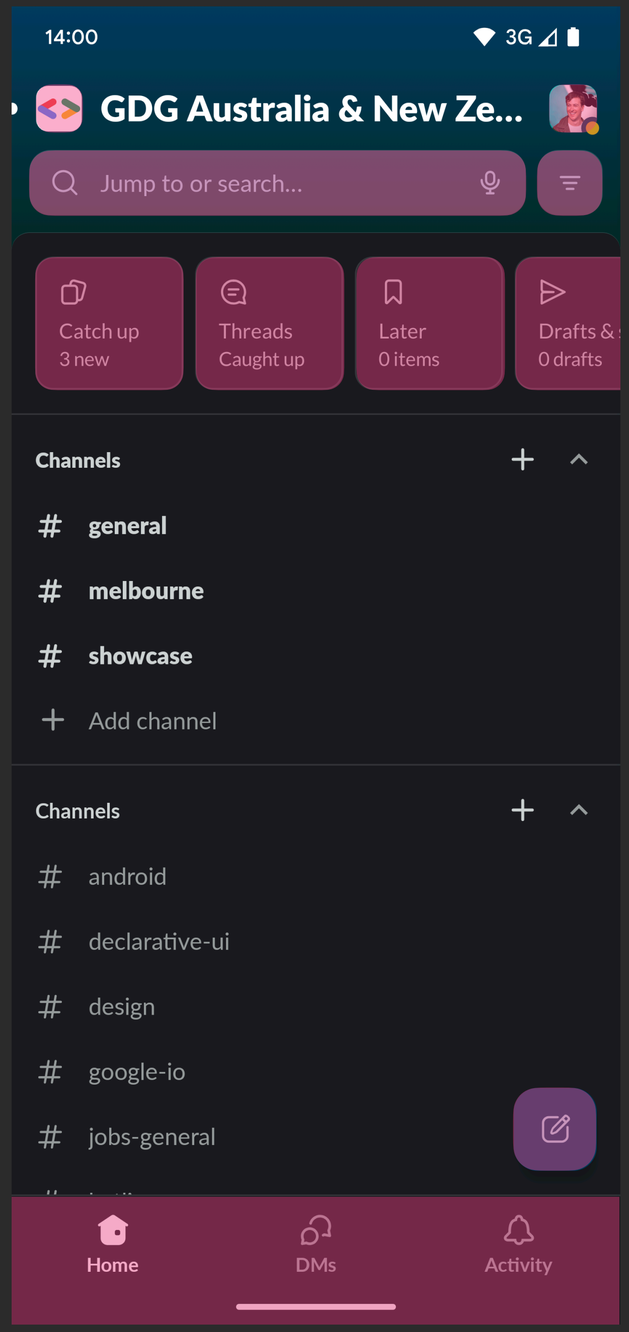

14:00
3G
Your Library
Playlists
Audiobooks
Albums
Artists
Recents
Liked Songs
Playlist • 2427 songs
Charcoal Grace
Album • Caligula's Horse
It Leads To This
Album • The Pineapple Thief
Versions of the Truth
Album • The Pineapple Thief
Union
Album • Ihlo
Casualties of Cool
Album • Devin Townsend
Petals
The World Breathes with Me
Caligula's Horse
Discover Weekly
Home
Search
Your Library

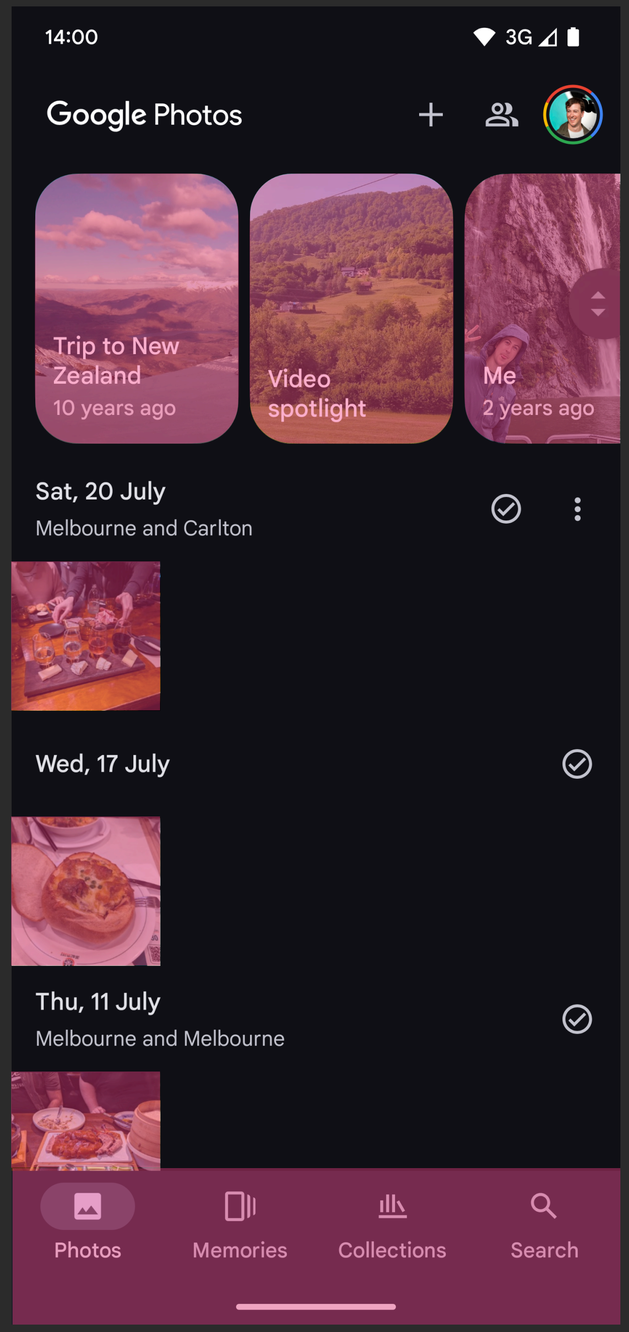

14:00
3G
GDG Australia & New Ze...
Jump to or search...
Catch up
3 new
Threads
Caught up
Later
0 items
Drafts &
0 drafts
Channels
general
melbourne
showcase
Add channel
Channels
android
declarative-ui
design
google-io
jobs-general
Home
DMs
Activity

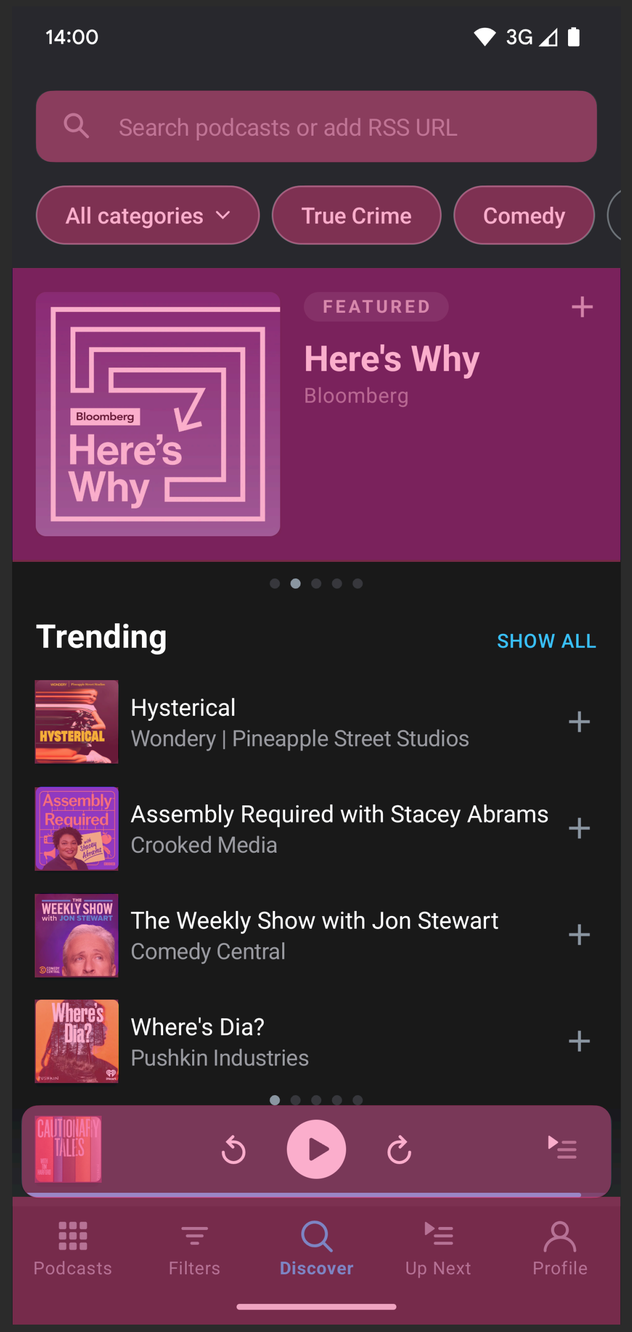

14:00
3G
Google Photos
Trip to New Zealand
10 years ago
Video spotlight
Me
2 years ago
Sat, 20 July
Melbourne and Carlton
Wed, 17 July
Thu, 11 July
Melbourne and Melbourne
Photos
Memories
Collections
Search

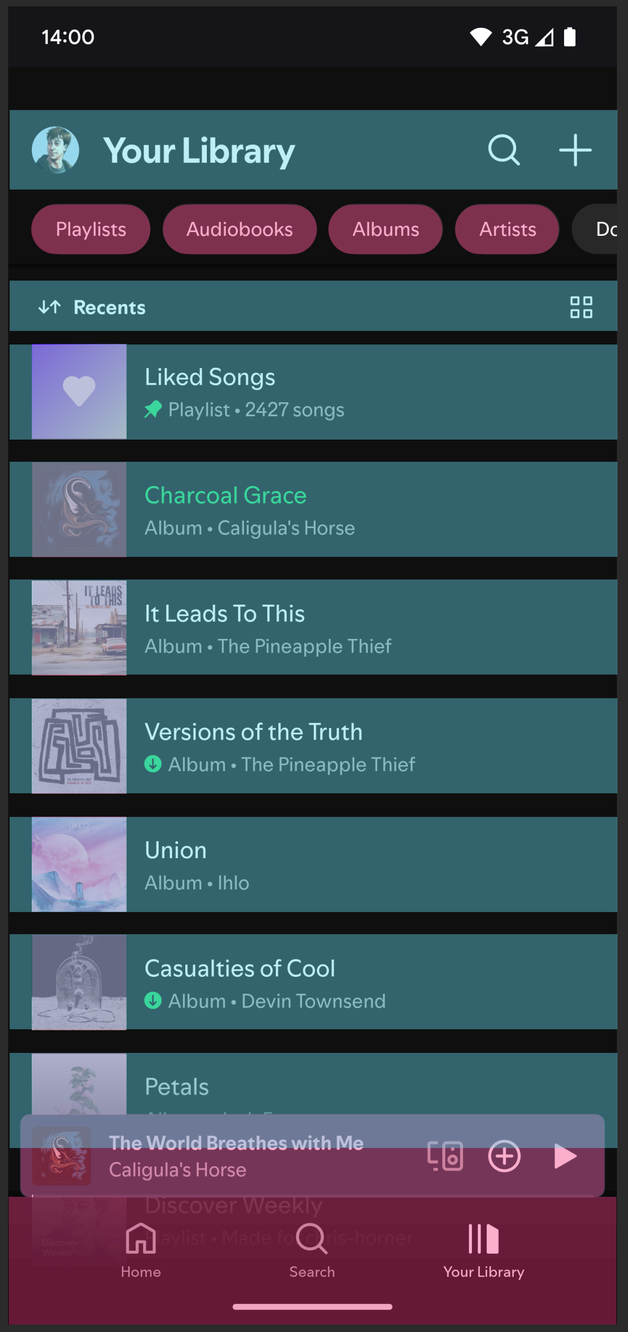

14:00
3G
Search podcasts or add RSS URL
All categories
True Crime
Comedy
FEATURED
Here's Why
Bloomberg
Trending
SHOW ALL
Hysterical
Wondery | Pineapple Street Studios
Assembly Required with Stacey Abrams
Crooked Media
The Weekly Show with Jon Stewart
Comedy Central
Where's Dia?
Pushkin Industries
Podcasts
Filters
Discover
Up Next
Profile

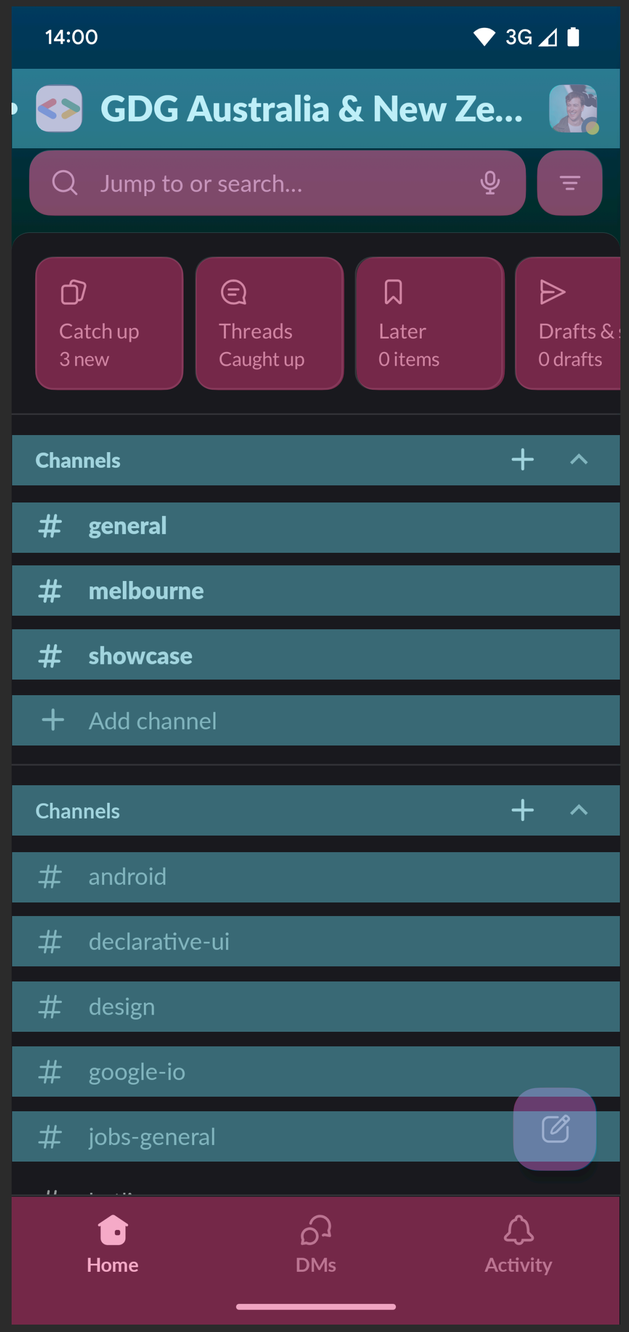

14:00
3G
Your Library
Playlists
Audiobooks
Albums
Artists
Recents
Liked Songs
Playlist • 2427 songs
Charcoal Grace
Album • Caligula's Horse
It Leads To This
Album • The Pineapple Thief
Versions of the Truth
Album • The Pineapple Thief
Union
Album • Ihlo
Casualties of Cool
Album • Devin Townsend
Petals
The World Breathes with Me
Caligula's Horse
Discover Weekly
Home
Search
Your Library

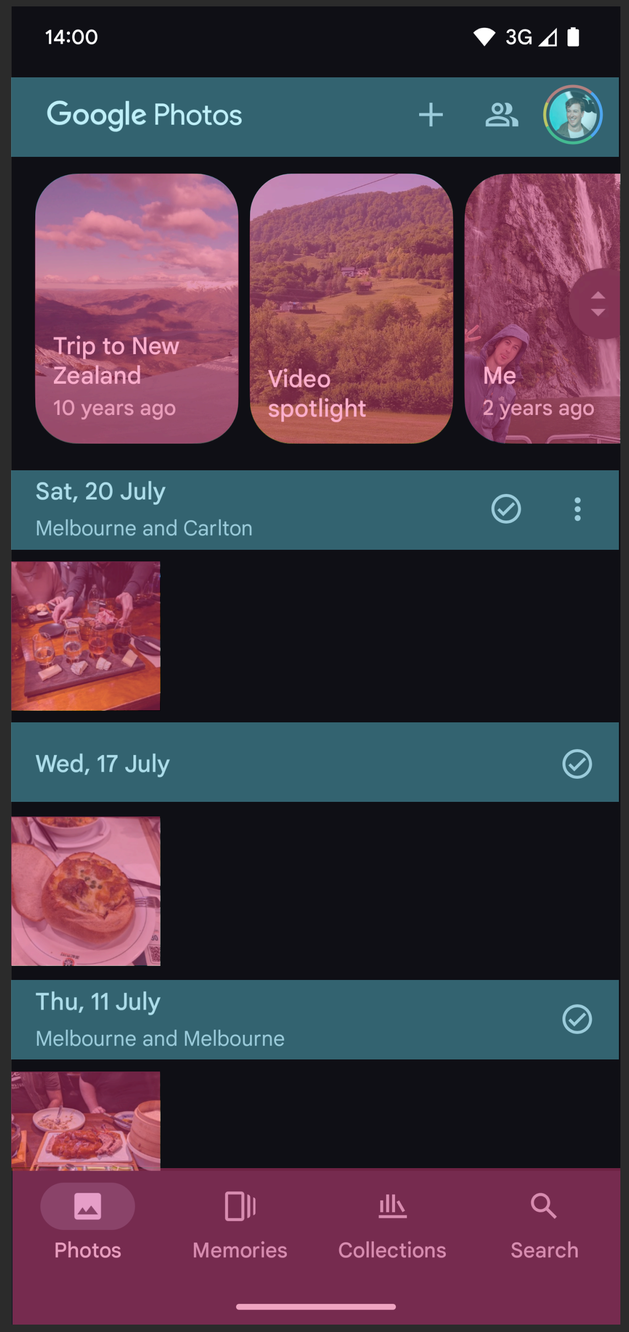

14:00
3G
GDG Australia & New Ze...
Jump to or search...
Catch up
3 new
Threads
Caught up
Later
0 items
Drafts &
0 drafts
Channels
general
melbourne
showcase
Add channel
Channels
android
declarative-ui
design
google-io
jobs-general
Home
DMs
Activity

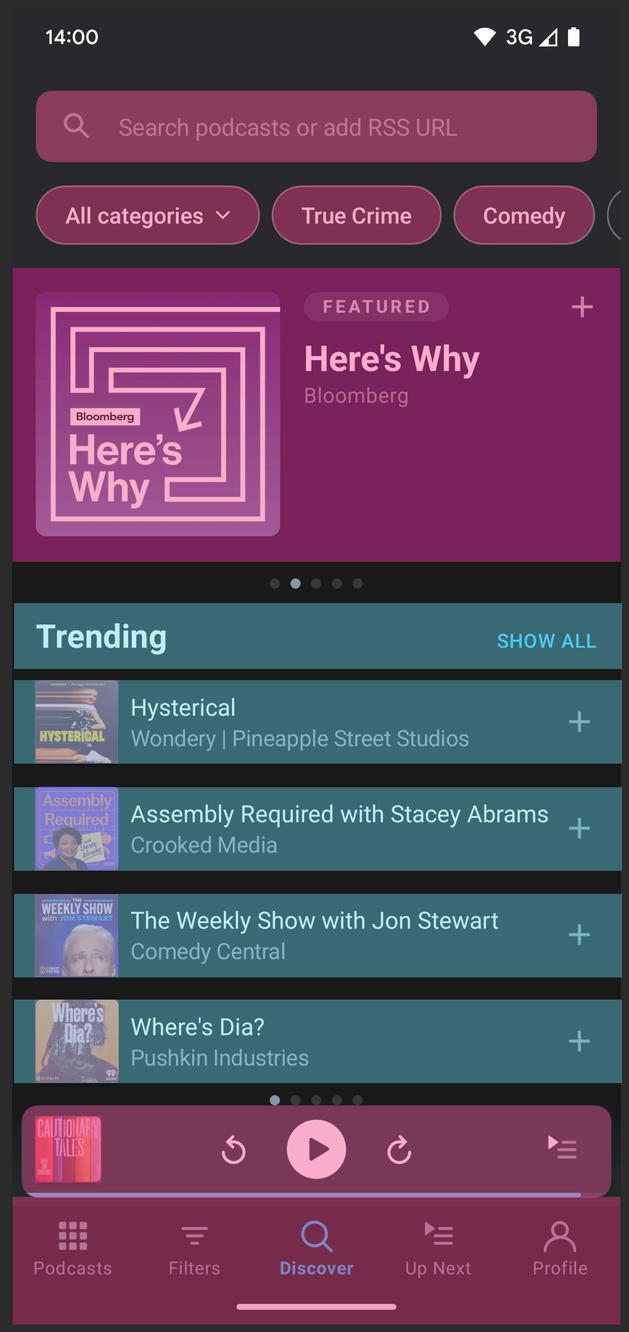

14:00
3G
Google Photos
Trip to New Zealand
10 years ago
Video spotlight
Me
2 years ago
Sat, 20 July
Melbourne and Carlton
Wed, 17 July
Thu, 11 July
Melbourne and Melbourne
Photos
Memories
Collections
Search


14:00
3G
Search podcasts or add RSS URL
All categories
True Crime
Comedy
FEATURED
Here's Why
Bloomberg
Trending
SHOW ALL
Hysterical
Wondery | Pineapple Street Studios
Assembly Required with Stacey Abrams
Crooked Media
The Weekly Show with Jon Stewart
Comedy Central
Where's Dia?
Pushkin Industries
Podcasts
Filters
Discover
Up Next
Profile


14:00
3G
Your Library
Playlists
Audiobooks
Albums
Artists
Recents
Liked Songs
Playlist • 2427 songs
Charcoal Grace
Album • Caligula's Horse
It Leads To This
Album • The Pineapple Thief
Versions of the Truth
Album • The Pineapple Thief
Union
Album • Ihlo
Casualties of Cool
Album • Devin Townsend
Petals
The World Breathes with Me
Caligula's Horse
Discover Weekly
Home
Search
Your Library

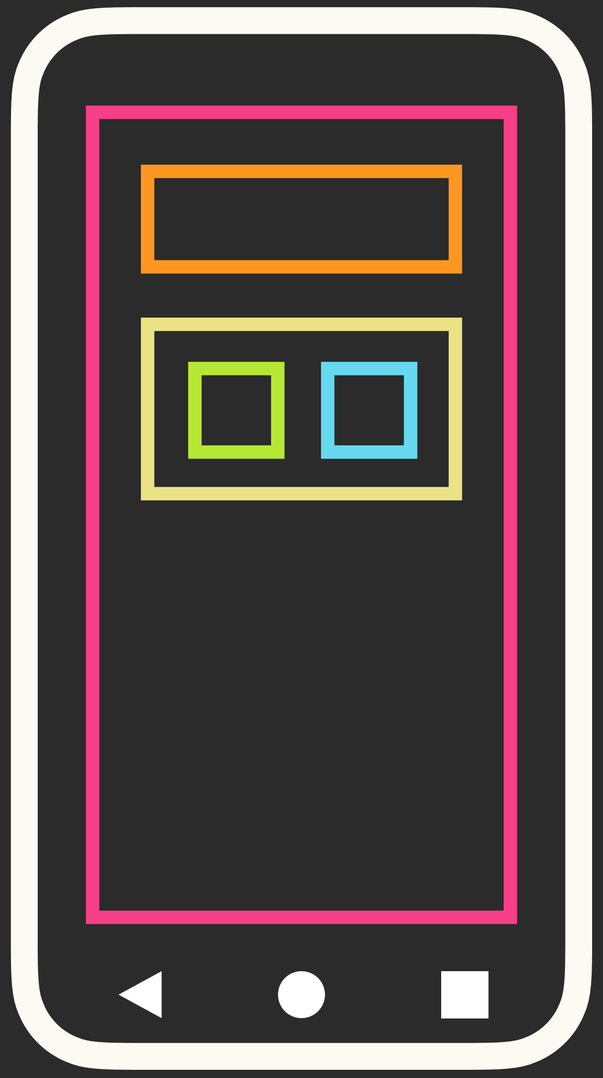

14:00
3G
GDG Australia & New Ze...
Jump to or search...
Catch up
3 new
Threads
Caught up
Later
0 items
Drafts &
0 drafts
Channels
general
melbourne
showcase
Add channel
Channels
android
declarative-ui
design
google-io
jobs-general
Home
DMs
Activity


14:00
3G
Google Photos
Trip to New Zealand
10 years ago
Video spotlight
Me
2 years ago
Sat, 20 July
Melbourne and Carlton
Wed, 17 July
Thu, 11 July
Melbourne and Melbourne
Photos
Memories
Collections
Search


14:00
3G
Search podcasts or add RSS URL
All categories
True Crime
Comedy
FEATURED
Here's Why
Bloomberg
Trending
SHOW ALL
Hysterical
Wondery | Pineapple Street Studios
Assembly Required with Stacey Abrams
Crooked Media
The Weekly Show with Jon Stewart
Comedy Central
Where's Dia?
Pushkin Industries
Podcasts
Filters
Discover
Up Next
Profile

UI frameworks are built around rectangles

UI のフレームワークは四角形から構築されています。



```
Column {
   Box()

   Row {
      Box()
      Box()
   }
}
```

Flat design
フラットデザイン

Easy for a designer to put together
デザイナーにとっては簡単に全てを組み合わせられる

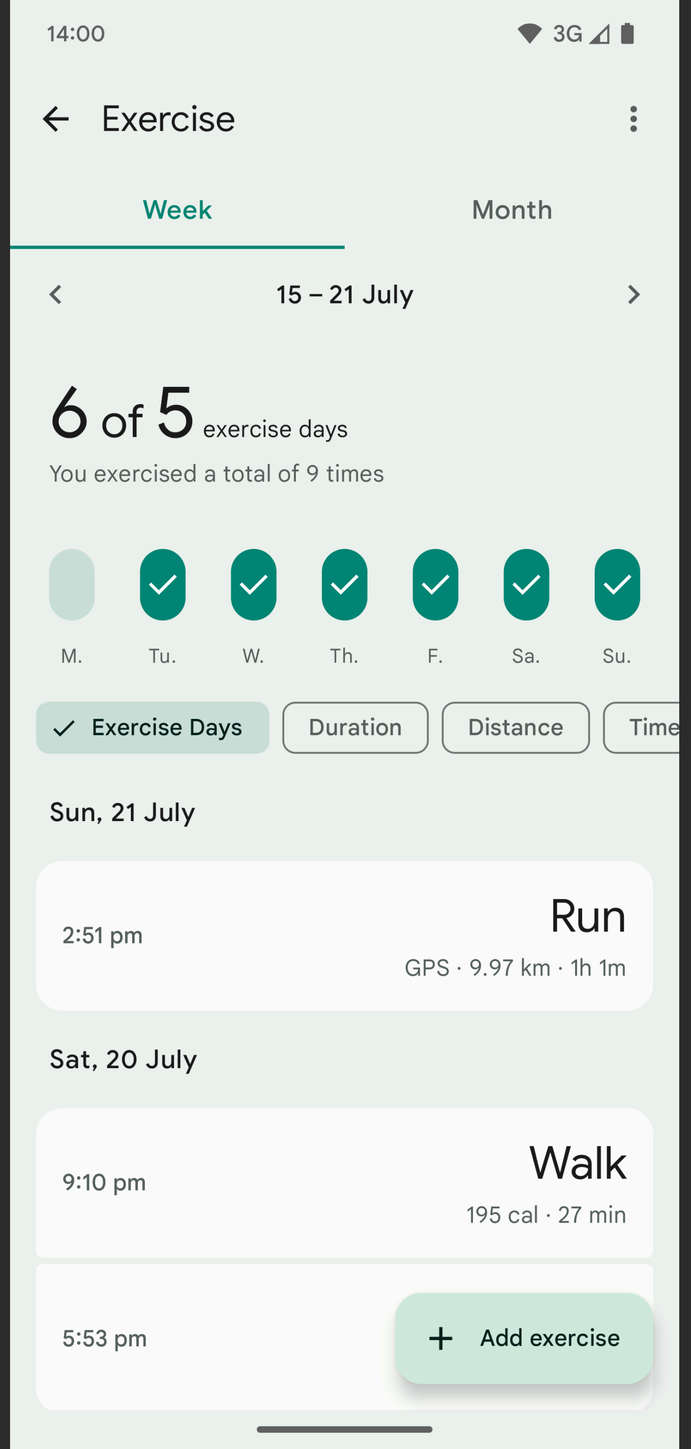
Simple for a developer to build
開発者にとって構築が簡単なことになる

Low effort to read
読むのに手間がかからない

Renders quickly
素早く描き上げる

Responsive
反応がいい

Dark mode
ダークモード



Boring 😴

is good

平凡ないいことですね。

Boring 😴 平凡ないいことですね。
is good

Phone calls
電話

Customer support
カスタマーサービス

Maps / navigation
地図／ナビ

Email
メール

But... have we gone too far?
しかし...行き過ぎたのだろうか？

Let’s go back in time...

過去に戻りましょう

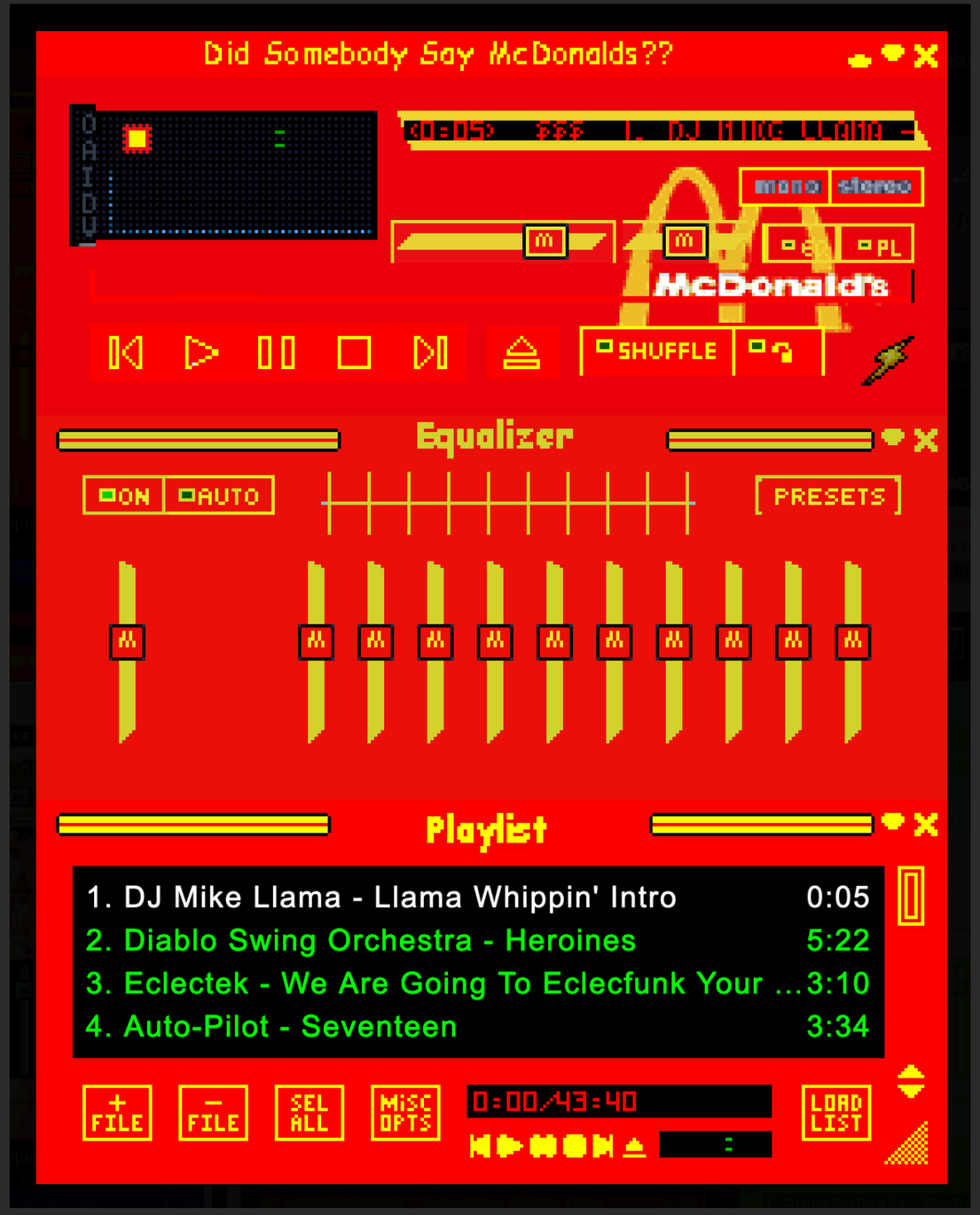
2009年


htc
10:08 AM
Wed, Jun 24
menu

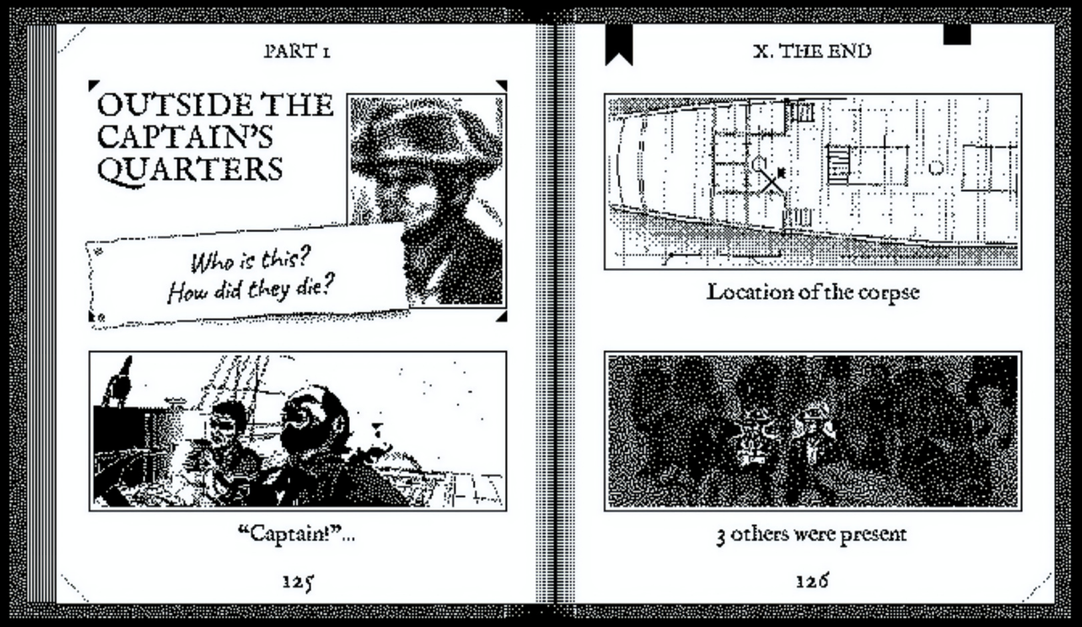

WINAMP
00:04
SUMMER REMIX> <5:14> *** 9.
192 kbps 44 kHz
mono stereo
EQ PL
SHUFFLE
WINAMP EQUALIZER
ON AUTO
PRESETS
+12 db
+0 db
-12 db
PREAMP
60 170 310 600 1K 3K 6K 12K 14K 16K
WINAMP PLAYLIST
1. DJ Mike Llama - Llama Whippin' Intro 0:05
2. Diablo Swing Orchestra - Heroines 5:22
3. Eclectek - We Are Going To Eclecfunk Your ... 3:10
4. Auto-Pilot - Seventeen 3:34
5. Muha - Microphone 3:01
6. Just Plain Ant - Stumble 1:26
7. Sleaze - God Damn 3:46
8. Juanitos - Hola Hola Bossa Nova 3:27
9. Entertainment for the Braindead - Resolutio... 5:14
10. Nobara Hayakawa - Trail 3:24
11. Paper Navy - Tongue Tied 3:21
12. 60 Tigres - Garage 4:05
13. CM aka Creative - The Cycle (Featuring Mi... 3:41
ADD REM SEL MISC
5:14/43:40
00:04
LIST OPTS


WINAMP
00:00
DJ MIKE LLAMA - LLAMA WHIPPIN'
KBPS
KHZ
MONO STEREO
EQ PL
EQUALIZER
ON AUTO
PRESETS
+12 db
+0 db
-12 db
PREAMP
60 170 310 600 1K 3K 6K 12K 14K 16K
PLAYLIST
1. DJ Mike Llama - Llama Whippin' Intro 0:05
2. Diablo Swing Orchestra - Heroines 5:22
3. Eclectek - We Are Going To Eclecfunk Your ... 3:10
4. Auto-Pilot - Seventeen 3:34
0:00/43:40


CREATED BY ELYSE
STAR WARS
EPISODE II
IN' INTRO (0:05) *** 1. DJ MI
EQ
PL
ON
AUTO
PRESETS
OBI-WAN KENOBI
1. DJ Mike Llama - Llama Whippin' Intro 0:05
2. Diablo Swing Orchestra - Heroines 5:22
3. Eclectek - We Are Going To Eclecfunk Y... 3:10
4. Auto-Pilot - Seventeen 3:34
+ FILE
- FILE
SEL ALL
MISC OPTS
0:00/43:40
LOAD LIST
USE THE FORCE

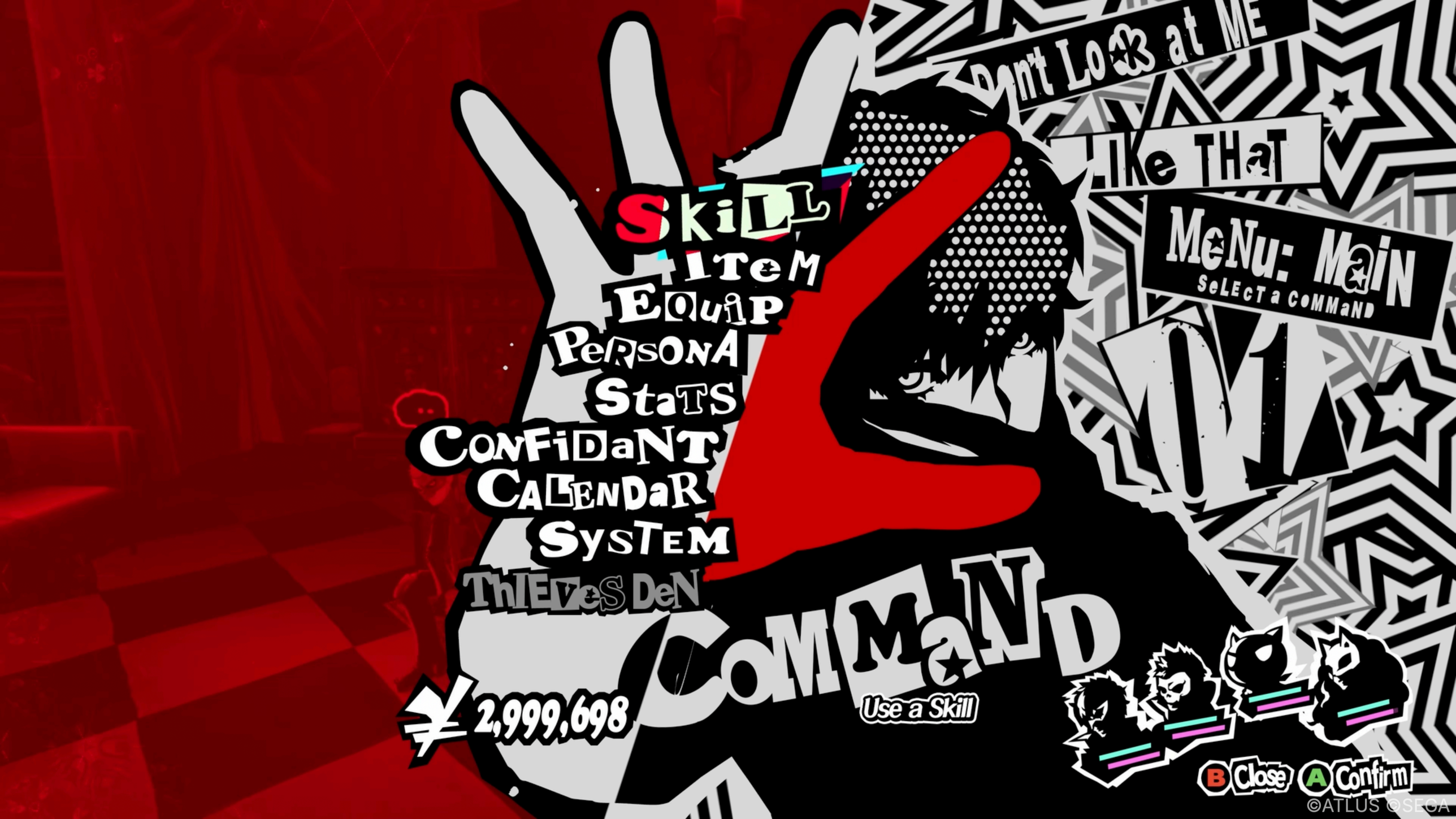

Did Somebody Say McDonalds??
mono
stereo
PL
McDonald's
SHUFFLE
Equalizer
ON
AUTO
PRESETS
Playlist
1. DJ Mike Llama - Llama Whippin' Intro 0:05
2. Diablo Swing Orchestra - Heroines 5:22
3. Eclectek - We Are Going To Eclecfunk Your ... 3:10
4. Auto-Pilot - Seventeen 3:34
+ FILE
- FILE
SEL ALL
MISC OPTS
0:00/43:40
LOAD LIST

skins.webamp.org




PERSONA5 ROYAL
PRESS ANY BUTTON
ATLUS
© ATLUS. © SEGA. All rights reserved.
©ATLUS ©SEGA

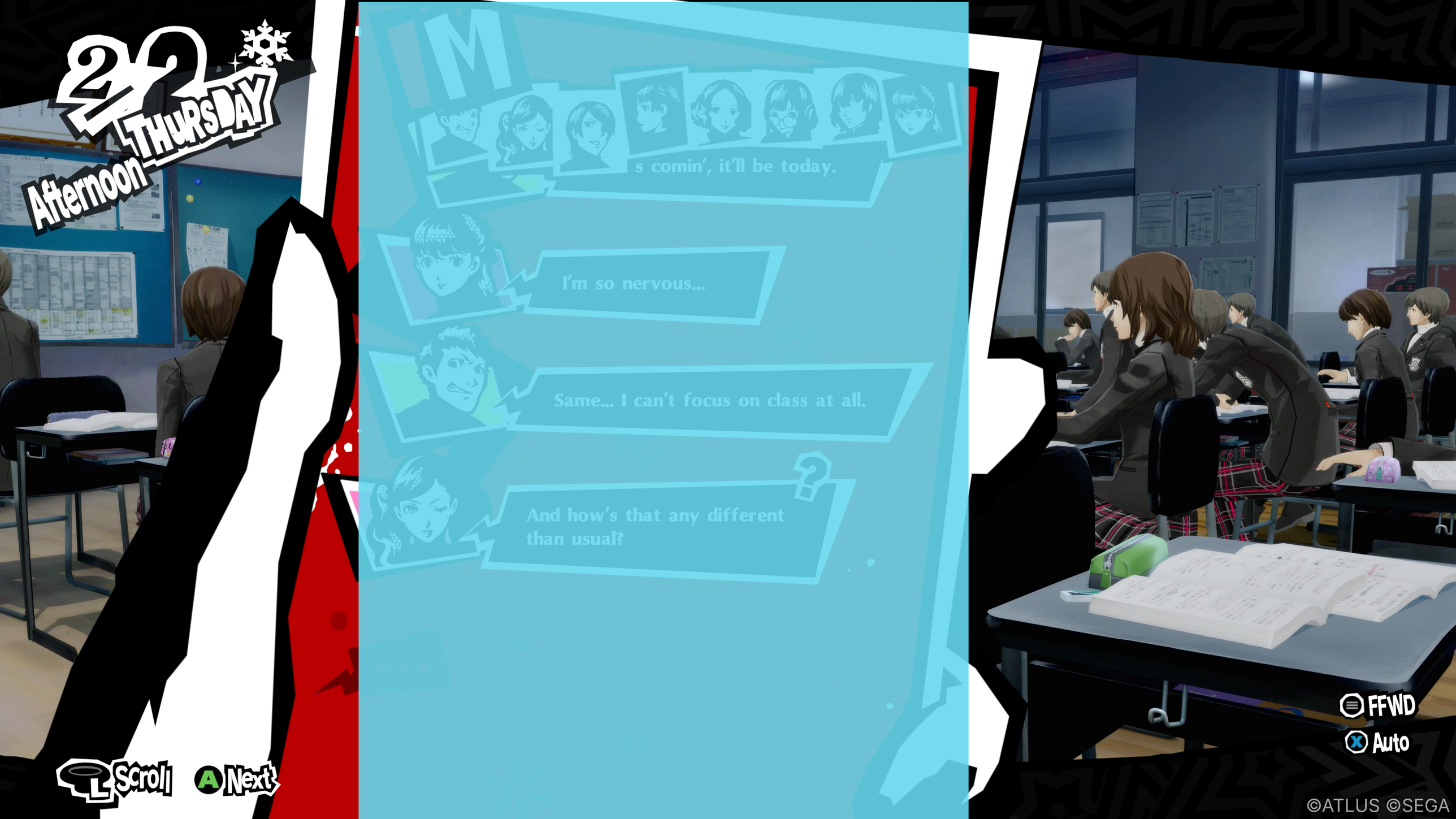

Oni
LV 20
Don't bother with these guys!
Just finish the job!
ORDER
Change Tactics
LT
GUN
Take Aim
PERSONA
Use a Skill
ITEM
Use an Item
GUARD
Defend Yourself
ATTACK
Use Melee Weapon
HP 308 / SP 165
295 122
272 128
275 162
LB Analyze
Target
Intel
Rush
RB Assist
©ATLUS ©SEGA


Don't Look at ME
Like THaT
MeNu: MaiN
SeLECT a coMMaND
SkiLL
ITeM
EQuiP
PeRSONA
StaTS
CONFiDaNT
CALENDaR
SYSTEM
ThIEVeS DeN
COMMaND
¥ 2,999,698
Use a Skill
B Close
A Confirm
©ATLUS ©SEGA

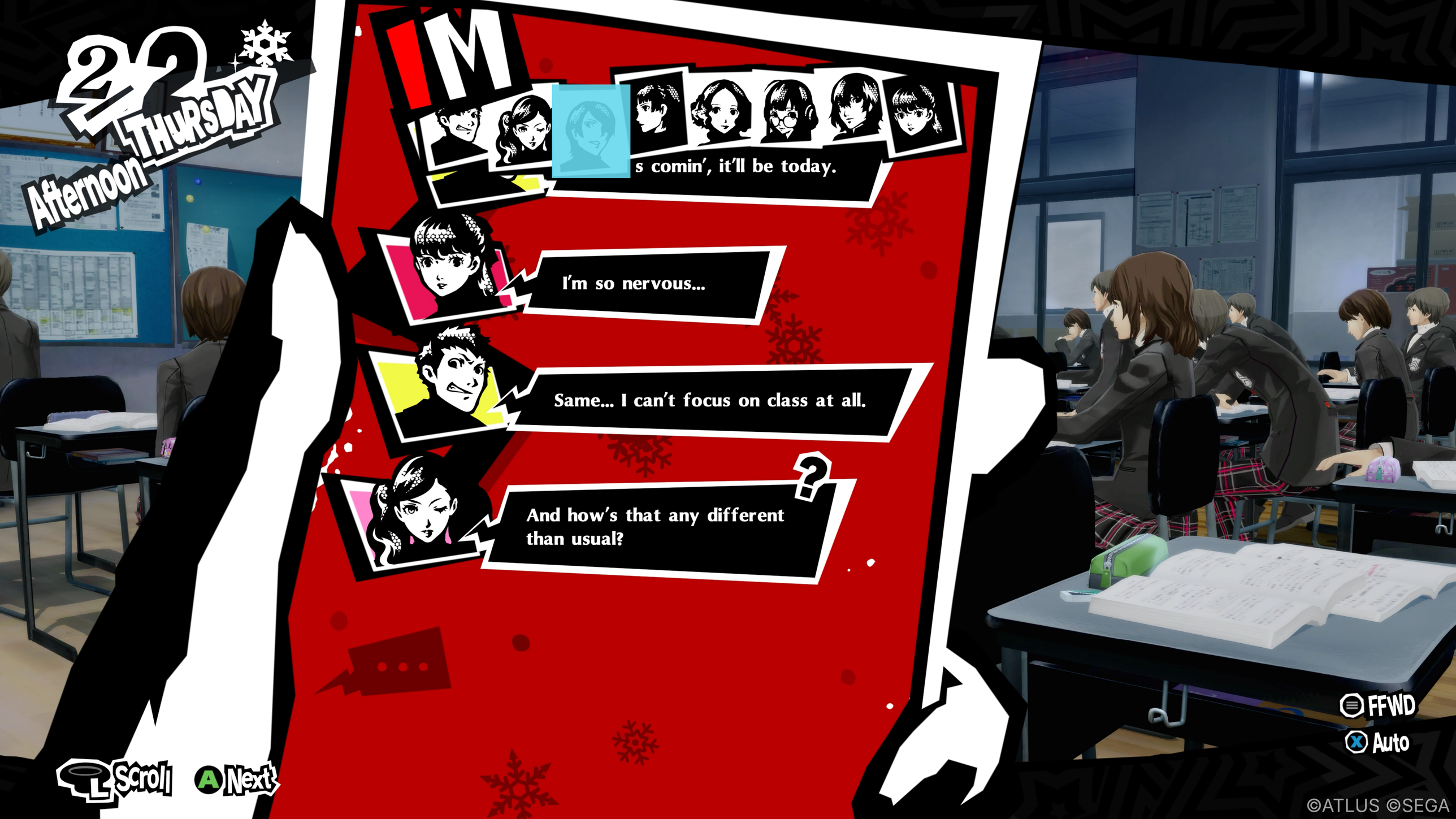

22
THURSDAY
Afternoon
s comin', it'll be today.
I'm so nervous...
Same... I can't focus on class at all.
And how's that any different than usual?
?
L Scroll
A Next
FFWD
Auto
©ATLUS ©SEGA

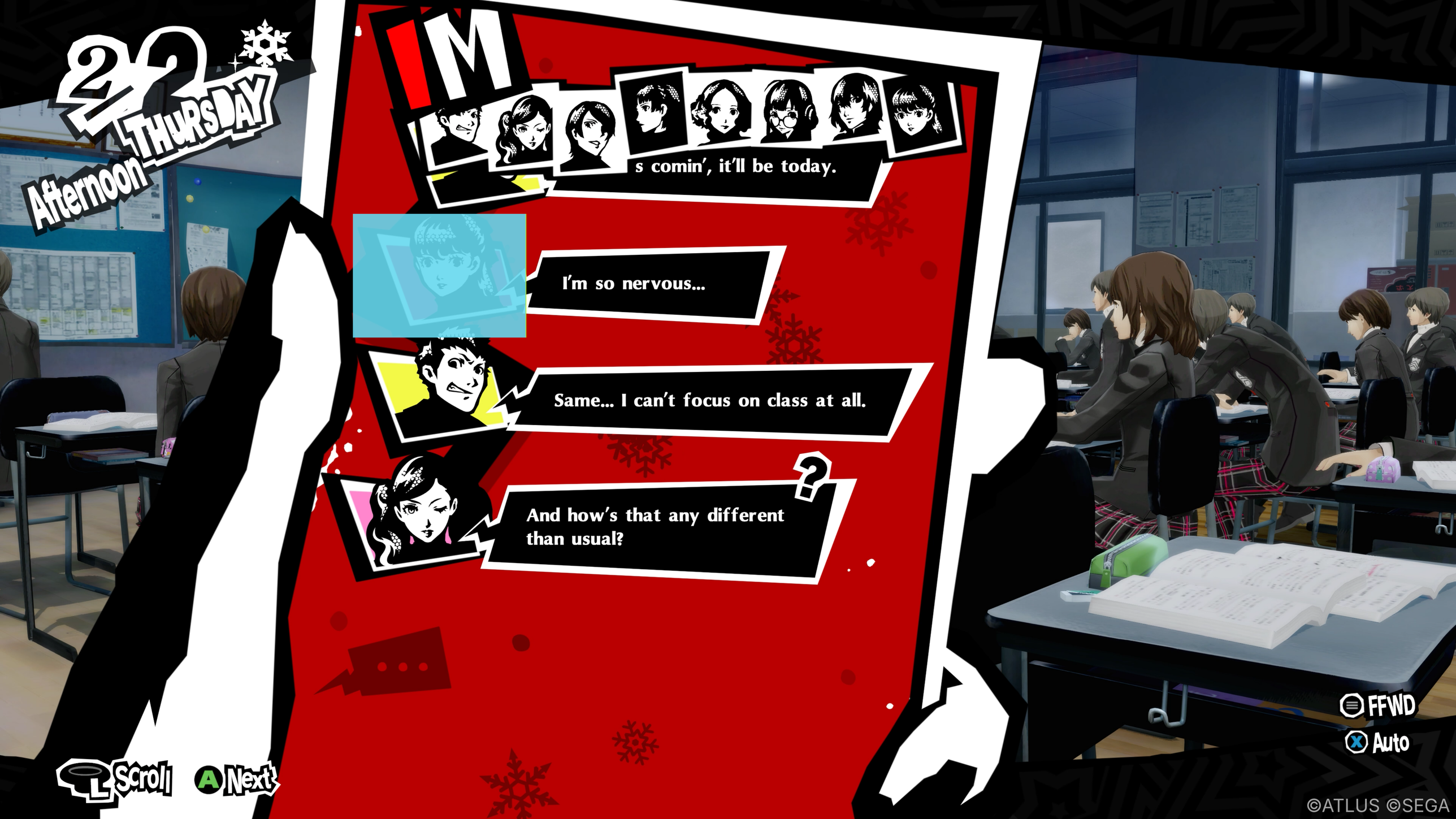
How could we build this with Compose UI?

どのようにUI構築をしていけばいいのでしょうか？


22
THURSDAY
Afternoon
s comin', it'll be today.
I'm so nervous...
Same... I can't focus on class at all.
And how's that any different than usual?
L Scroll
A Next
FFWD
Auto
©ATLUS ©SEGA


22
THURSDAY
Afternoon
s comin', it'll be today.
I'm so nervous...
Same... I can't focus on class at all.
And how's that any different than usual?
FFWD
Auto
L Scroll
A Next
©ATLUS ©SEGA


22
THURSDAY
Afternoon
s comin', it'll be today.
I'm so nervous...
Same... I can't focus on class at all.
And how's that any different than usual?
?
L Scroll
A Next
FFWD
Auto
©ATLUS ©SEGA

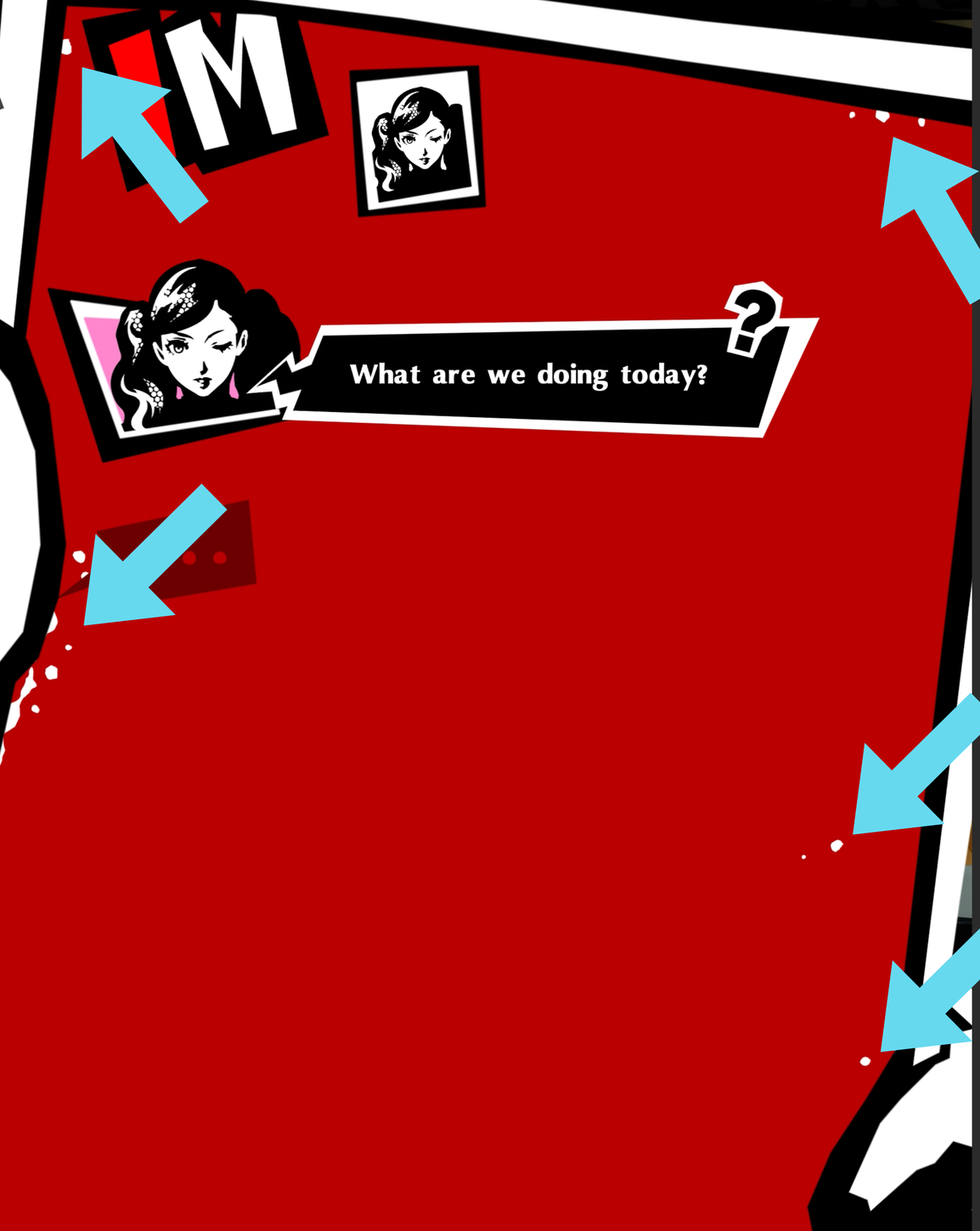

22
THURSDAY
Afternoon
s comin', it'll be today.
I'm so nervous...
Same... I can't focus on class at all.
And how's that any different than usual?
Scroll
Next
FFWD
Auto
©ATLUS ©SEGA


22
THURSDAY
Afternoon
s comin', it'll be today.
I'm so nervous...
Same... I can't focus on class at all.
And how's that any different than usual?
?
L Scroll
A Next
FFWD
Auto
©ATLUS ©SEGA

Background


2/1 WEDNESDAY
After School
3 IM
IM
What are we doing today?
Days until Maruki Acts: 2
Steal Maruki's heart
Prepare for meeting with Maruki
FFWD
Auto
Scroll
Next
©ATLUS ©SEGA


```
val PersonaRed = Color(0xFFC41001)

Box(
   modifier = Modifier
      .fillMaxSize()
      .background(color = PersonaRed)
) {

}
```




```
val PersonaRed = Color(0xFFC41001)

Box(
   modifier = Modifier
      .fillMaxSize()
      .background(color = PersonaRed)
) {

}
```


```
Box(
  modifier = Modifier
    .fillMaxSize()
    .background(color = PersonaRed)
) {
  Image(
    painter = painterResource(R.drawable.bg_splatter),
    contentDescription = null,
    contentScale = ContentScale.FillWidth,
    modifier = Modifier.statusBarsPadding()
  )
}
```

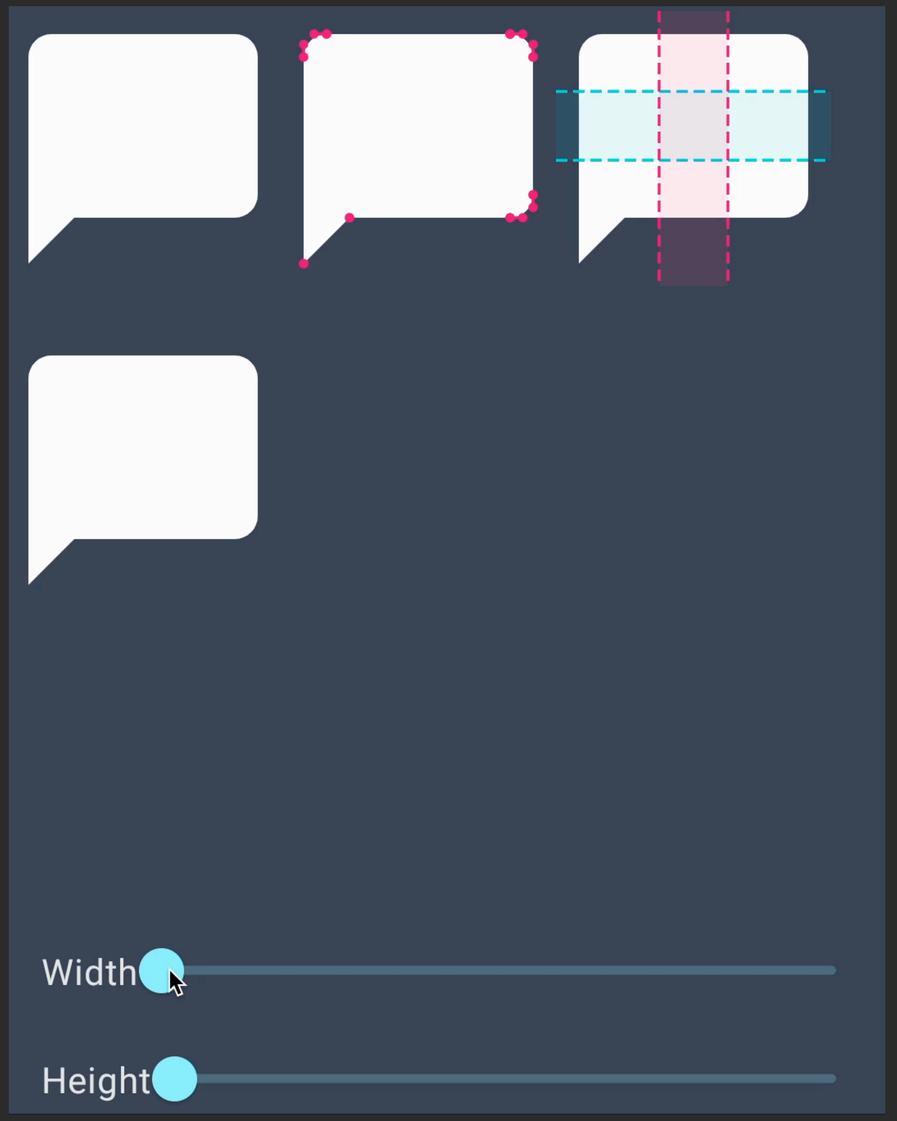

```
Box(
  modifier = Modifier
    .fillMaxSize()
    .background(color = PersonaRed)
) {
  Image(
    painter = painterResource(R.drawable.bg_splatter),
    contentDescription = null,
    contentScale = ContentScale.FillWidth,
    modifier = Modifier.statusBarsPadding()
  )

  Image(
    painter = painterResource(R.drawable.logo_im),
    contentDescription = null,
    modifier = Modifier
      .height(100.dp)
      .statusBarsPadding()
      .offset(x = 8.dp, y = (-4).dp),
  )
}
```


2/1 WEDNESDAY
After School
3
IM
IM
What are we doing today?
Days until
Maruki Acts: 2
Steal Maruki's heart
Prepare for meeting with Maruki
FFWD
Auto
Scroll
Next
©ATLUS ©SEGA

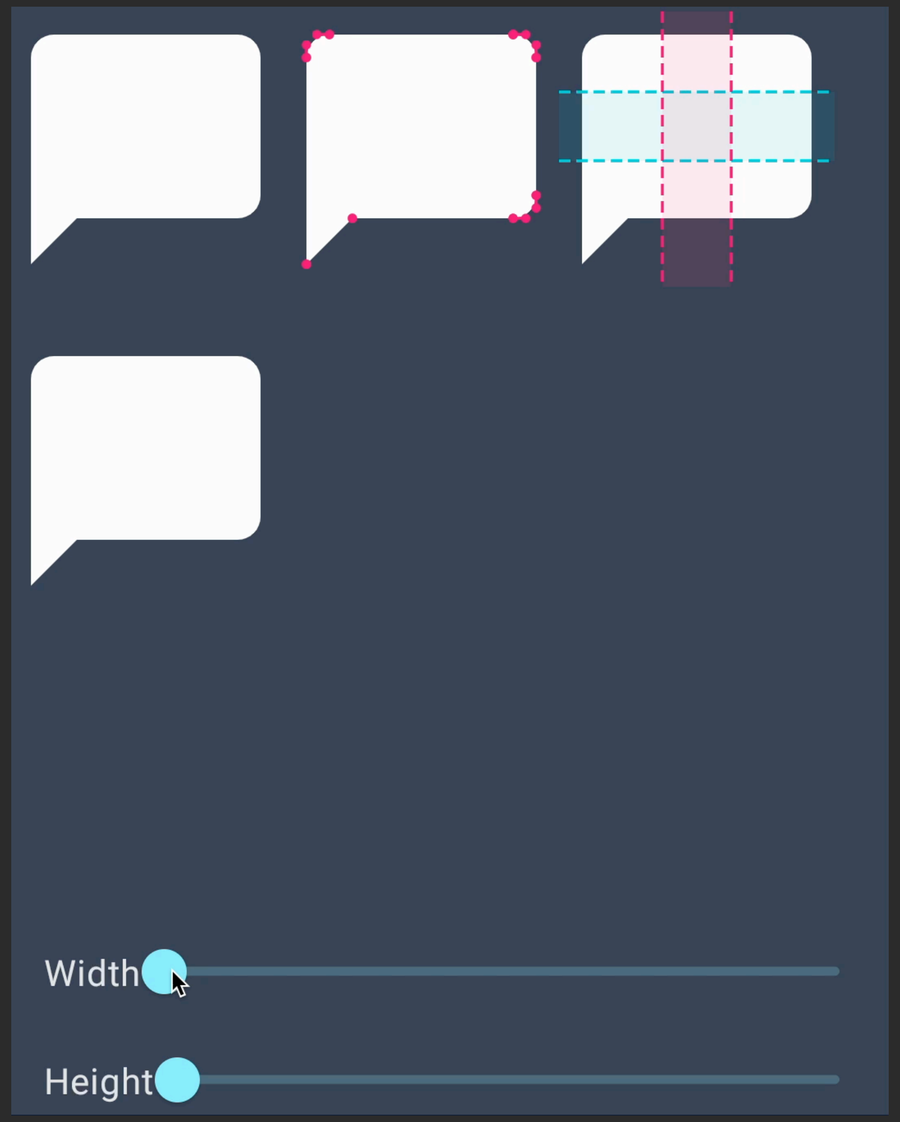

2/1 WEDNESDAY
After School
3 IM
IM
What are we doing today?
Sup?
Days until Maruki Acts: 2
Steal Maruki's heart
Prepare for meeting with Maruki
FFWD
Auto
Scroll
Next
©ATLUS ©SEGA


2/1 WEDNESDAY
After School
3
IM
IM
What are we doing today?
...
Sup?
Days until
Maruki Acts: 2
Steal Maruki's heart
Prepare for meeting with Maruki
FFWD
Auto
Scroll
Next
©ATLUS ©SEGA

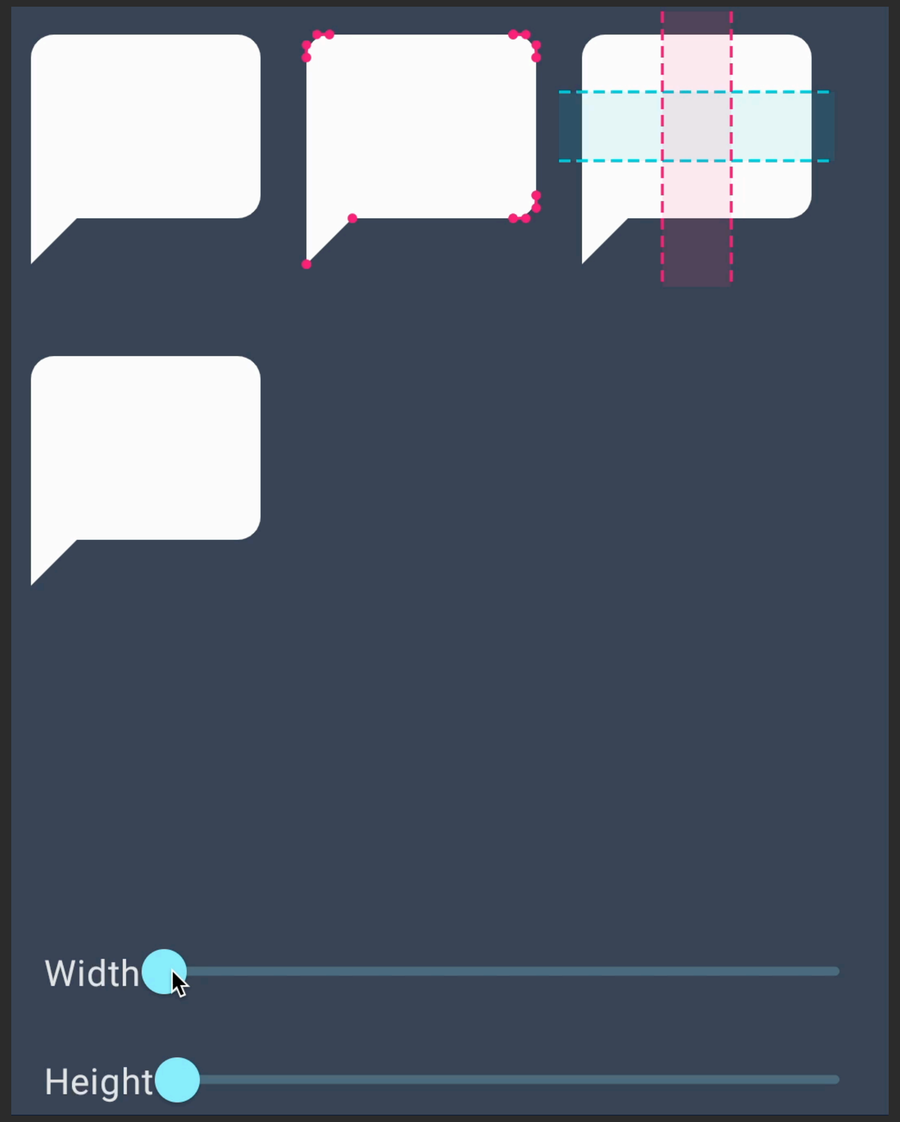
romainguy / v9



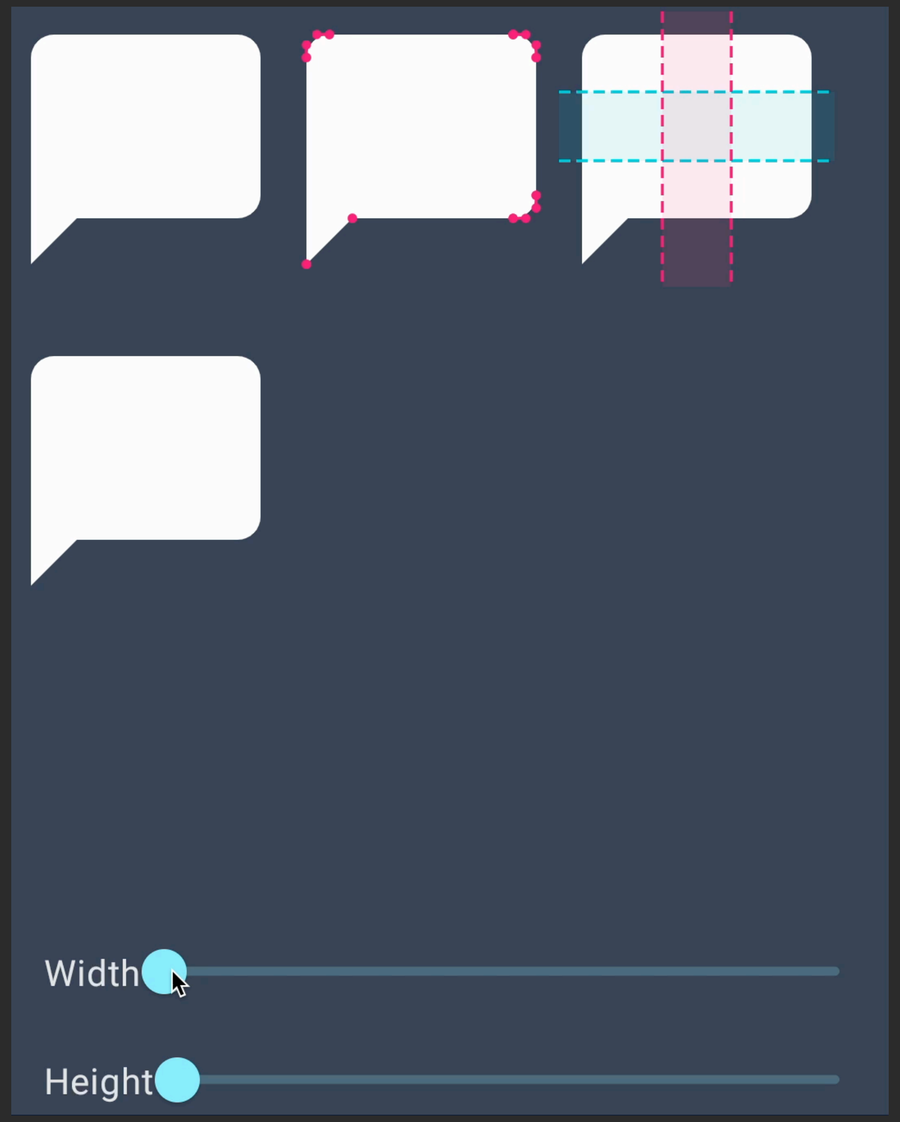
romainguy / v9

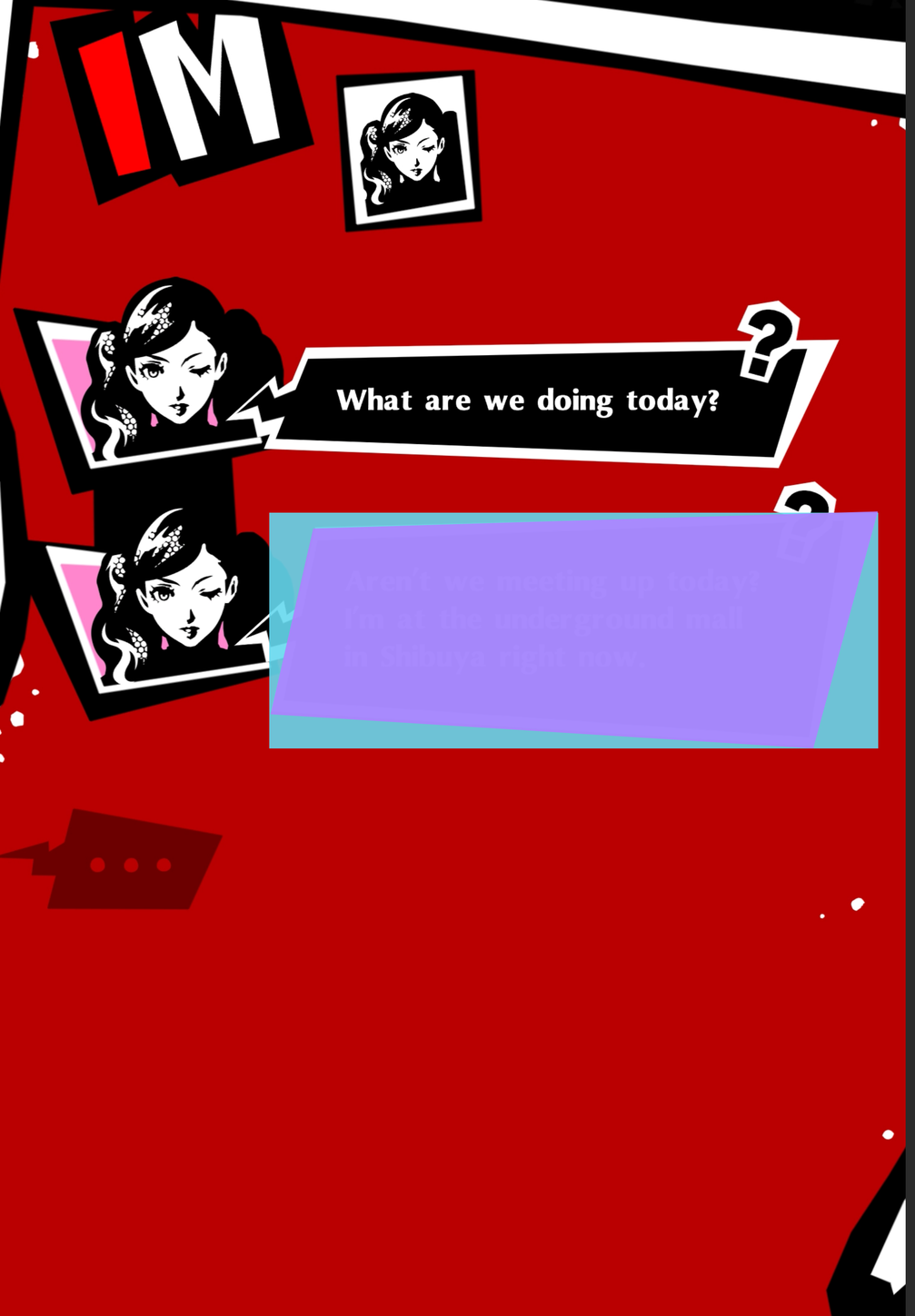


romainguy / v9

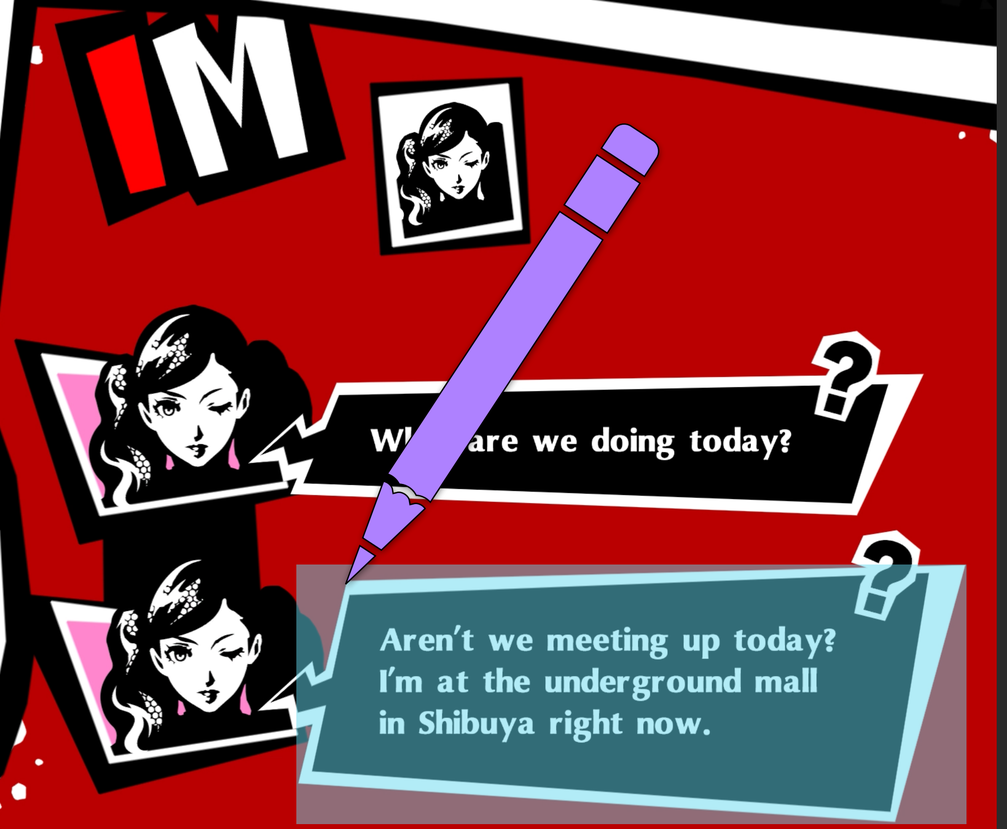


romainguy / v9

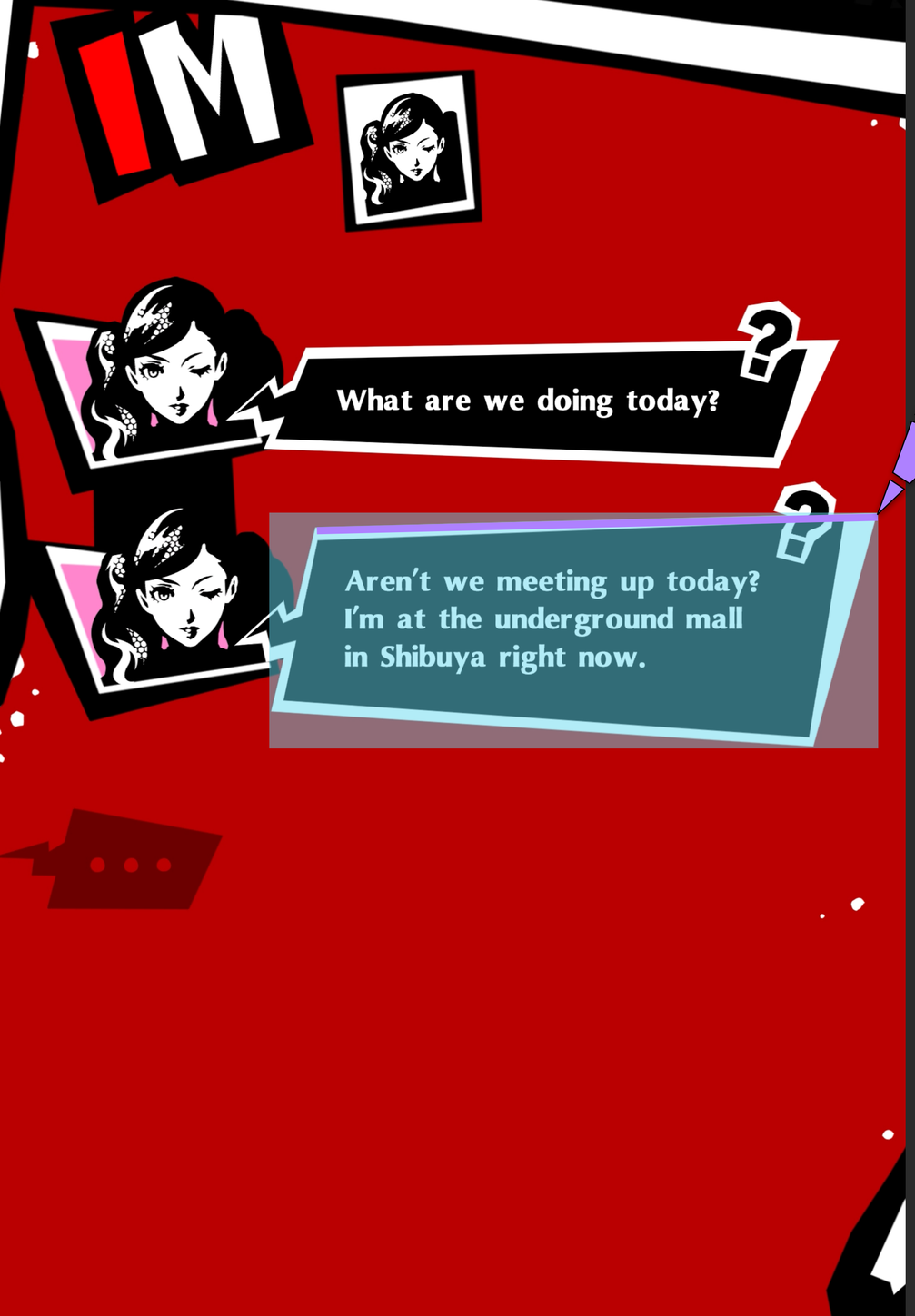




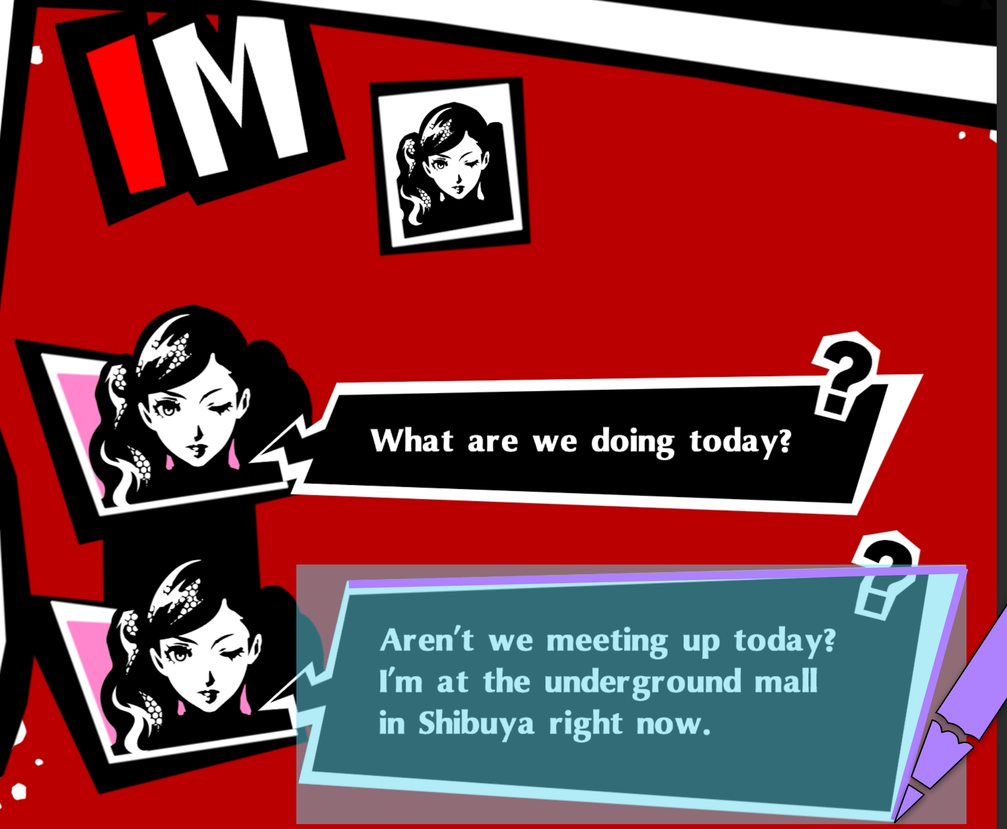

```
fun outerBox(): Shape = GenericShape { size →

}
```



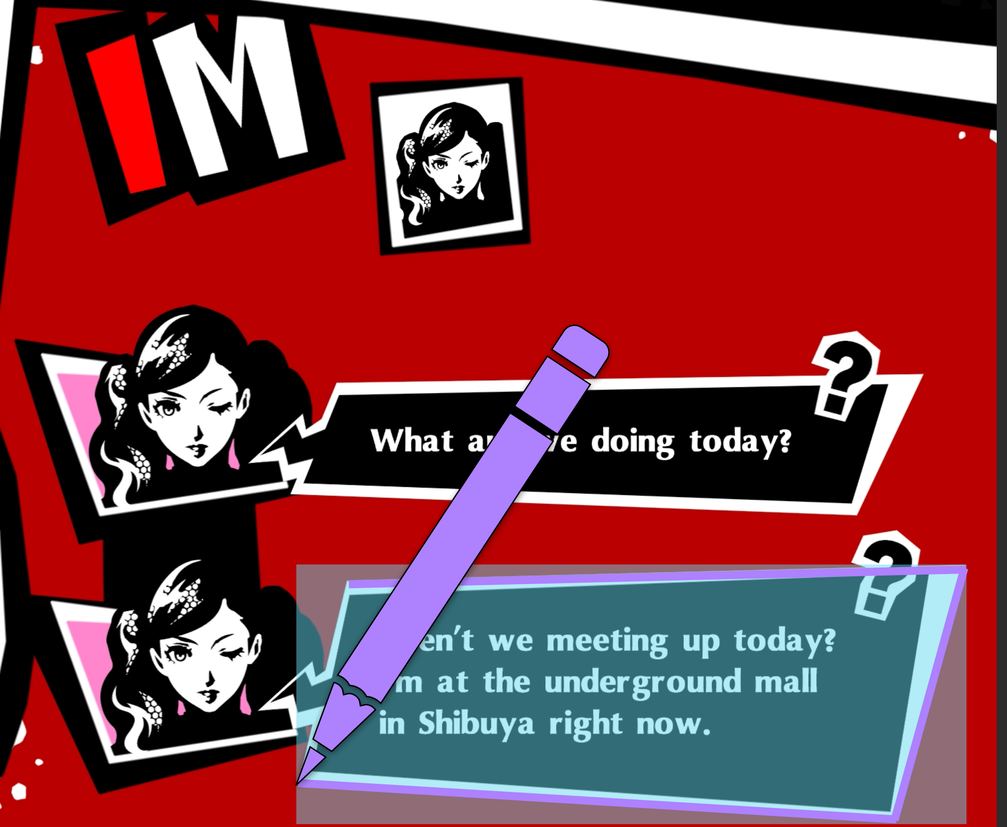

```
fun outerBox(): Shape = GenericShape { size →
  moveTo(31.7, 3.1)
}
```



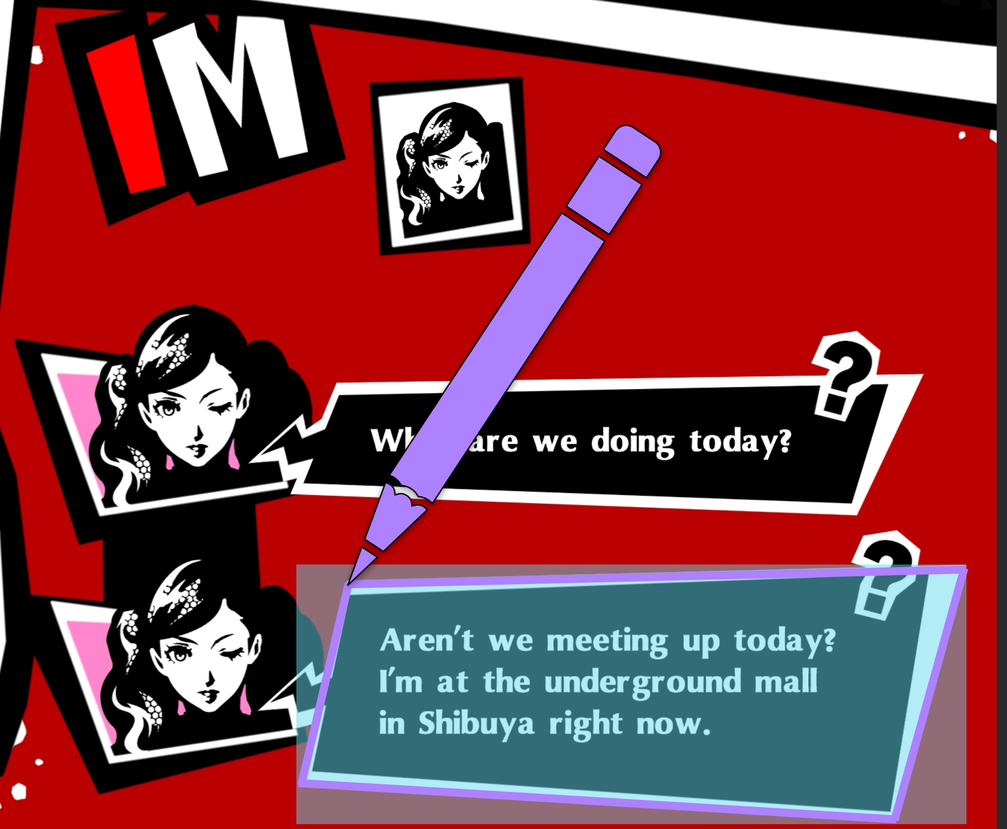

```
fun outerBox(): Shape = GenericShape { size →
  moveTo(31.7, 3.1)
  lineTo(size.width, 0f)
}
```



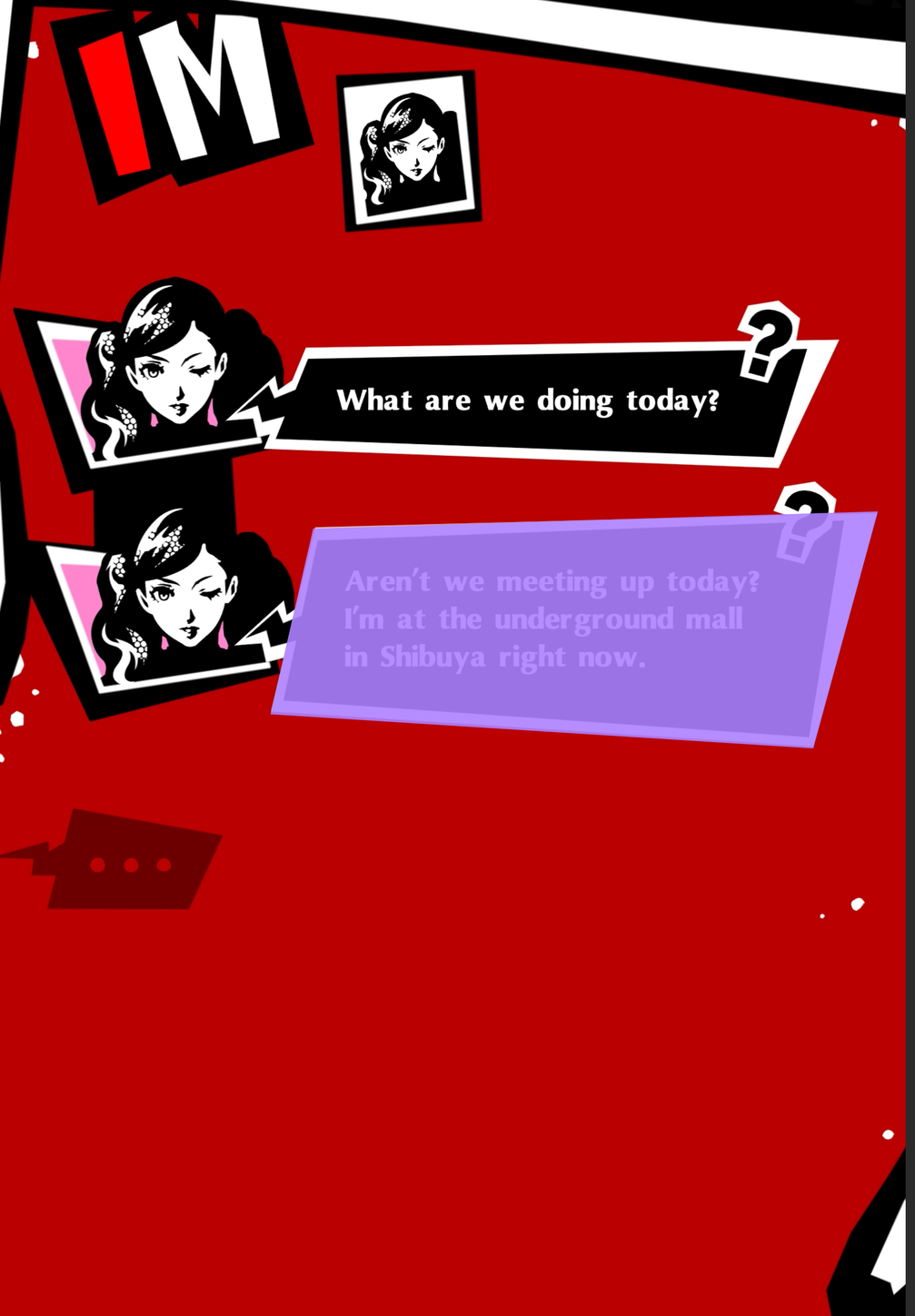

```
un outerBox(): Shape = GenericShape { size →
  moveTo(31.7, 3.1)
  lineTo(size.width, 0f)
  lineTo(size.width - 23, size.height)
}
```



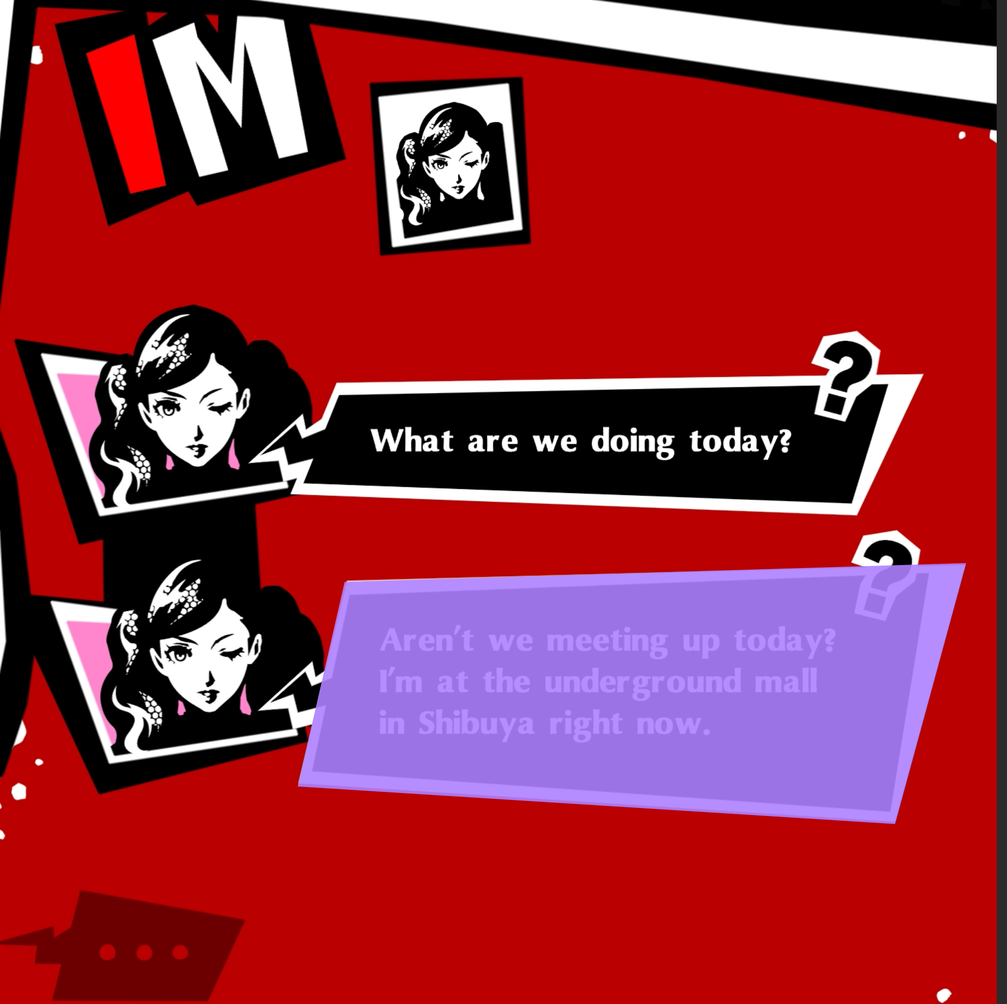

```
fun outerBox(): Shape = GenericShape { size →
  moveTo(31.7, 3.1)
  lineTo(size.width, 0f)
  lineTo(size.width - 23, size.height)
  lineTo(0f, size.height - 8)
}
```




```
fun outerBox(): Shape = GenericShape { size →
  moveTo(31.7, 3.1)
  lineTo(size.width, 0f)
  lineTo(size.width - 23, size.height)
  lineTo(0f, size.height - 8)
  close()
}
```




```
fun outerBox(): Shape = GenericShape { size →
  moveTo(31.7, 3.1)
  lineTo(size.width, 0f)
  lineTo(size.width - 23, size.height)
  lineTo(0f, size.height - 8)
  close()
}
```

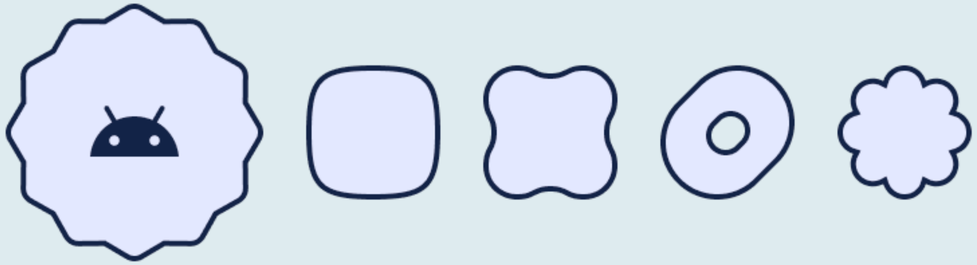



```
fun Density.outerBox(): Shape = GenericShape { s
  moveTo(31.7.dp.toPx(), 3.1.dp.toPx())
  lineTo(size.width, 0f)
  lineTo(size.width - 23.dp.toPx(), size.height)
  lineTo(0f, size.height - 8.dp.toPx())
  close()
}
```


```
fun Density.outerBox(): Shape = GenericShape { size →
  moveTo(31.7.dp.toPx(), 3.1.dp.toPx())
  lineTo(size.width, 0f)
  lineTo(size.width - 23.dp.toPx(), size.height)
  lineTo(0f, size.height - 8.dp.toPx())
  close()
}
```

```
fun Density.innerBox(): Shape = GenericShape { size →
  moveTo(33.dp.toPx(), 7.7.dp.toPx())
  lineTo(size.width - 13.dp.toPx(), 3.7.dp.toPx())
  lineTo(size.width - 25.7.dp.toPx(), size.height - 4.6.dp.toPx
  lineTo(20.4.dp.toPx(), size.height - 12.dp.toPx())
  close()
}
```

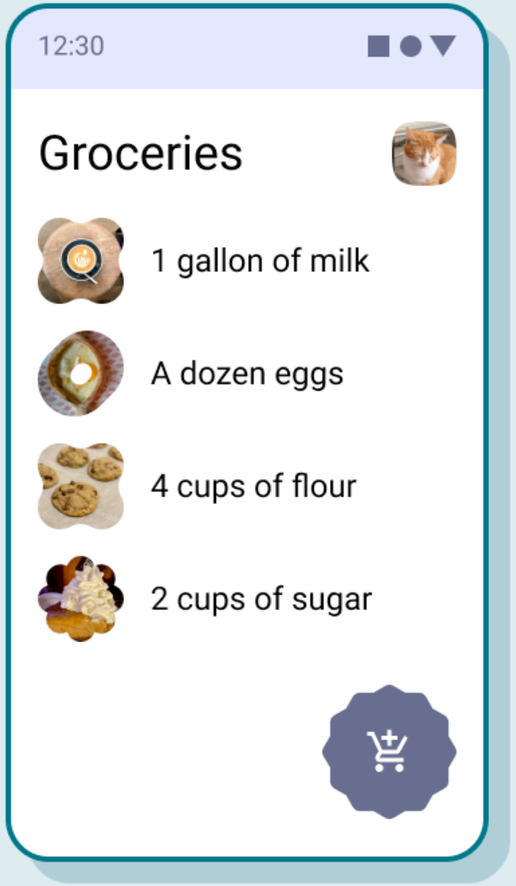
c5inco / shape-composer-figma

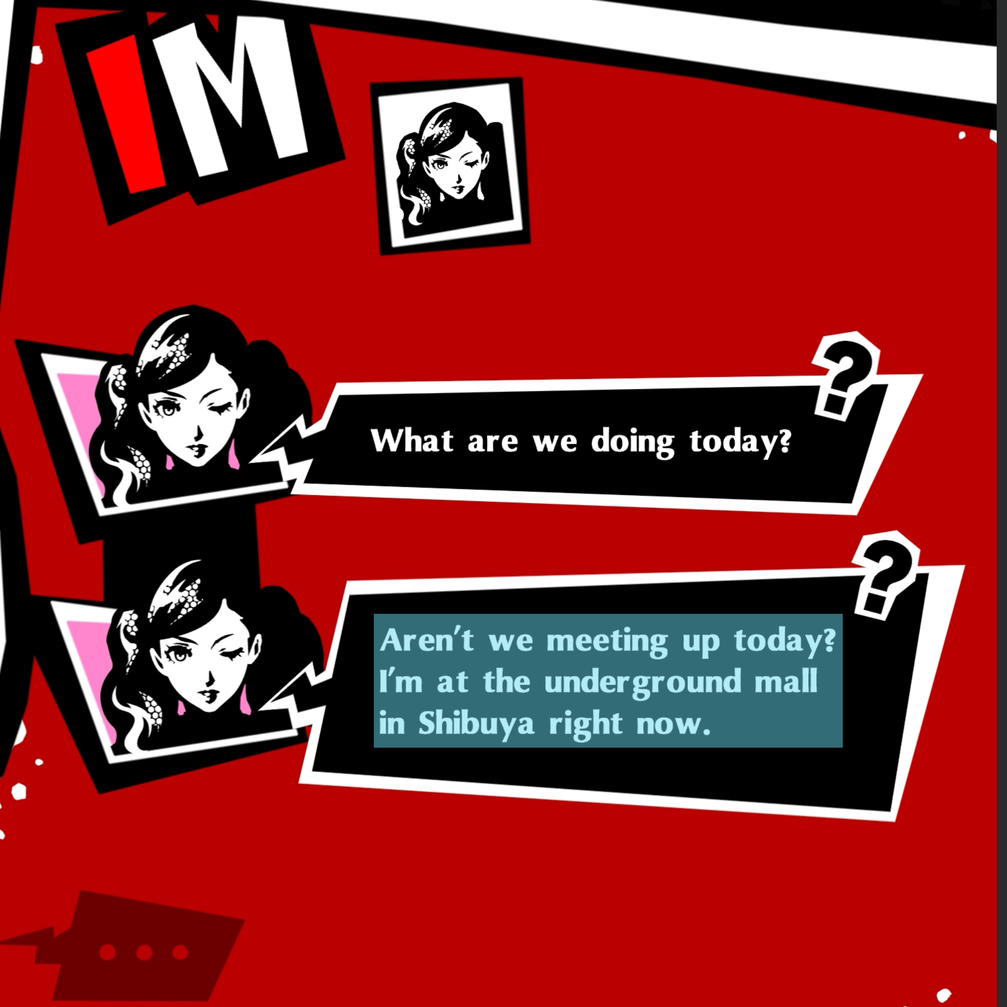


Shape Composer

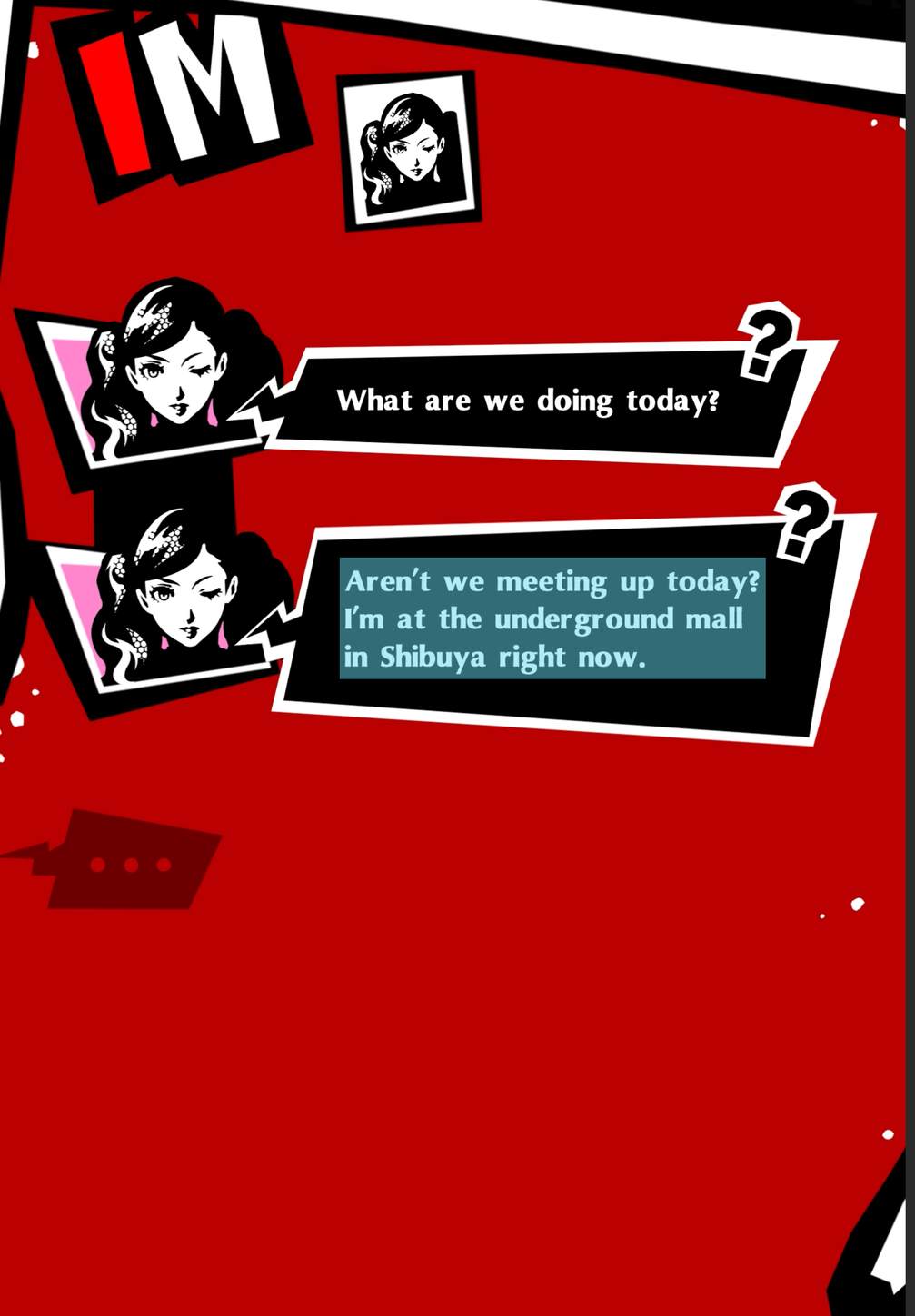
Export custom Shapes for Jetpack Compose

```
Modifier.clip(SquircleShape())
Modifier.shadow(elevation, PillCutoutShape())
Modifier.background(primaryColor, BottlecapShape())
```

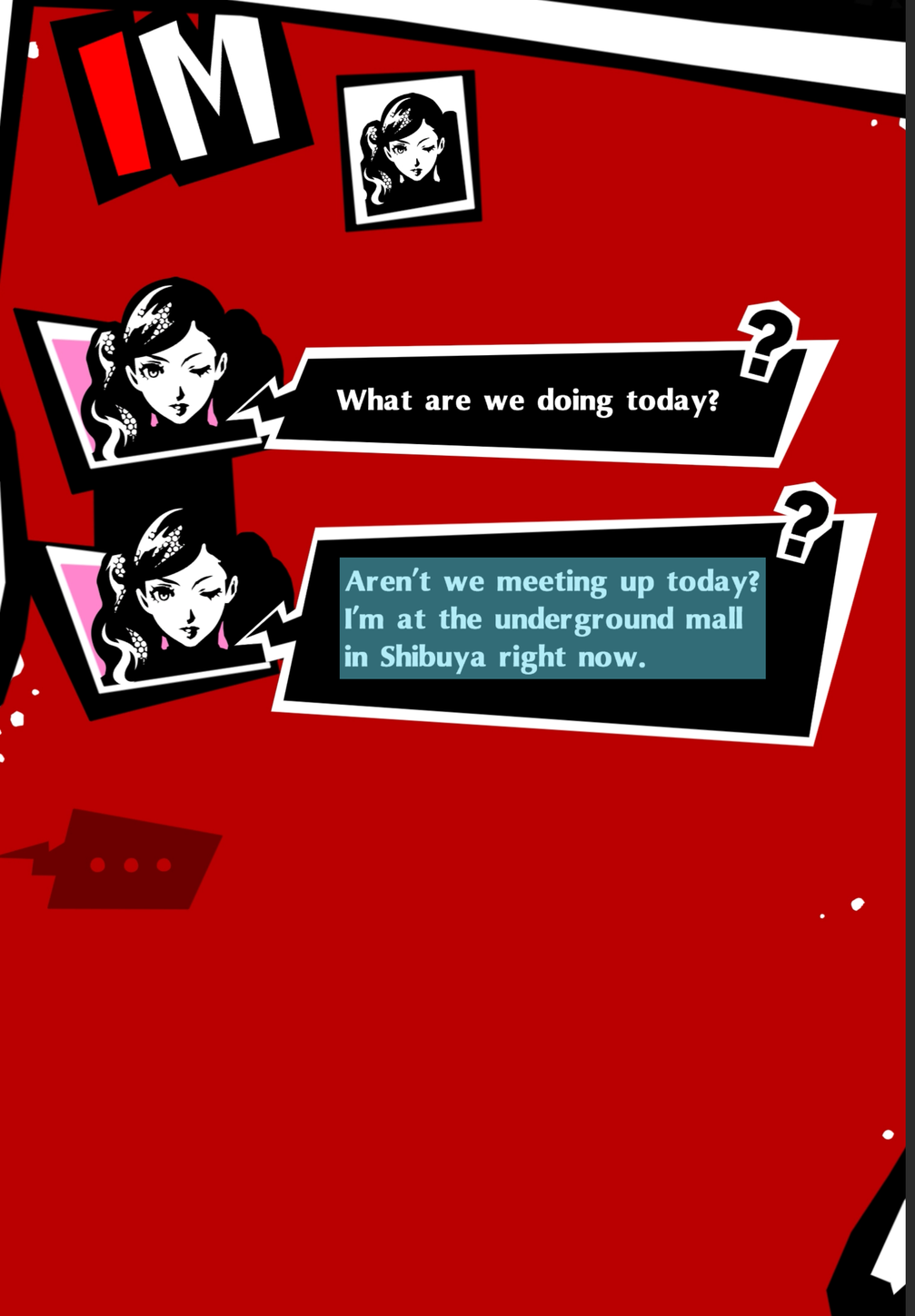




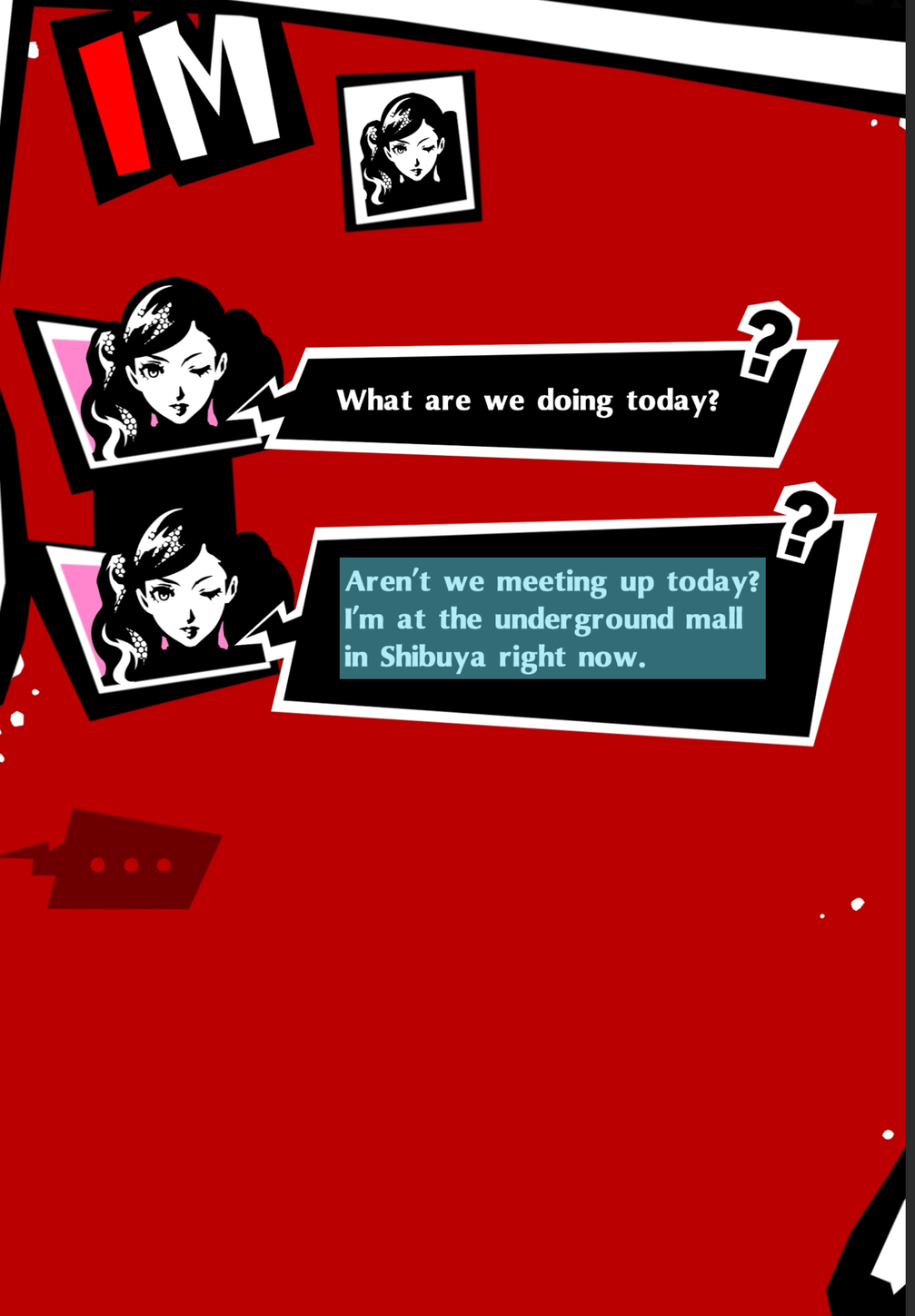

```
Text(
  text = message.text,
  style = MaterialTheme.typography.bodyMedium,
  color = Color.White,
  fontFamily = OptimaNova,
)
```



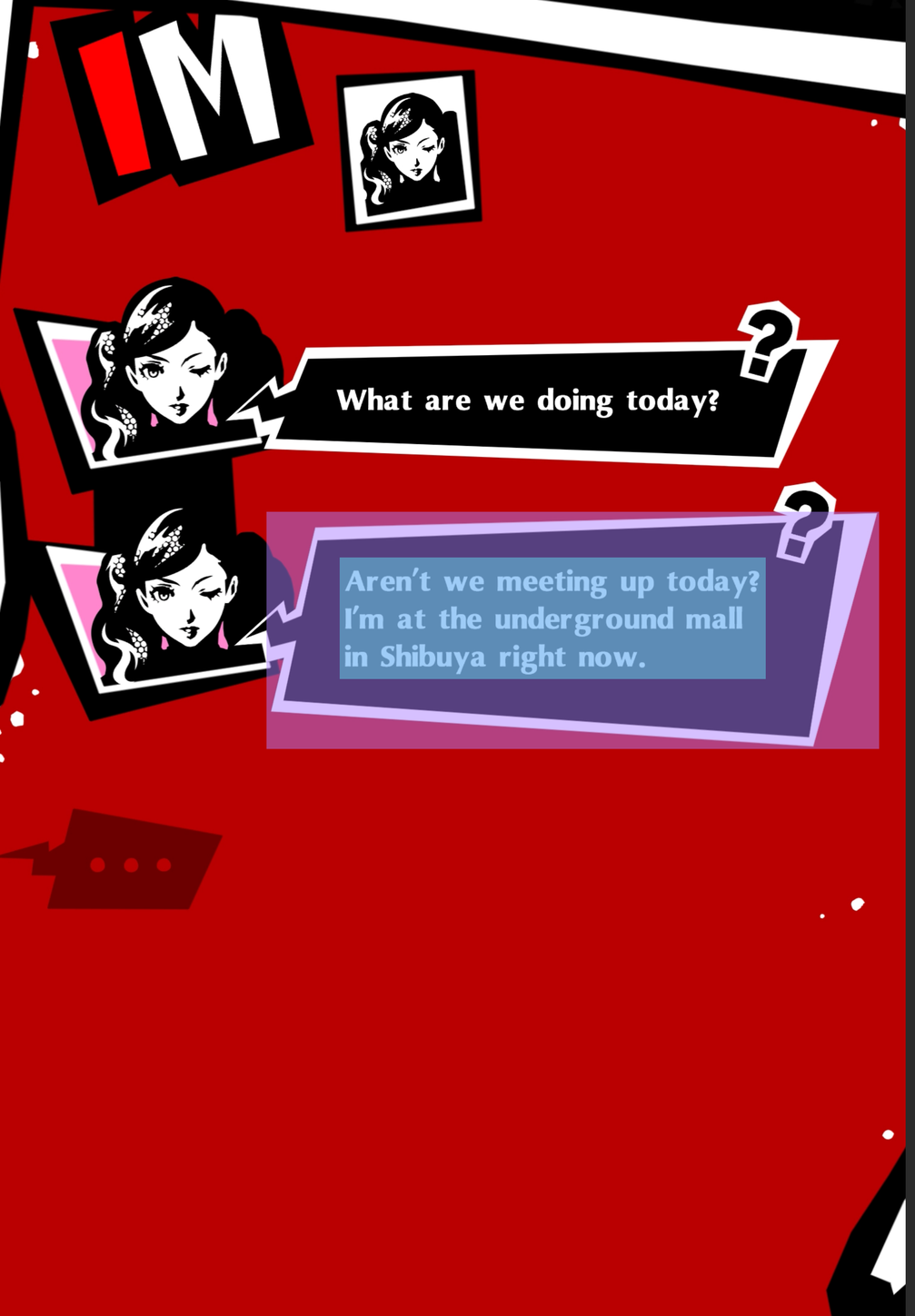

```
Text(
  text = message.text,
  style = MaterialTheme.typography.bodyMedium,
  color = Color.White,
  fontFamily = OptimaNova,
  modifier = Modifier
    .drawBehind {
      val outerBox = outerBox()
      val innerBox = innerBox()
    }
)
```



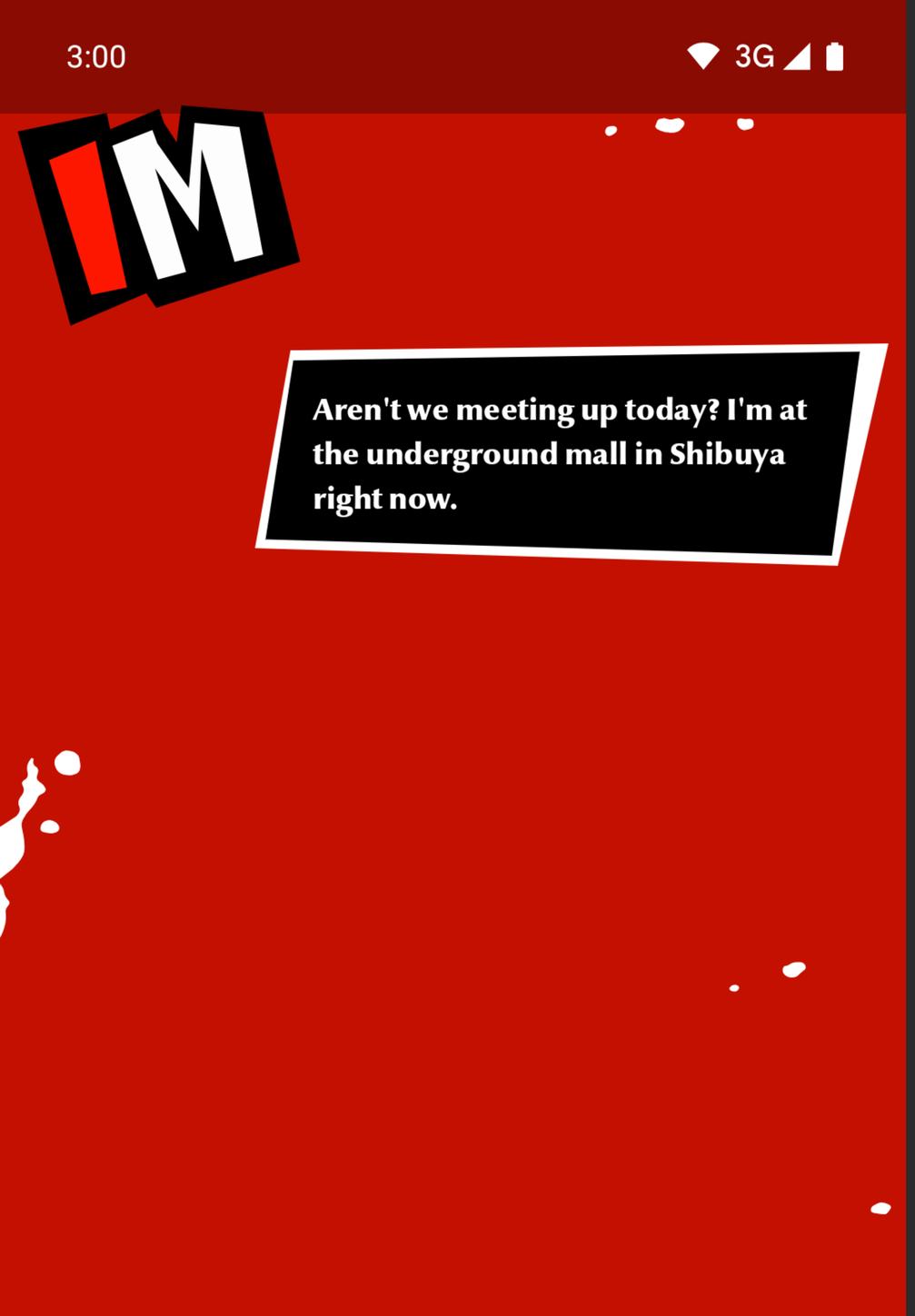

```
Text(
  text = message.text,
  style = MaterialTheme.typography.bodyMedium,
  color = Color.White,
  fontFamily = OptimaNova,
  modifier = Modifier
    .drawBehind {
      val outerBox = outerBox()
      val innerBox = innerBox()
      drawShape(outerBox, color = Color.White)
      drawShape(innerBox, color = Color.Black)
    }
)
```



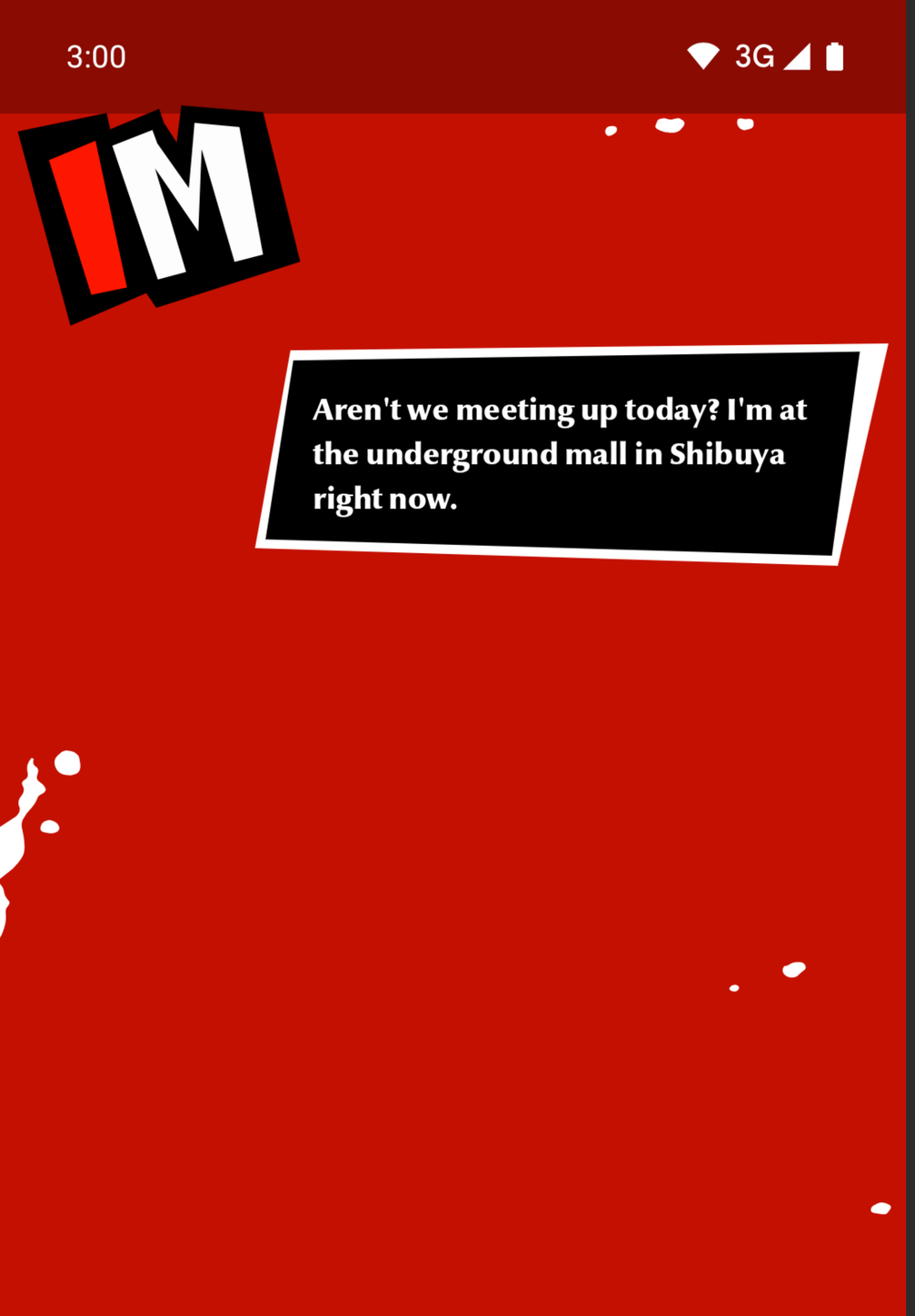

```
Text(
  text = message.text,
  style = MaterialTheme.typography.bodyMedium,
  color = Color.White,
  fontFamily = OptimaNova,
  modifier = Modifier
    .drawBehind {
      val outerBox = Outline(outerBox())
      val innerBox = Outline(innerBox())
      drawOutline(outerBox, color = Color.White)
      drawOutline(innerBox, color = Color.Black)
    }
)
```

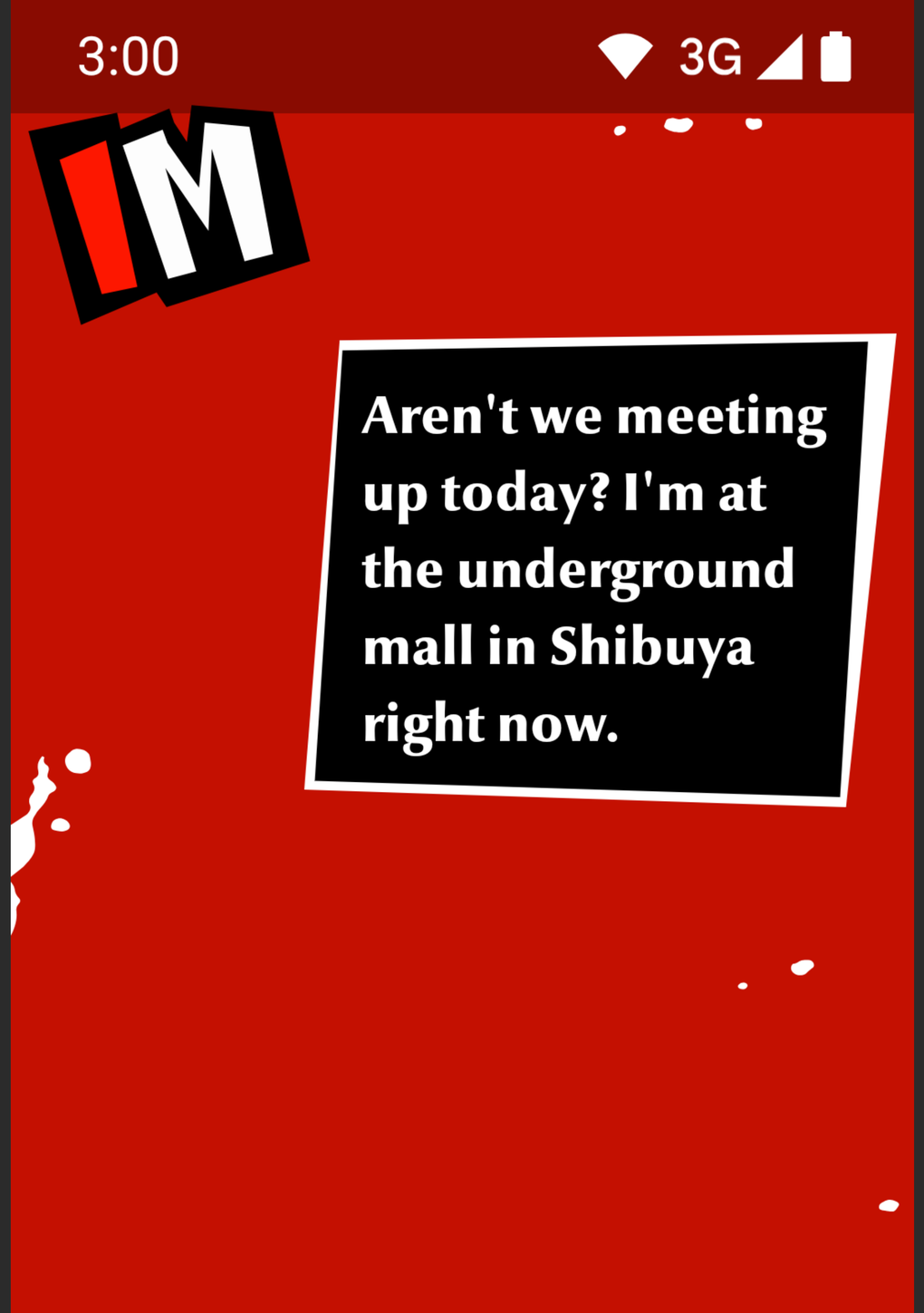



```
Text(
  text = message.text,
  style = MaterialTheme.typography.bodyMedium,
  color = Color.White,
  fontFamily = OptimaNova,
  modifier = Modifier
    .drawBehind {
      val outerBox = Outline(outerBox())
      val innerBox = Outline(innerBox())
      drawOutline(outerBox, color = Color.White)
      drawOutline(innerBox, color = Color.Black)
    }
    .padding(...)
)
```



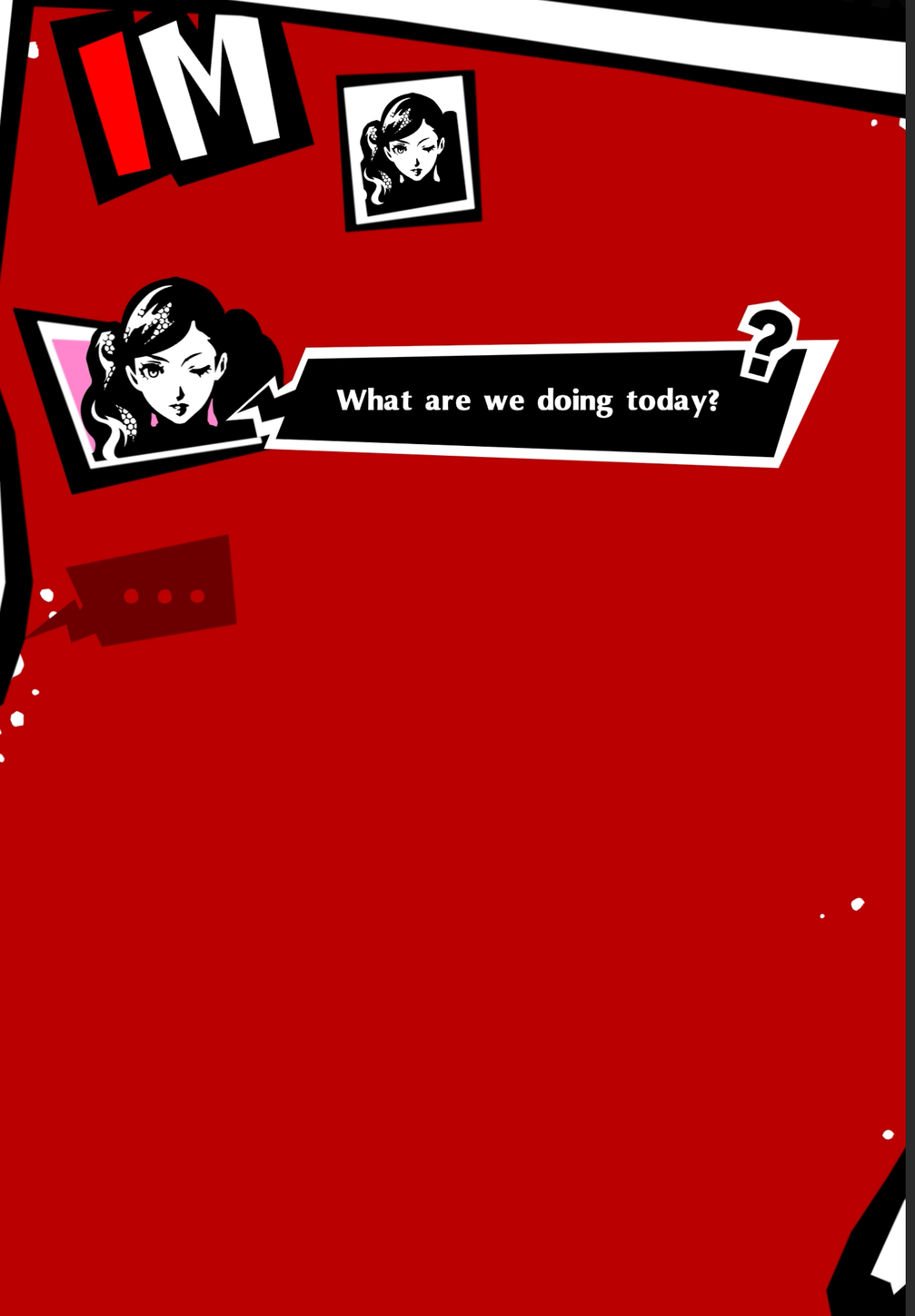

```
Text(
  text = message.text,
  style = MaterialTheme.typography.bodyMedium,
  color = Color.White,
  fontFamily = OptimaNova,
  modifier = Modifier
    .drawBehind {
      val outerBox = Outline(outerBox())
      val innerBox = Outline(innerBox())
      drawOutline(outerBox, color = Color.White)
      drawOutline(innerBox, color = Color.Black)
    }
    .padding(...)
)
```


3:00
3G
Aren't we meeting up today? I'm at the underground mall in Shibuya right now.

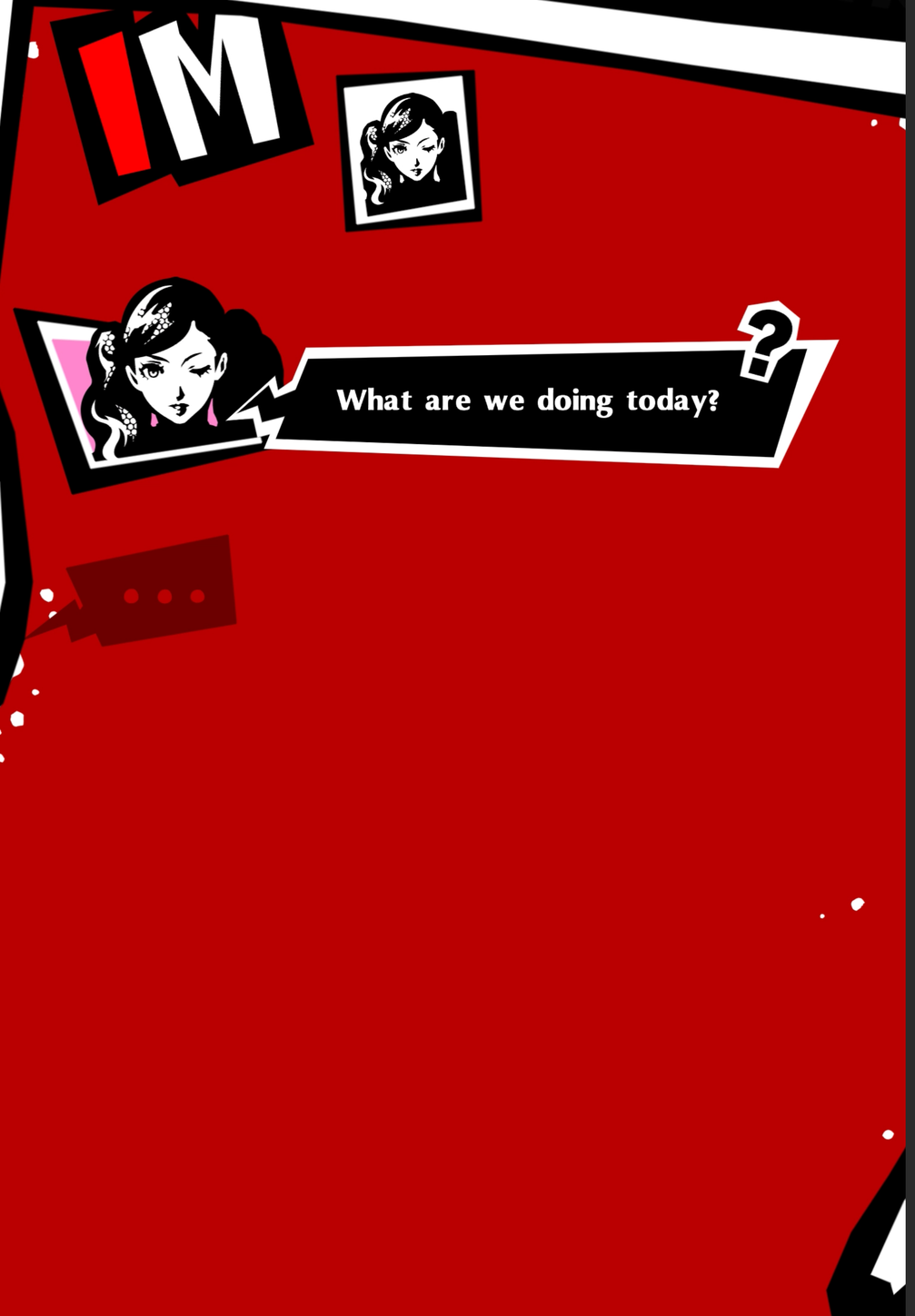

3:00
3G
Aren't we meeting up today? I'm at the underground mall in Shibuya right now.

Avatar

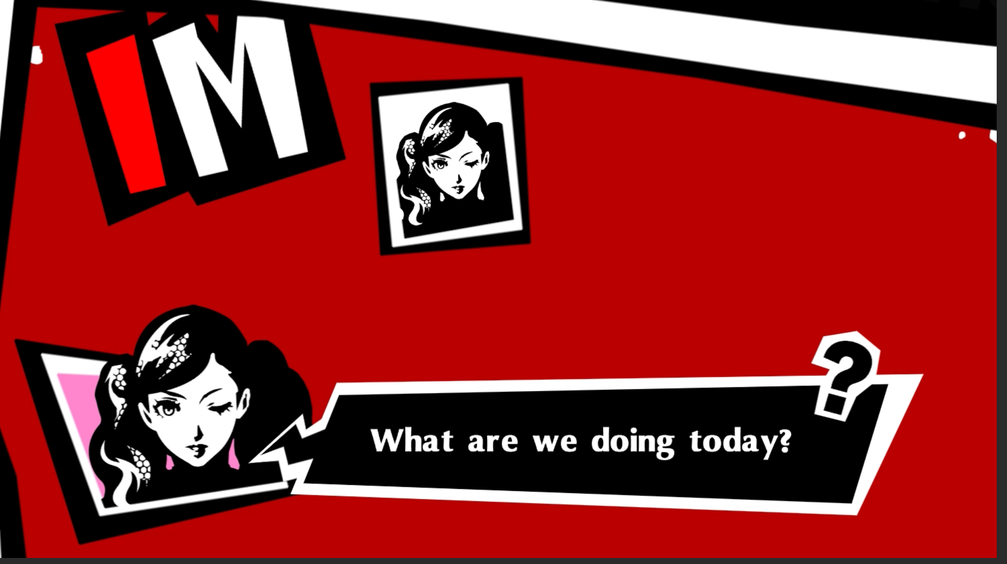

2/1 WEDNESDAY
After School
3
IM
IM
What are we doing today?
...
Days until
Maruki Acts: 2
Steal Maruki's heart
Prepare for meeting with Maruki
FFWD
Auto
Scroll
Next
©ATLUS ©SEGA

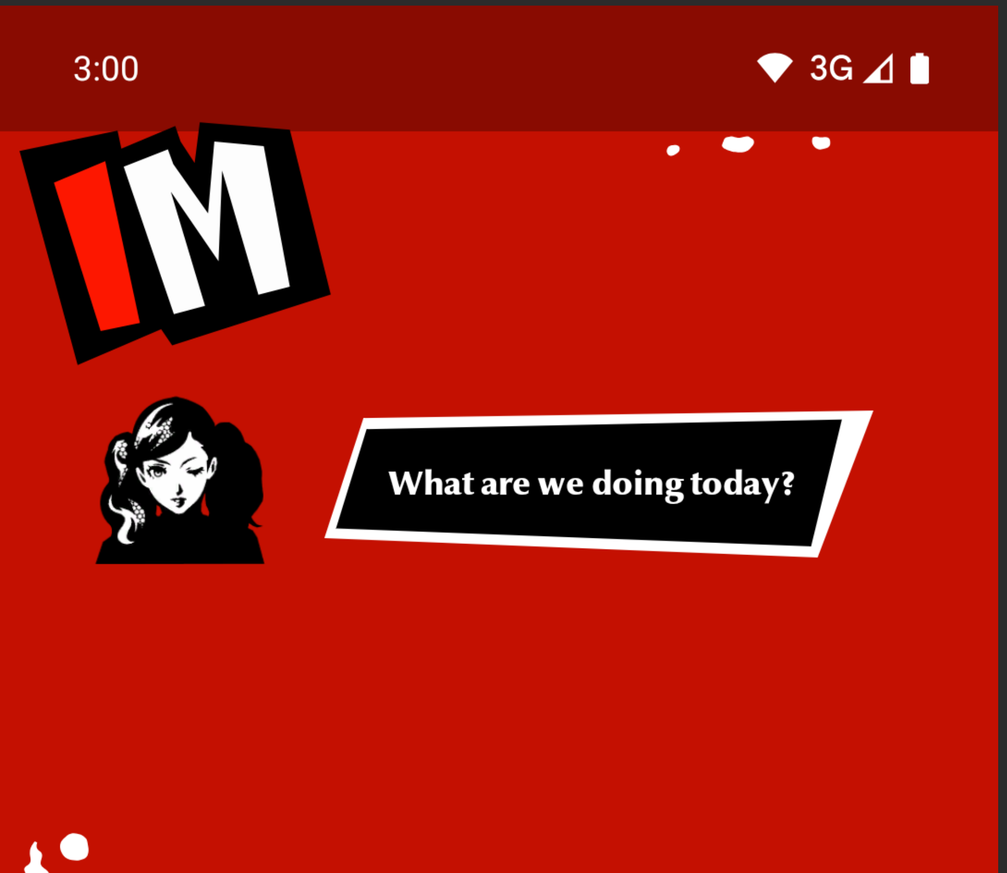



```
Row {
  TextBox()
}
```

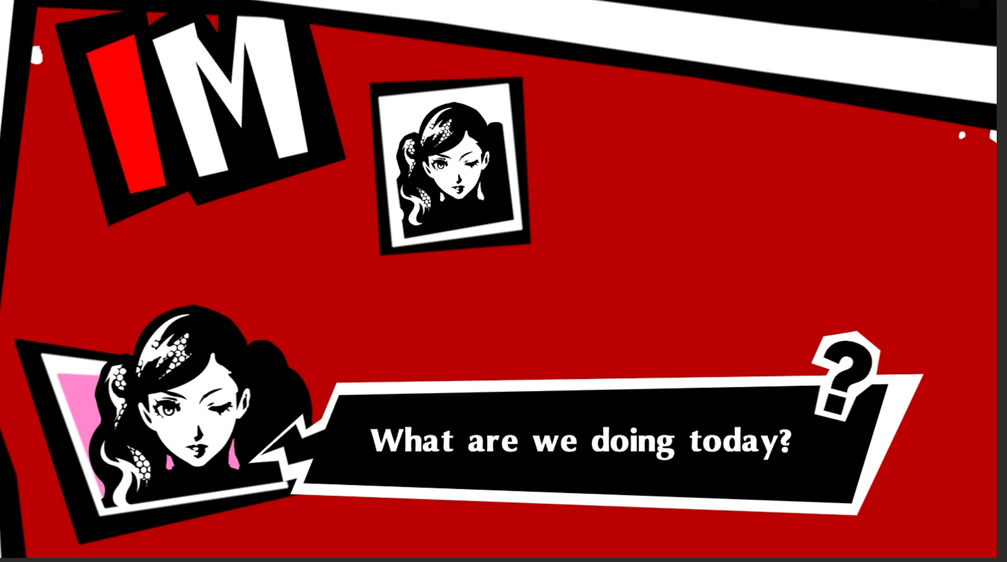


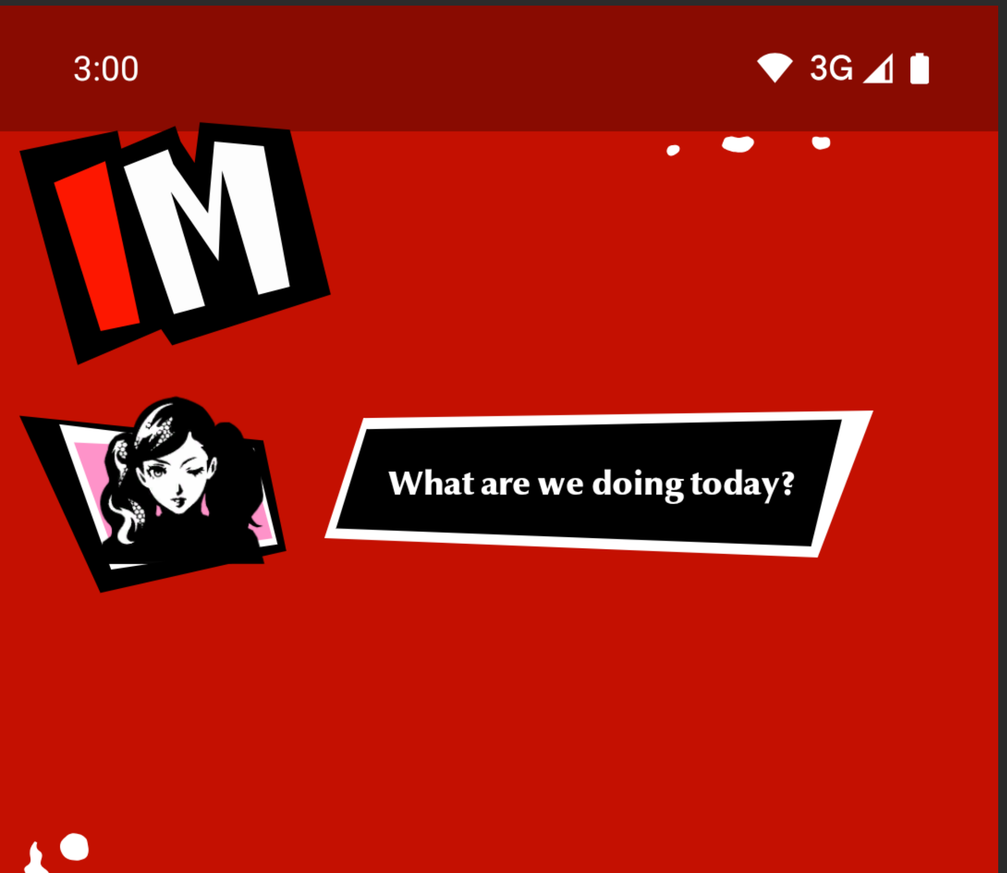

```
Row {
  Image(
    painter = painterResource(R.drawable.ann),
    contentDescription = null,
  )
  TextBox()
}
```



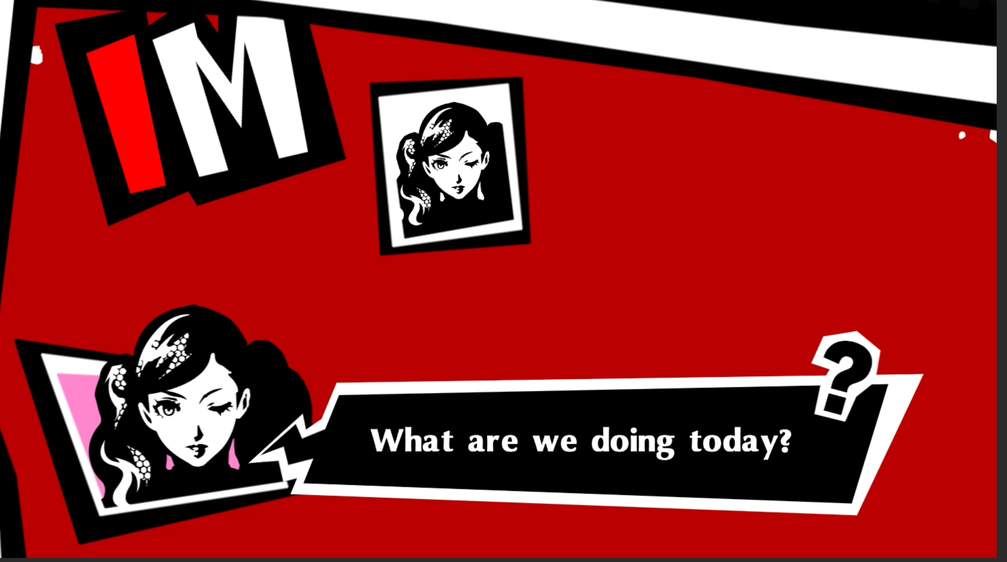


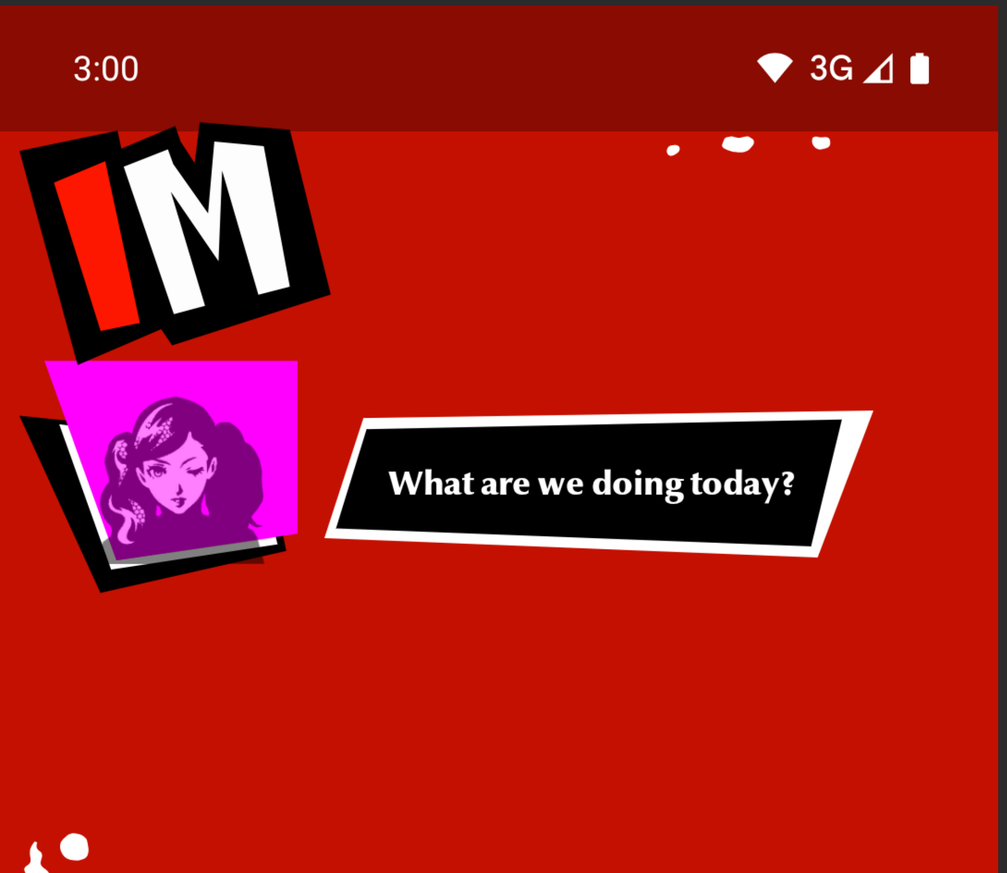

```
Row {
  Image(
    painter = painterResource(R.drawable.ann),
    contentDescription = null,
  )
  TextBox()
}
```



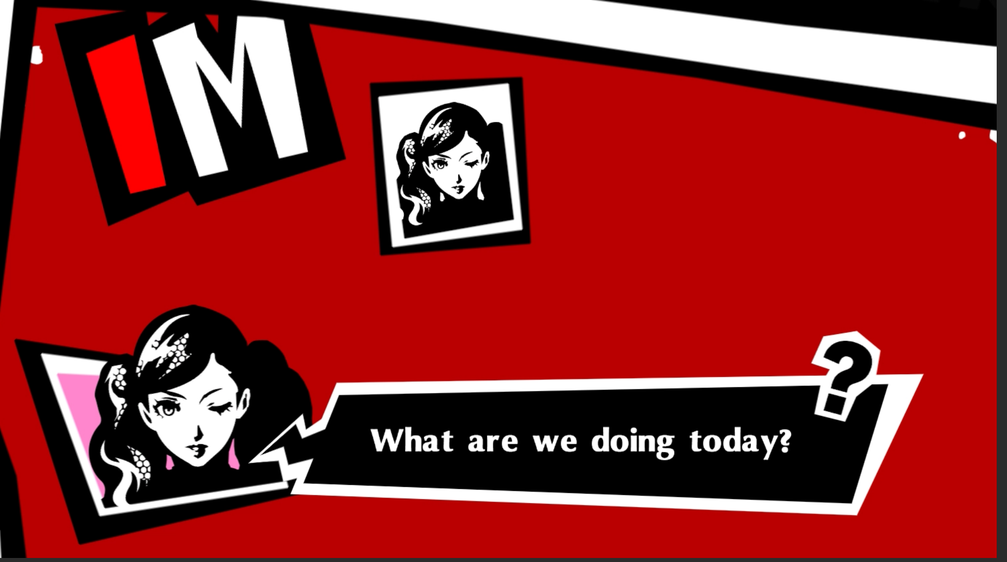


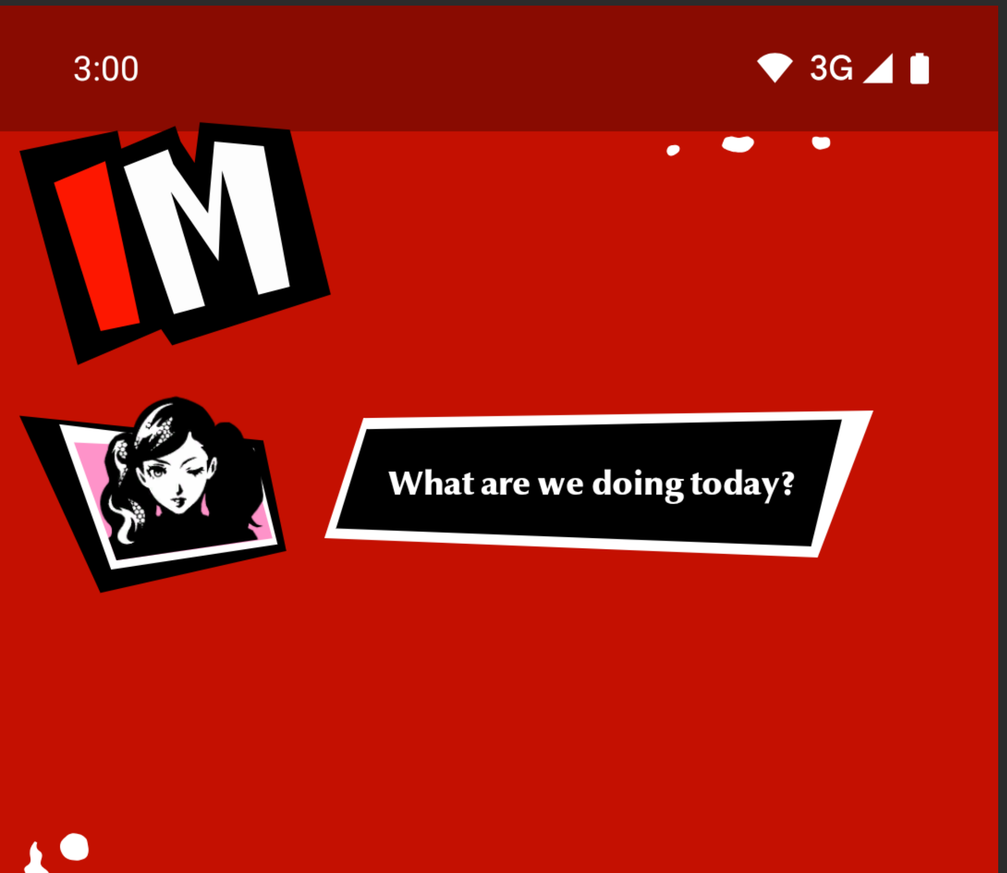

```
Box(
  modifier = Modifier
    .drawBehind {
      drawOutline(avatarBlackBox(), Color.Black)
      drawOutline(avatarWhiteBox(), Color.White)
      drawOutline(avatarColoredBox(), pink)
    }
) {
  Image(
    painter = painterResource(R.drawable.ann),
    contentDescription = null,
  )
}
```



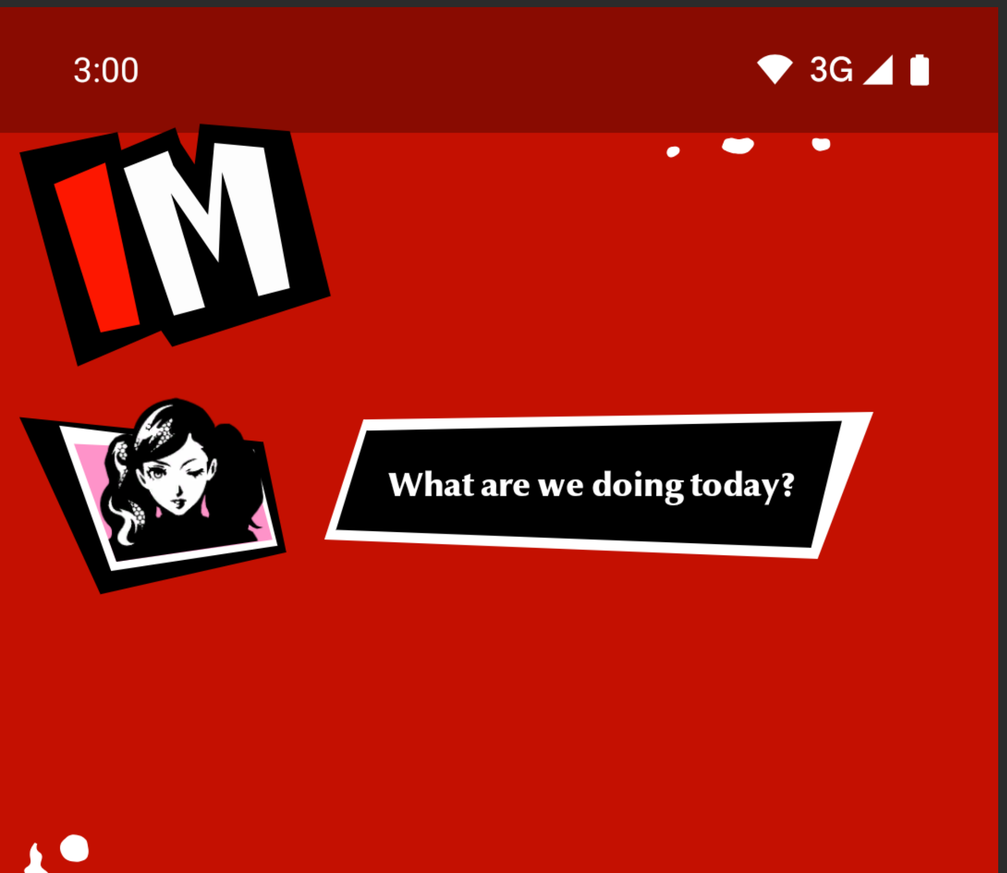



```
Box(
  modifier = Modifier
    .drawBehind {
      drawOutline(avatarBlackBox(), Color.Black)
      drawOutline(avatarWhiteBox(), Color.White)
      drawOutline(avatarColoredBox(), pink)
      drawOutline(avatarClipBox(), Color.Magenta)
    }
) {
  Image(
    painter = painterResource(R.drawable.ann),
    contentDescription = null,
  )
}
```

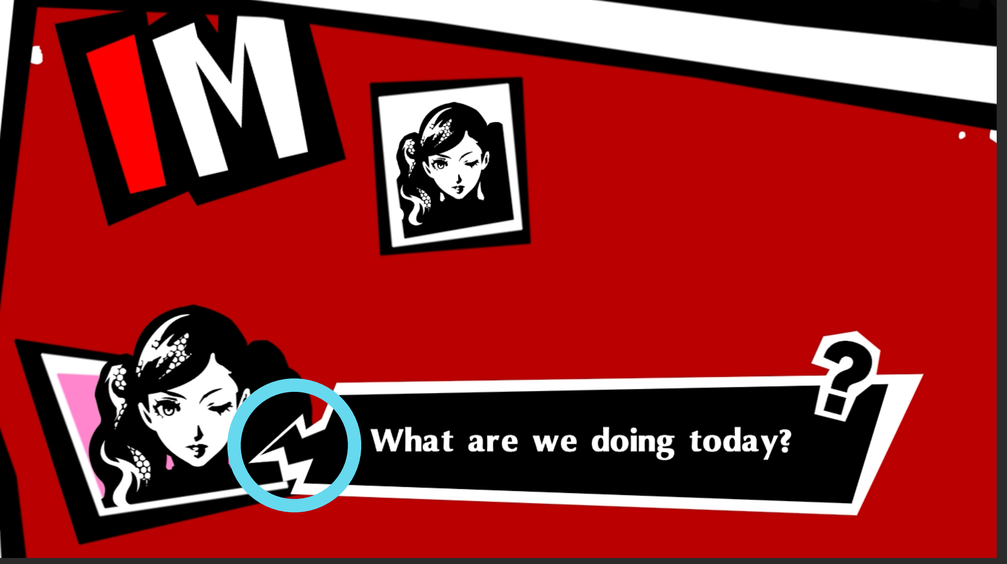


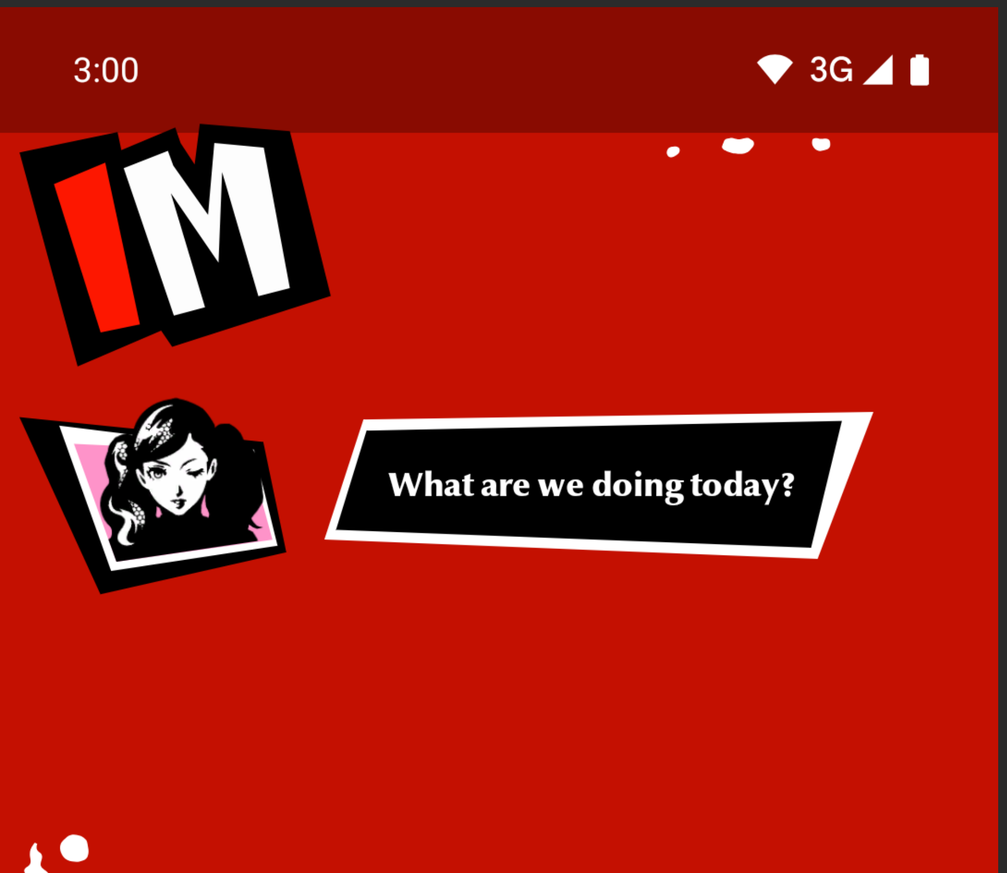



```
Box(
  modifier = Modifier
    .drawBehind {
      drawOutline(avatarBlackBox(), Color.Black)
      drawOutline(avatarWhiteBox(), Color.White)
      drawOutline(avatarColoredBox(), pink)
    }
    .clip(avatarClipBox())
) {
  Image(
    painter = painterResource(R.drawable.ann),
    contentDescription = null,
  )
}
```

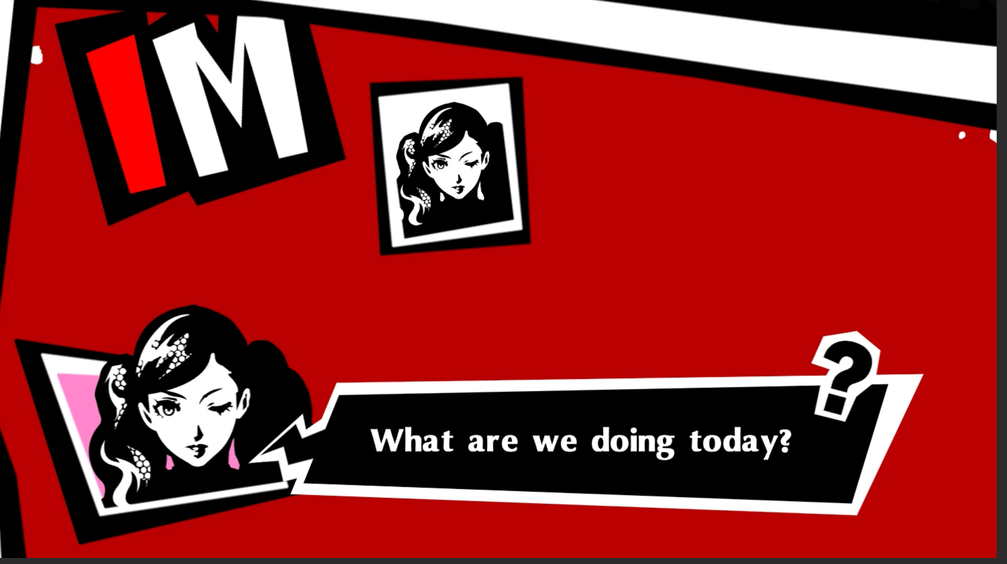


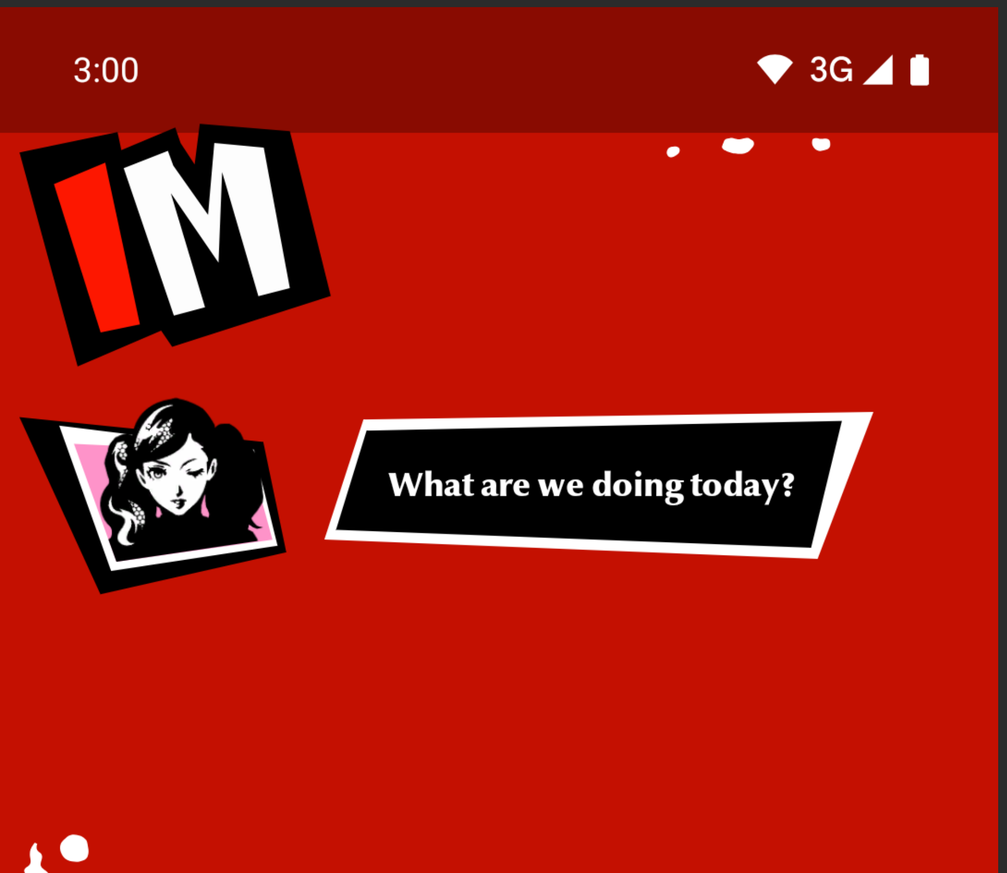

```
Row {
  Avatar()
  TextBox()
}
```

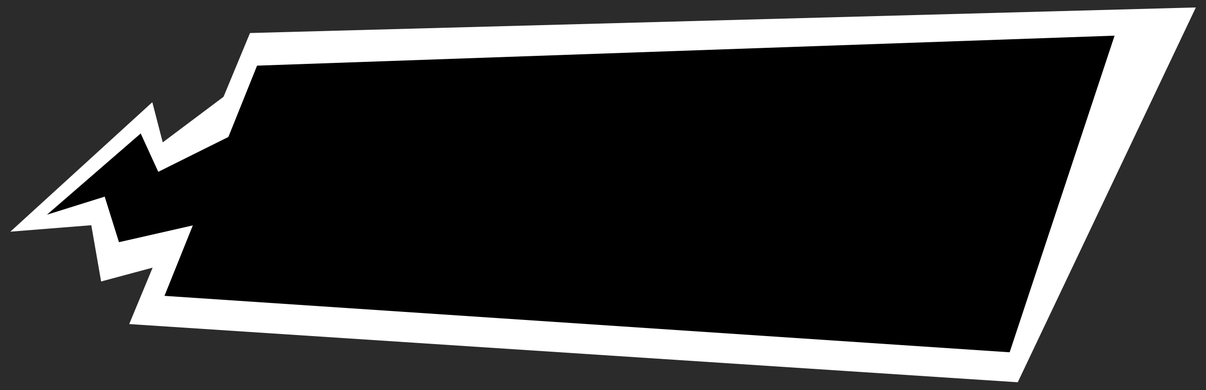


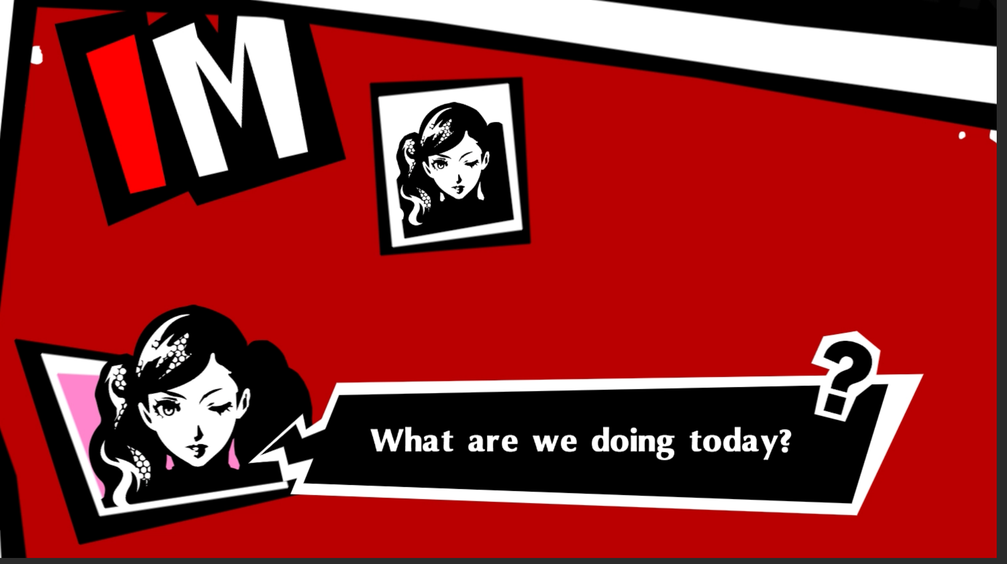


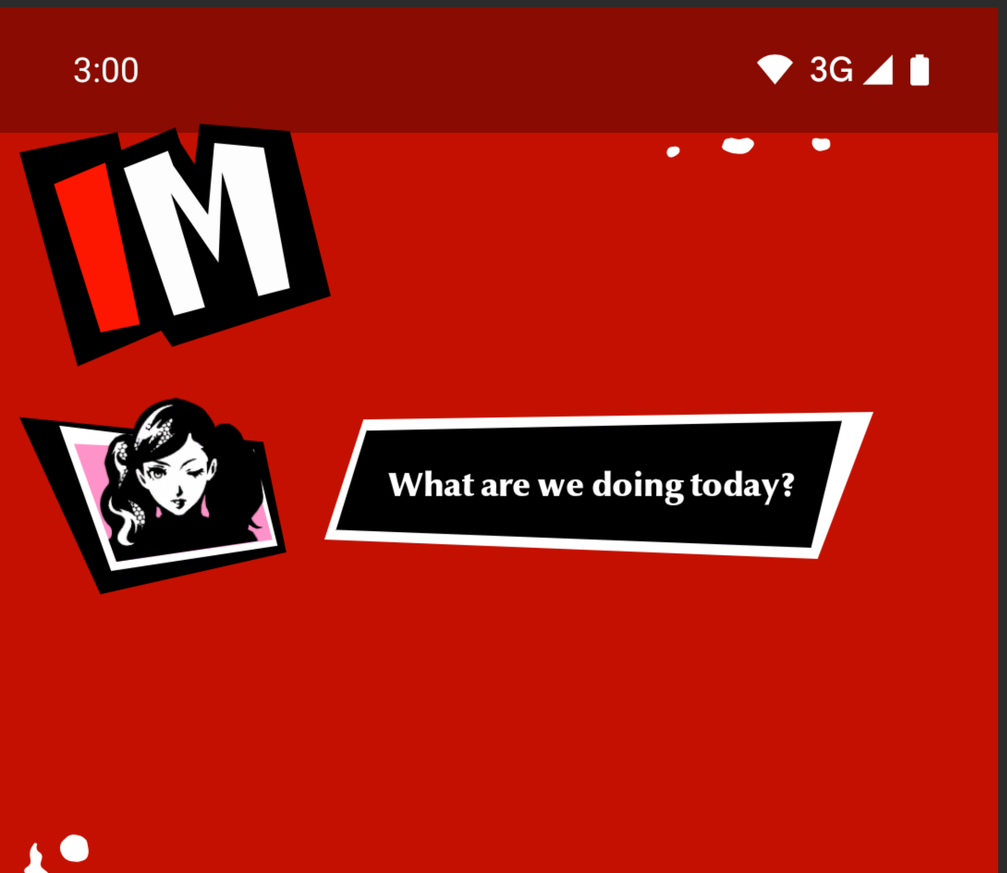

```
Row {
  Avatar()
  Stem()?
  TextBox()
}
```

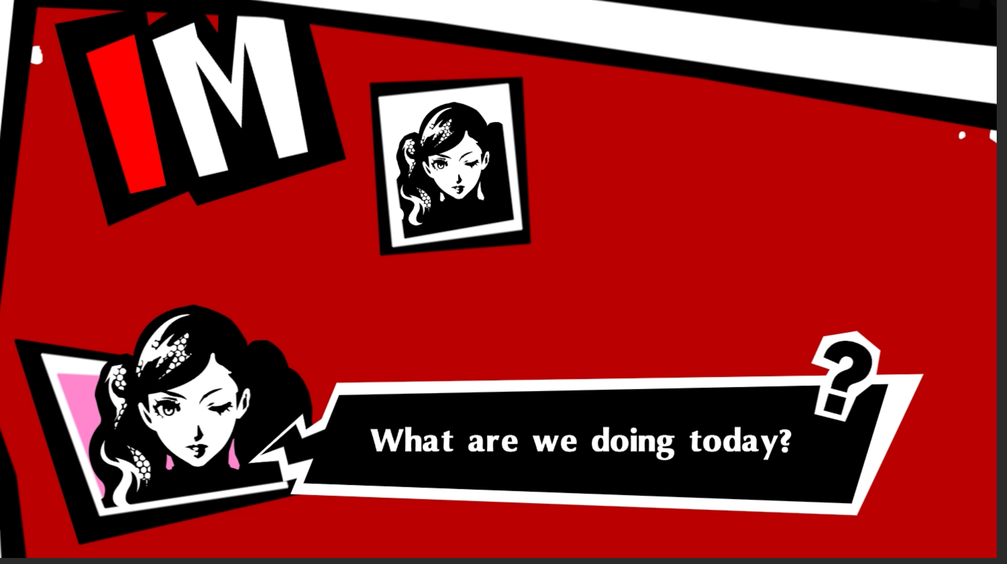


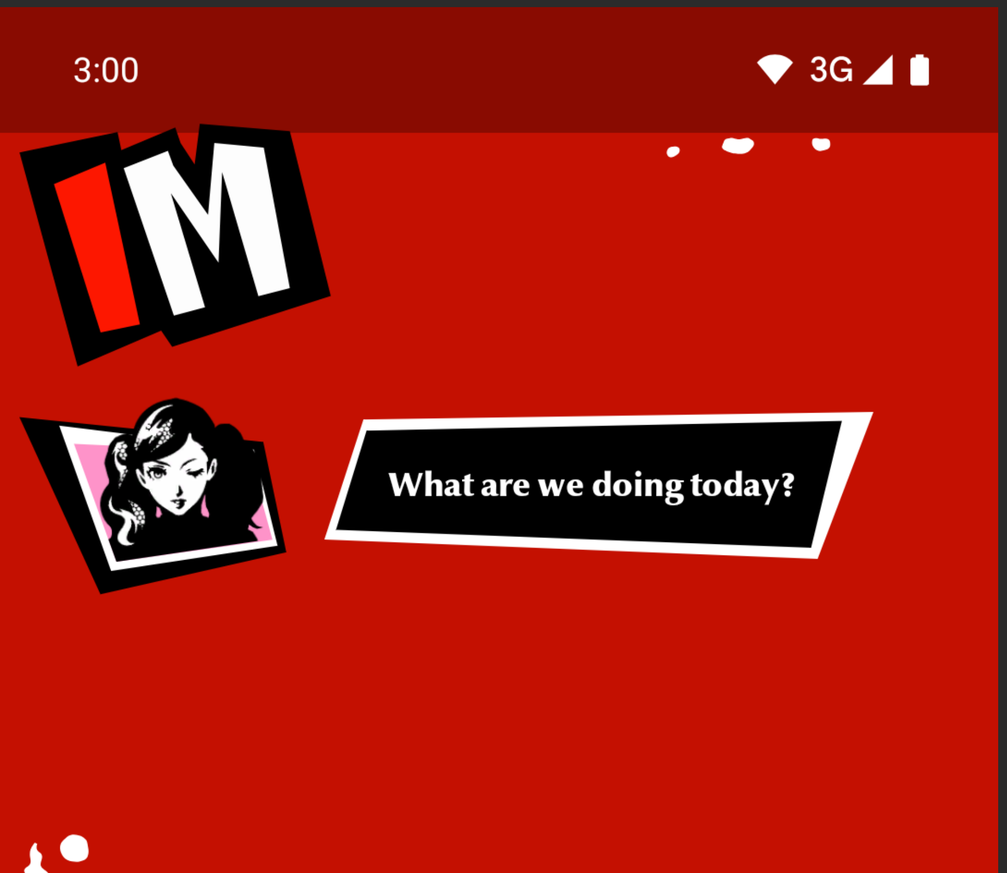


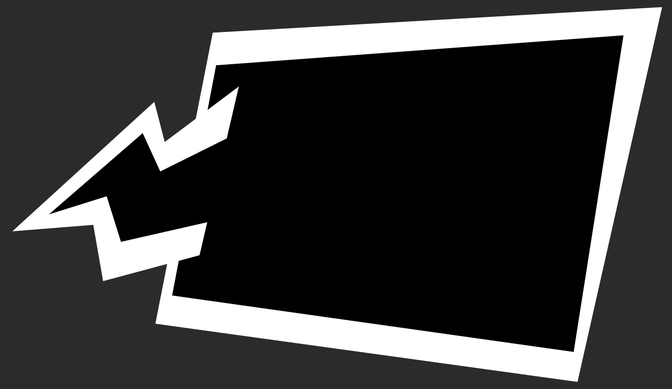


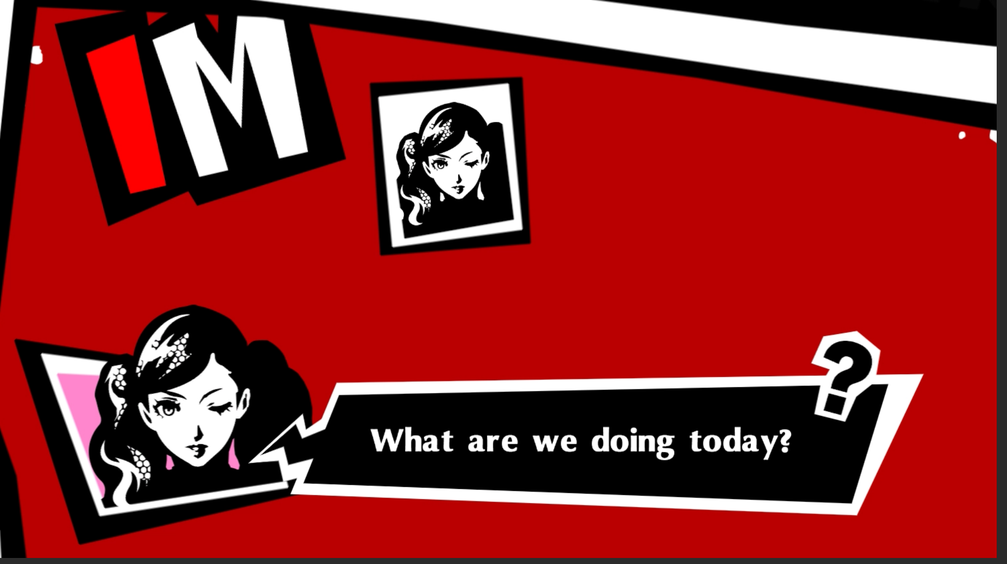


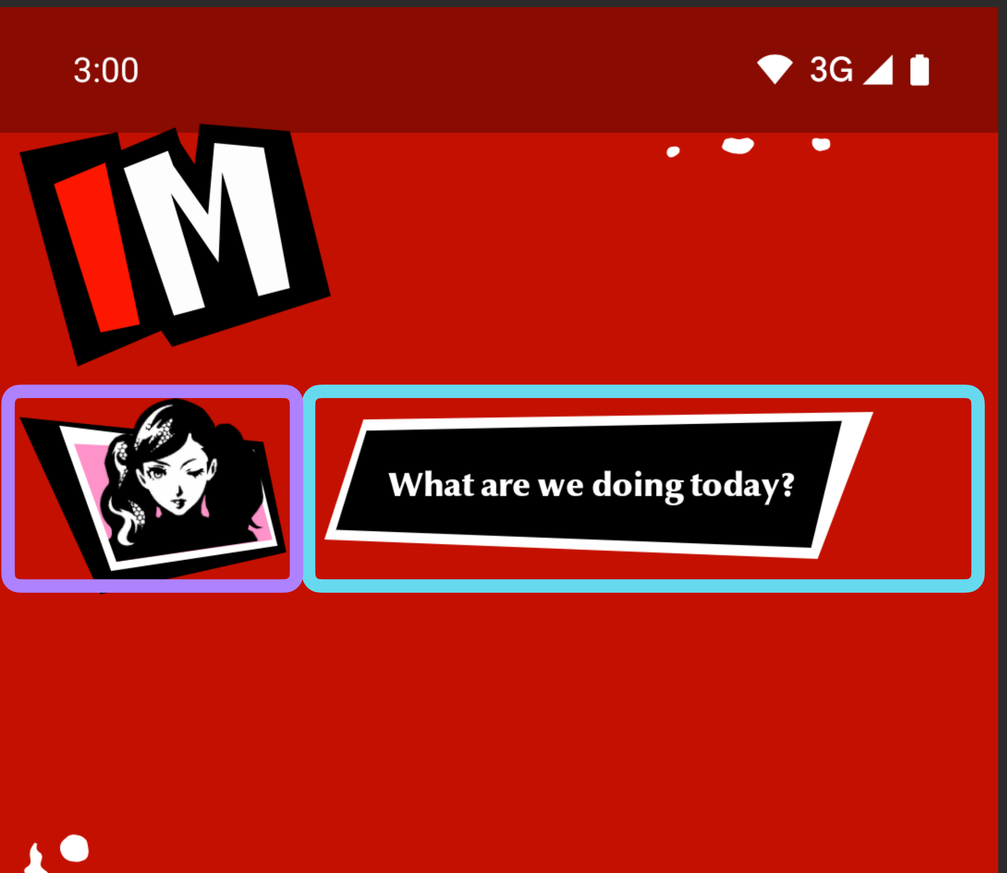




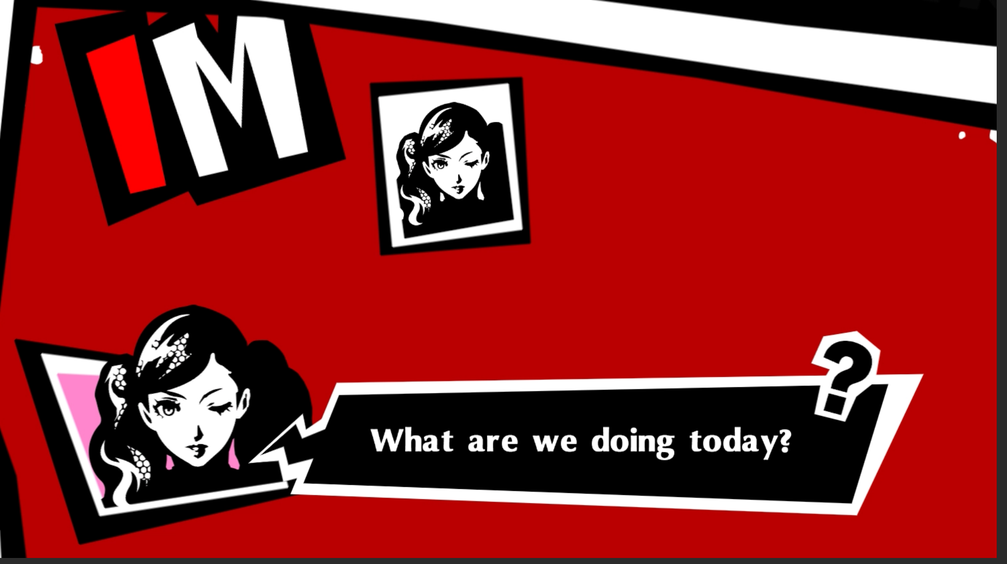


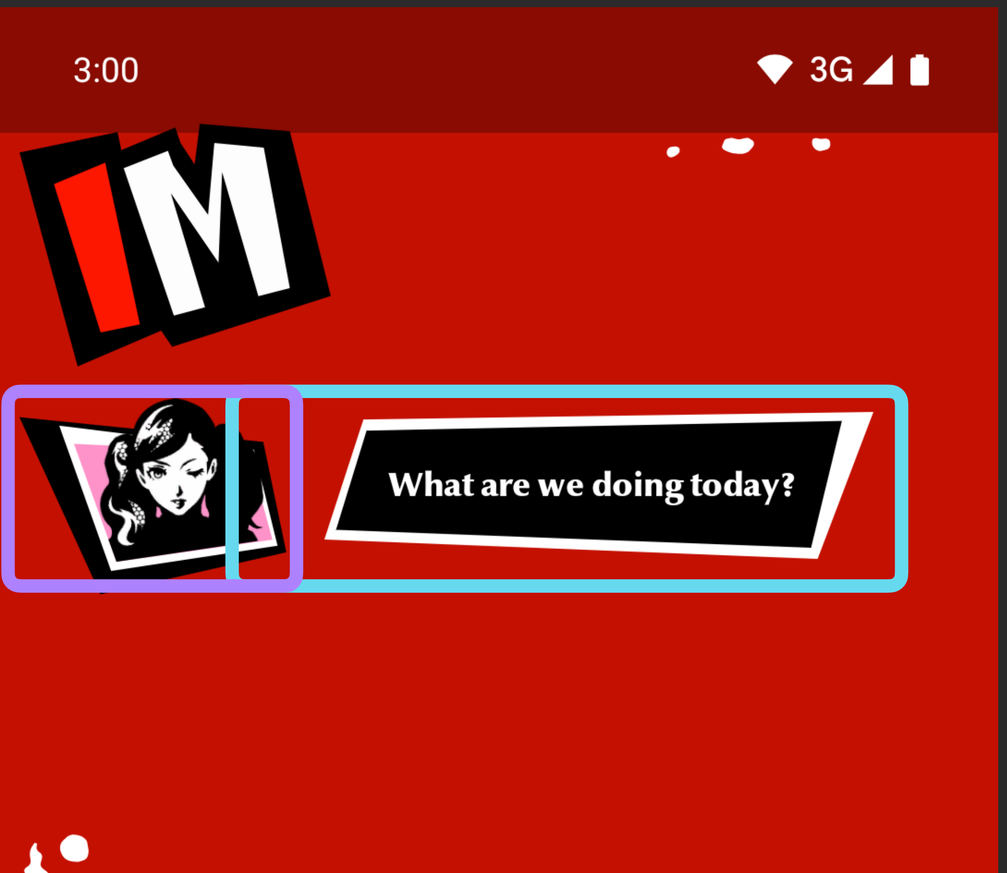




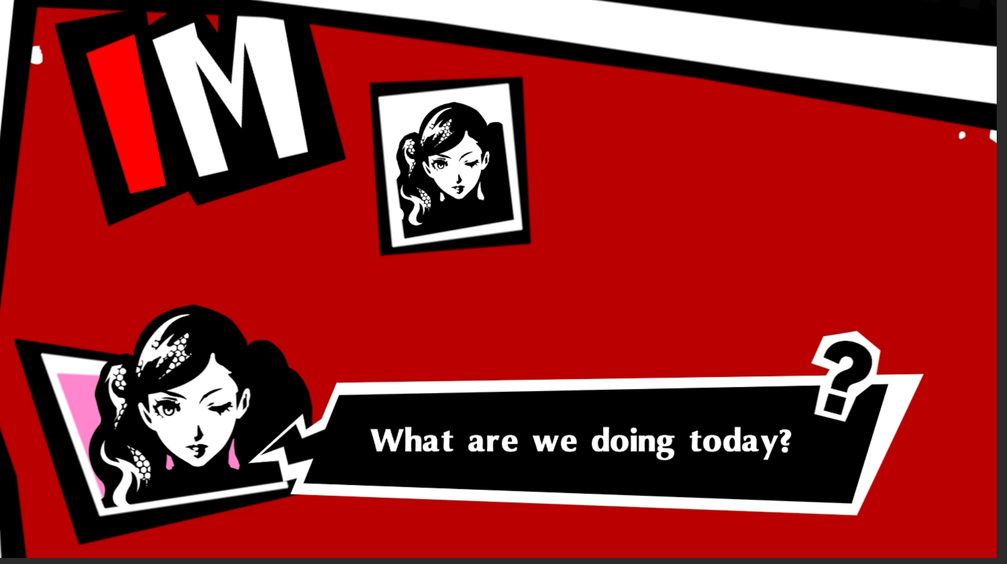


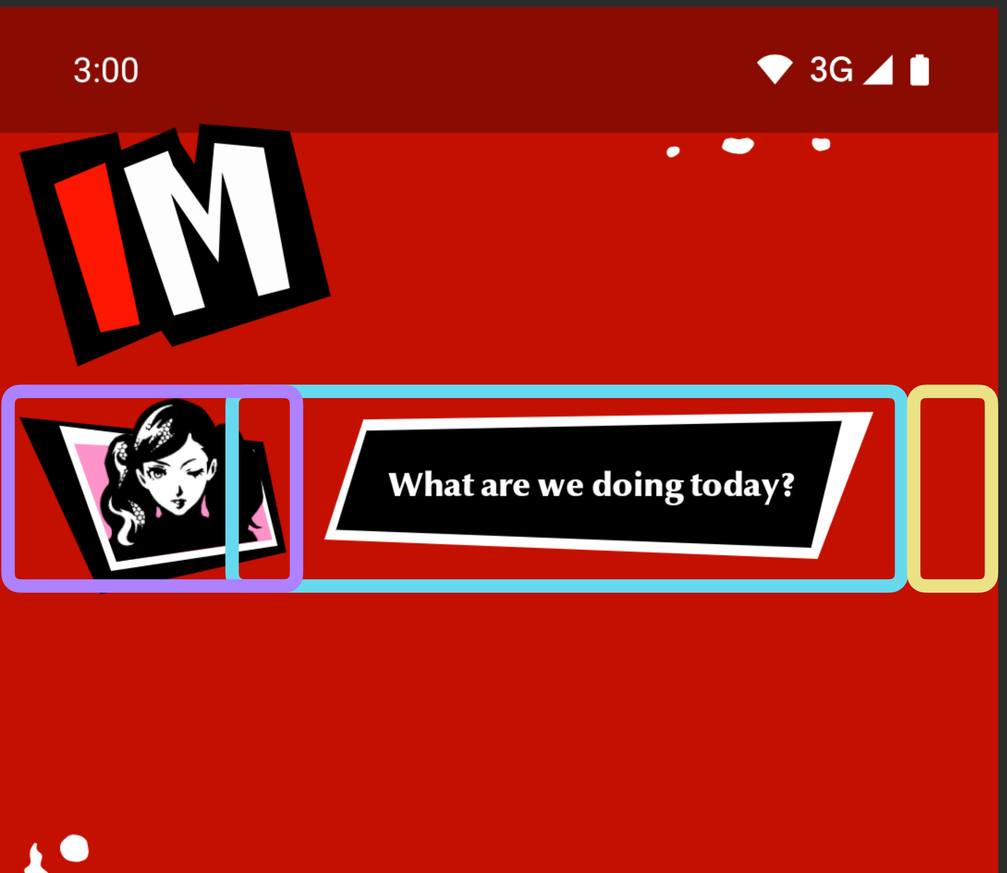

```
Row {
  Avatar()
  TextBox()
}
```



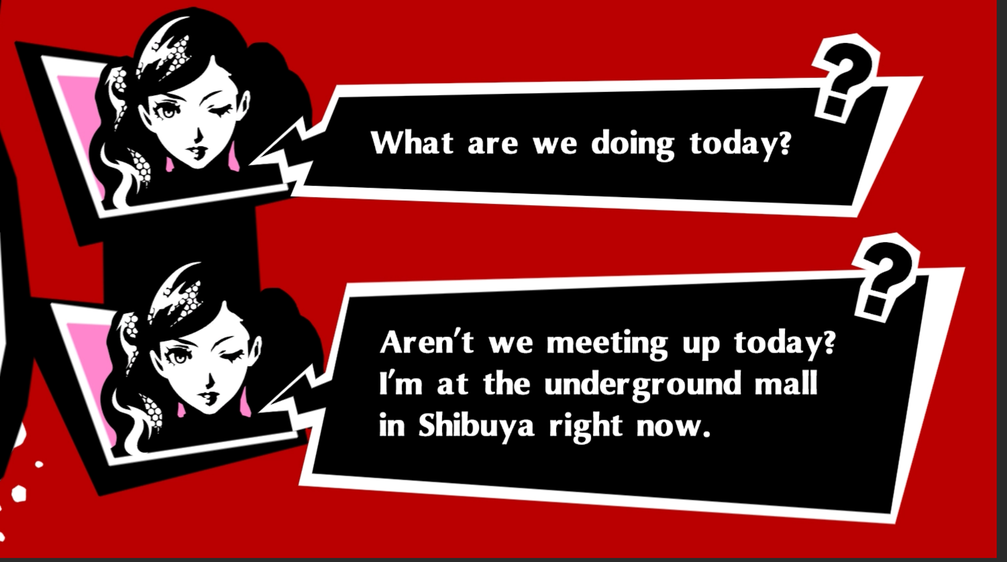


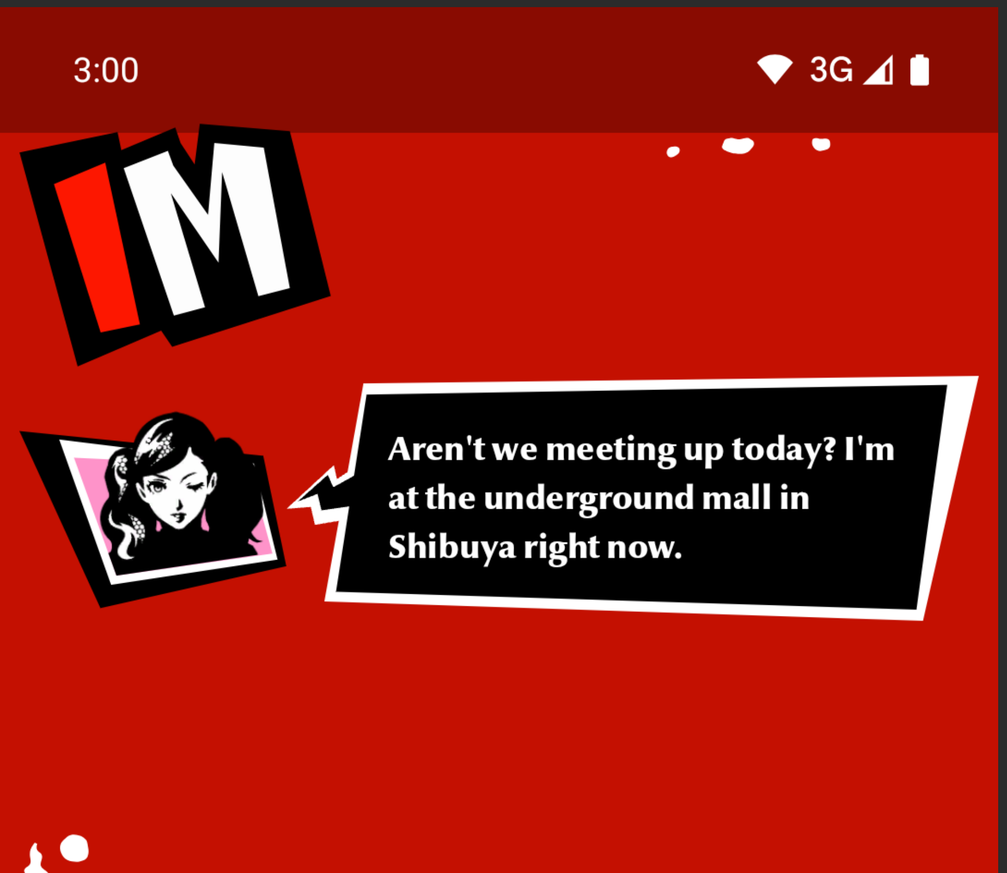

```
Row {
  Avatar()
  TextBox(
    modifier = Modifier.offset(x = (-40).dp)
  )
}
```



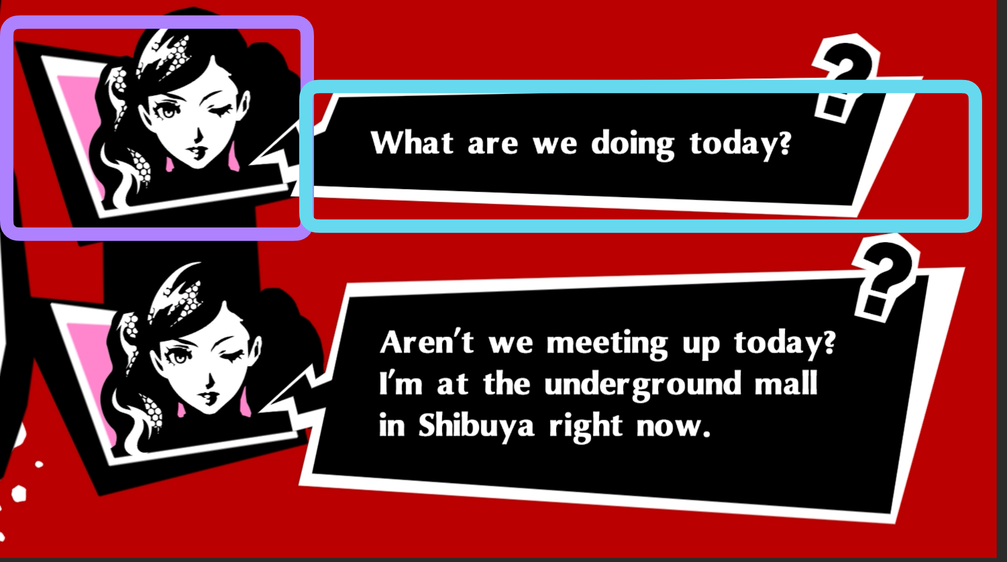


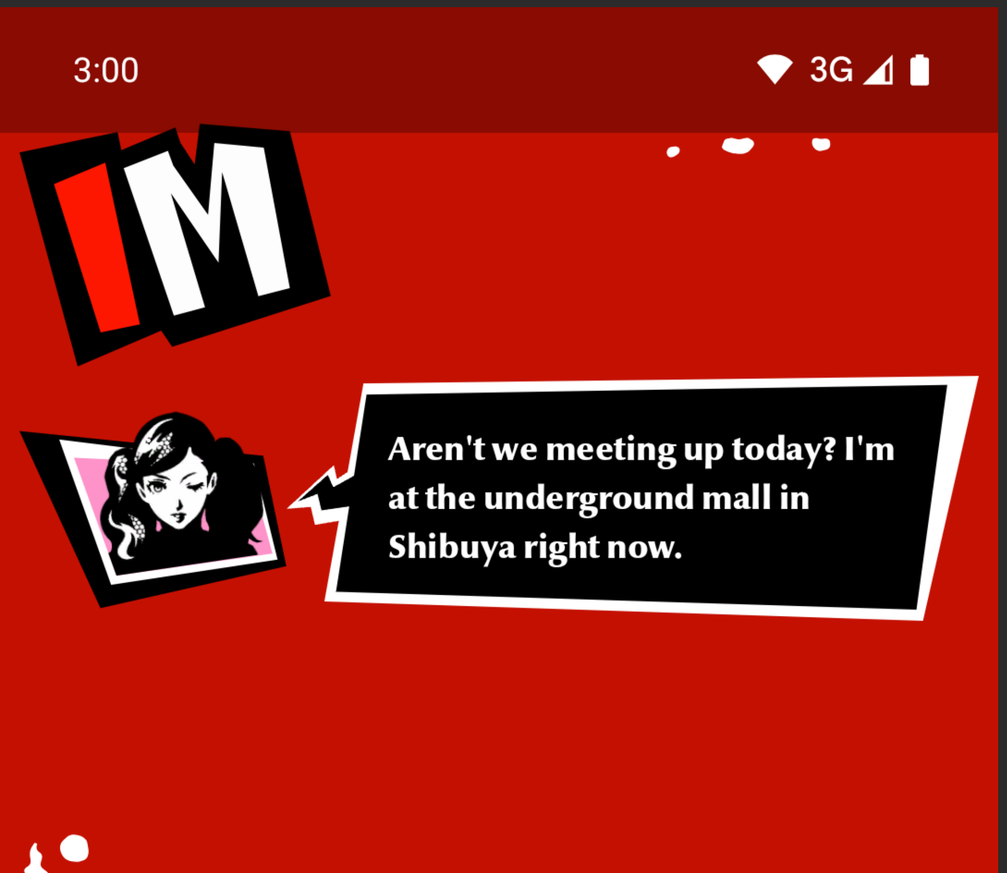

```
Row {
  Avatar()
  TextBox(
    modifier = Modifier.offset(x = (-40).dp)
  )
}
```



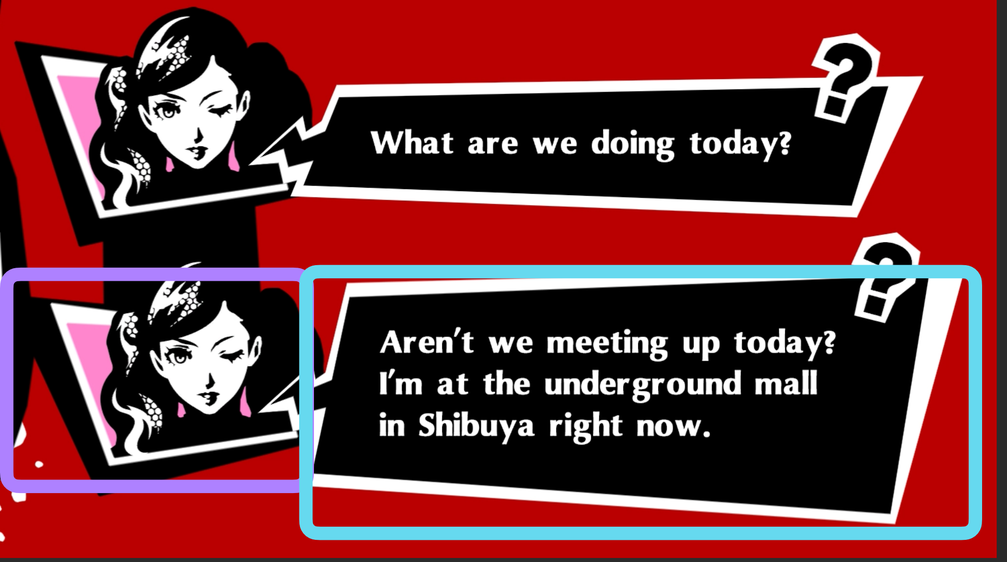


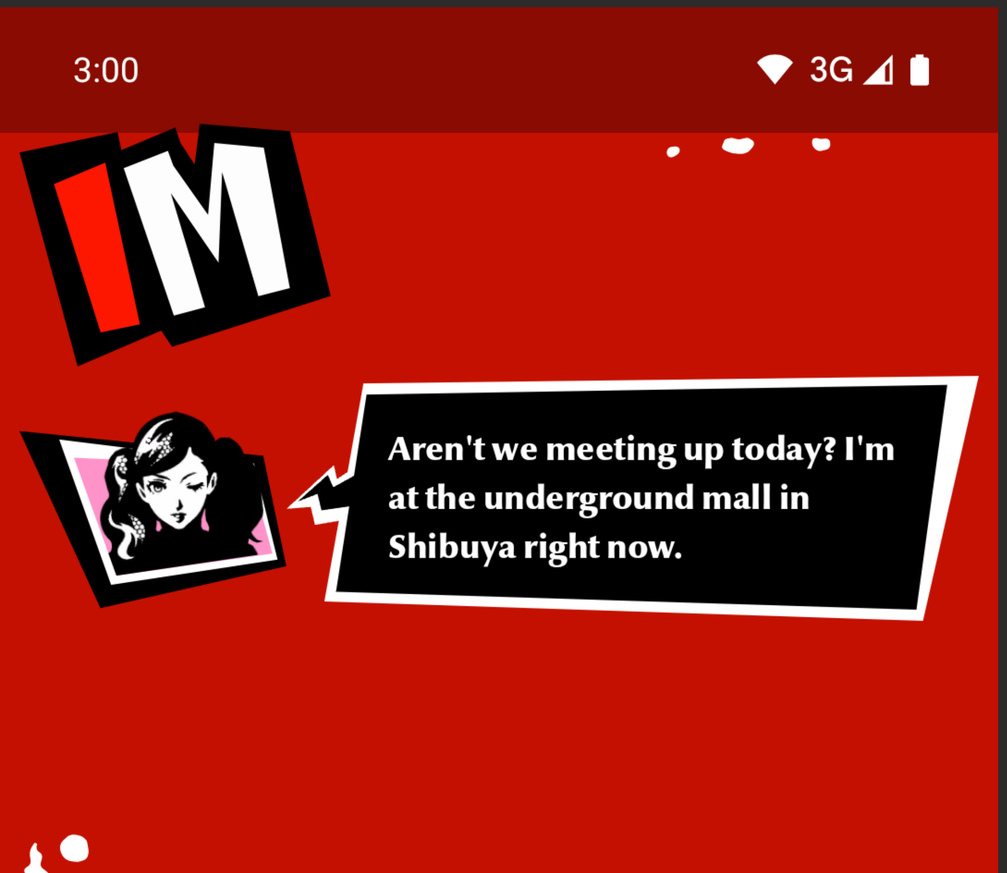

```
Row {
  Avatar()
  TextBox()
}
```






```
Row {
  Avatar()
  TextBox()
}
```



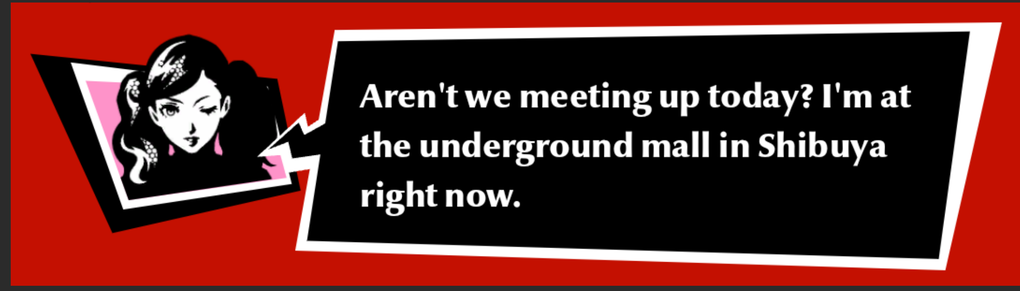



```
Row {
  Avatar()
  TextBox()
}
```

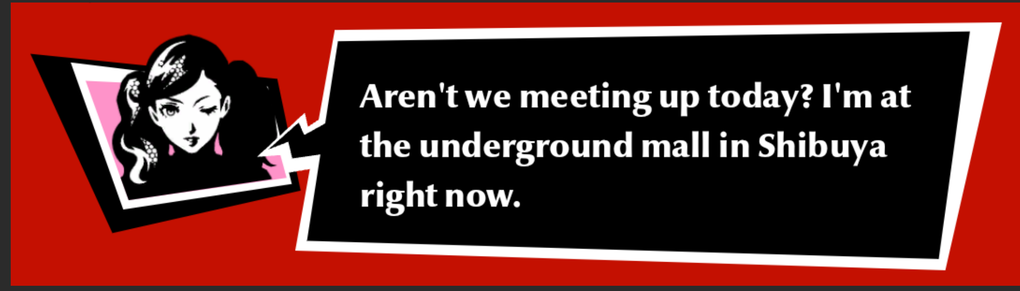
```
Layout(
  content = {
    // Composables...
  }
) { measurables, constraints →

}
```

```
Layout(
  content = {
    // Composables...
  }
) { measurables, constraints →

  val placeables = measurables.map {
    it.measure(constraints)
  }
}
```

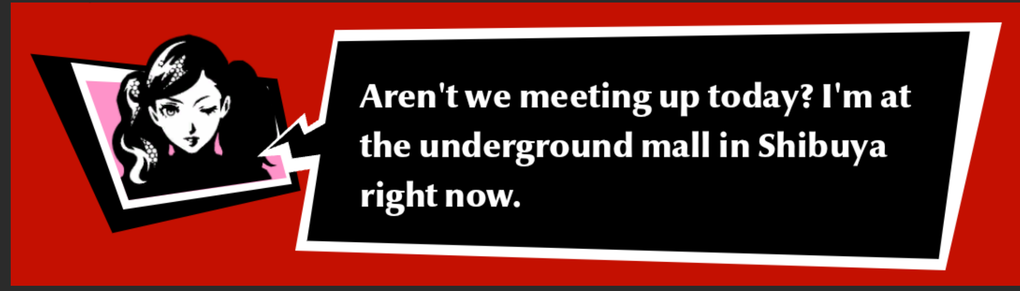
```
Layout(
  content = {
    // Composables...
  }
) { measurables, constraints →

  val placeables = measurables.map {
    it.measure(constraints)
  }

  layout(width, height) {
    placeables.forEach { it.place(x, y) }
  }
}
```



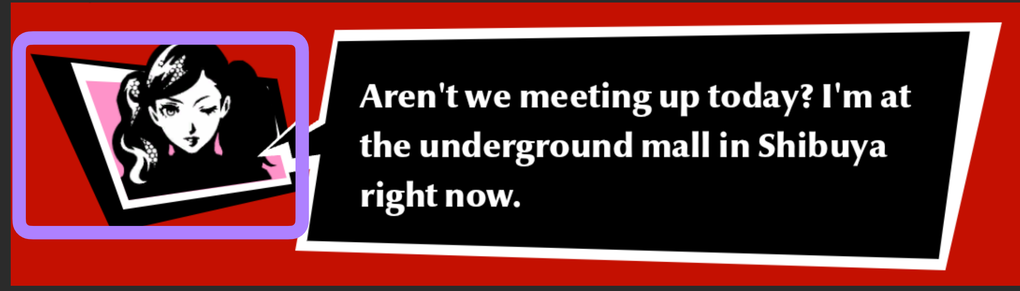

```
Layout(
  content = {
    // Composables...
  }
) { measurables, constraints →

  val placeables = measurables.map {
    it.measure(constraints)
  }

  layout(width, height) {
    placeables.forEach { it.place(x, y) }
  }
}
```



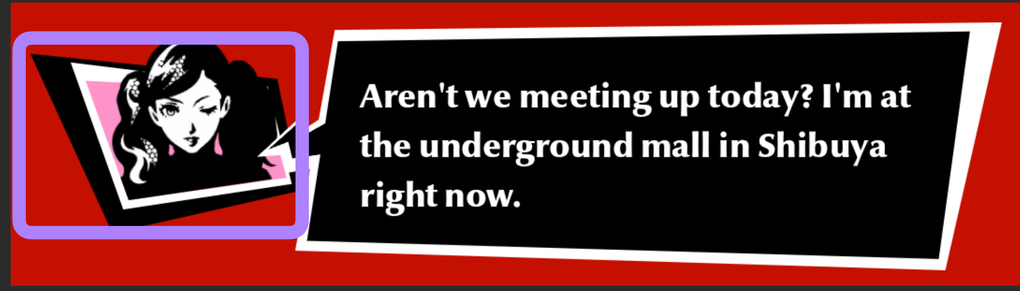

```
Layout(
  content = {
    Avatar()
    TextBox()
  }
) { (avatarMeasurable, textMeasurable), constraints →


}
```



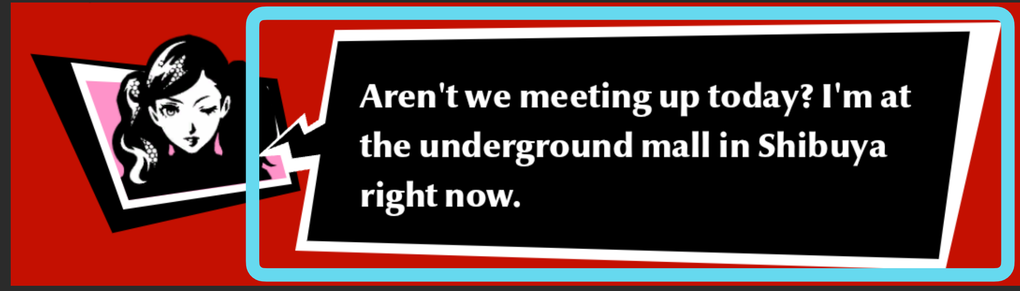

```
Layout(
  content = {
    Avatar()
    TextBox()
  }
) { (avatarMeasurable, textMeasurable), constraints →


}
```



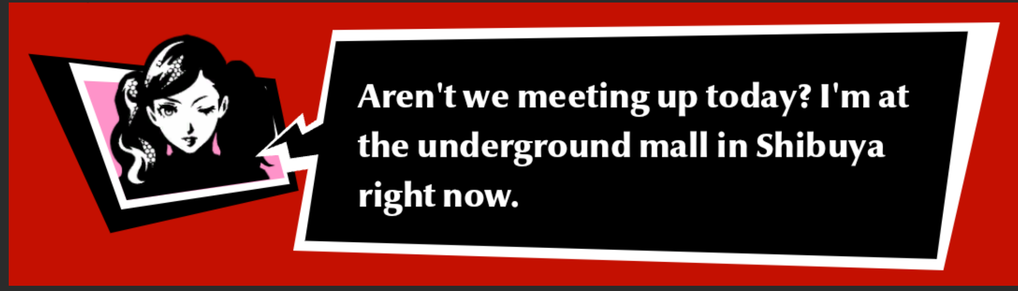

```
Layout(
  content = {
    Avatar()
    TextBox()
  }
) { (avatarMeasurable, textMeasurable), constraints →

  val avatarPlaceable = avatarMeasurable.measure(constraints)
}
```



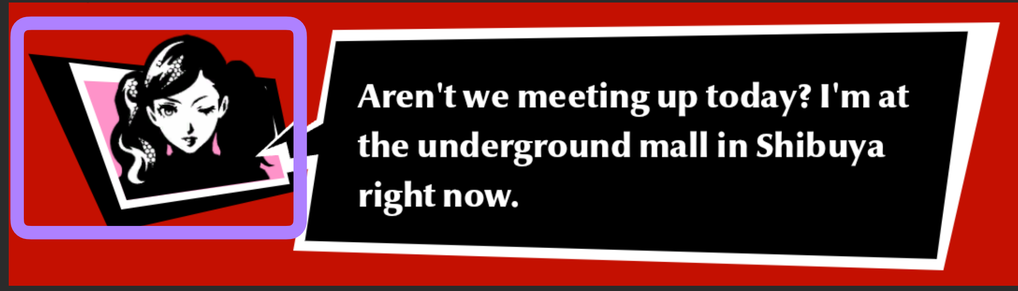
```
Layout(
  content = {
    Avatar()
    TextBox()
  }
) { (avatarMeasurable, textMeasurable), constraints →

  val avatarPlaceable = avatarMeasurable.measure(constraints)
  val textMaxWidth = constraints.maxWidth - avatarPlaceable.width + 18.dp
}
```




```
Layout(
  content = {
    Avatar()
    TextBox()
  }
) { (avatarMeasurable, textMeasurable), constraints →

  val avatarPlaceable = avatarMeasurable.measure(constraints)
  val textMaxWidth = constraints.maxWidth - avatarPlaceable.width + 18.dp
  val textConstraints = constraints.copy(maxWidth = textMaxWidth)
  val textPlaceable = textMeasurable.measure(textConstraints)
}
```

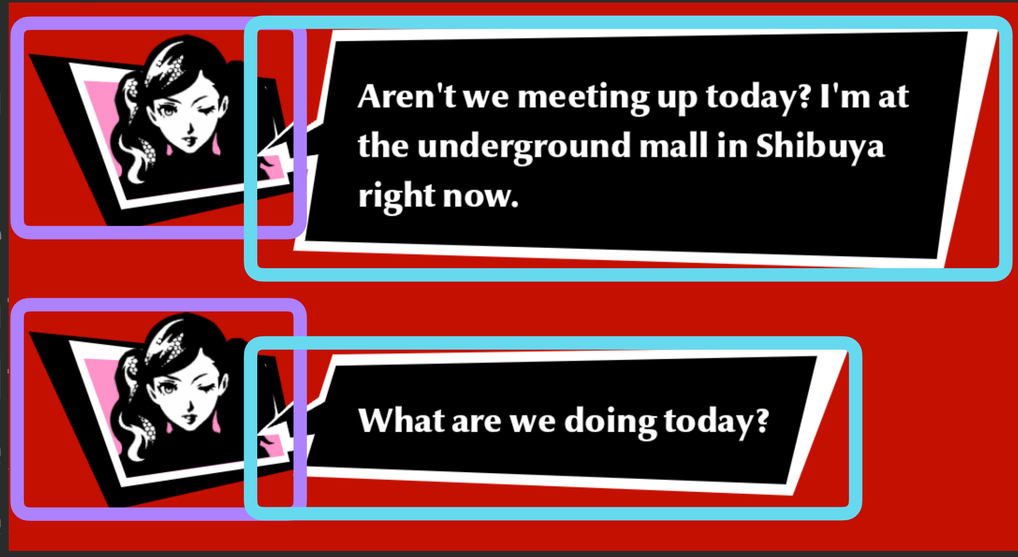



```
Layout(
  content = {
    Avatar()
    TextBox()
  }
) { (avatarMeasurable, textMeasurable), constraints →

  val avatarPlaceable = avatarMeasurable.measure(constraints)
  val textMaxWidth = constraints.maxWidth - avatarPlaceable.width + 18.dp
  val textConstraints = constraints.copy(maxWidth = textMaxWidth)
  val textPlaceable = textMeasurable.measure(textConstraints)
  val width = avatarPlaceable.width + textPlaceable.width - 18.dp
  val height = maxOf(avatarPlaceable.height, textPlaceable.height)
}
```




```
Layout(
  content = {
    Avatar()
    TextBox()
  }
) { (avatarMeasurable, textMeasurable), constraints →

  val avatarPlaceable = avatarMeasurable.measure(constraints)
  val textMaxWidth = constraints.maxWidth - avatarPlaceable.width + 18.dp
  val textConstraints = constraints.copy(maxWidth = textMaxWidth)
  val textPlaceable = textMeasurable.measure(textConstraints)
  val width = avatarPlaceable.width + textPlaceable.width - 18.dp
  val height = maxOf(avatarPlaceable.height, textPlaceable.height)

  layout(width, height) {
  ● avatarPlaceable.place(0, 0)
  }
}
```

```
    }
) { (avatarMeasurable, textMeasurable), constra

    val avatarPlaceable = avatarMeasurable.measur
    val textMaxWidth = constraints.maxWidth - ava
    val textConstraints = constraints.copy(maxWid
    val textPlaceable = textMeasurable.measure(te
    val width = avatarPlaceable.width + textPlace
    val height = maxOf(avatarPlaceable.height, textPlaceable.height)

    layout(width, height) {
      avatarPlaceable.place(0, 0)
      val textBoxX = avatarPlaceable.width - textBoxOverlap
      val textBoxY =
        if (textPlaceable.height > avatarPlaceable.height) 0
        else height - textPlaceable.height

      textPlaceable.place(textBoxX, textBoxY)
    }
}
```




```
LazyColumn(
  verticalArrangement = Arrangement.spacedBy(16.dp),
  contentPadding = WindowInsets.systemBars
    .add(WindowInsets(top = 100.dp, bottom = 100.dp))
    .asPaddingValues(),
  modifier = Modifier.fillMaxSize(),
)
```


22
THURSDAY
Afternoon
s comin', it'll be today.
I'm so nervous...
Same... I can't focus on class at all.
And how's that any different than usual?
?
L Scroll
A Next
FFWD
Auto
©ATLUS ©SEGA

Centres itself to the first item
最初のアイテムに中央に揃える

Alternates left and right by a random offset
左右交互にランダムなオフセット

Centres itself to the last item
最後のアイテムに中央に揃える

Jumps to a final position before animating
アニメーションの前に最終位置にジャンプする

Modelling data
データモデル

```
enum class Sender(
  @DrawableRes val image: Int,
  val color: Color,
) {
  Kasumi(R.drawable.kasumi, Color(0xFFD53359)),
  Ryuji(R.drawable.ryuji, Color(0xFFF0EA40)),
  Ann(R.drawable.ann, Color(0xFFFE93C9)),
}
```

```
data class Message(
  val sender: Sender,
  val text: String,
)
```

Modelling data
データモデル

```
data class LineCoordinates(
    val leftPoint: Offset,
    val rightPoint: Offset,
)
```

```
data class Entry(
    val message: Message,
    val lineCoordinates: LineCoordinates,
)
```

Where to store these data?
どこに保管する？

ViewModel is one option

UI state is another option

```
val scrollState = rememberScrollState()

val pagerState = rememberPagerState()

val textState = rememberTextFieldState()
```

ViewModel

Current list of messages
今までのメッセージがリストに

UI state

Entries

Background line coordinates
背景線コーディネート

Item animation progress
アイテムアニメーションの
進行状況

ViewModel → UI state → LazyColumn

```kotlin
@Composable
fun rememberTranscriptState(): TranscriptState {
  val density = LocalDensity.current
  val coroutineScope = rememberCoroutineScope()
  return remember(density) { TranscriptState(density, coroutineScope) }
}
```

```
@Composable
fun rememberTranscriptState(): TranscriptState {
  val density = LocalDensity.current
  val coroutineScope = rememberCoroutineScope()
  return remember(density) { TranscriptState(density, coroutineScope) }
}
```

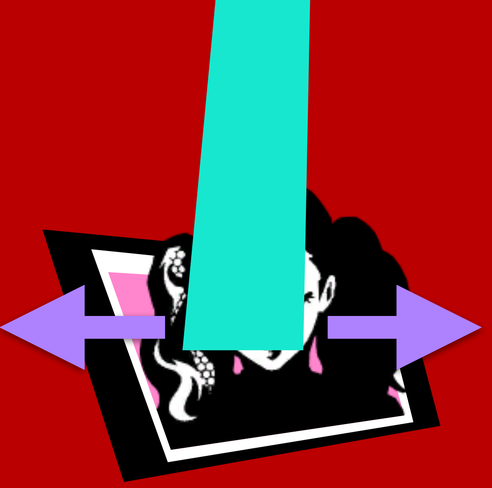
```
@Stable
class TranscriptState internal constructor(
  private val density: Density,
  private val coroutineScope: CoroutineScope,
) {
  private val _entries = mutableStateOf<List<Entry>>(emptyList())
  val entries: List<Entry> by _entries
}
```


```
val entries: List<Entry> by _entries
```


```
val entries: List<Entry> by _entries
```


```
val entries: List<Entry> by _entries
```

```
data class Entry(
  val message: Message,
  val lineCoordinates: LineCoordinates,
)


val leftX = (AvatarSize.width / 2f) - (lineWidth / 2f)
val rightX = leftX + lineWidth

val y = AvatarSize.height / 2f

val lineCoordinates = LineCoordinates(
  leftPoint = Offset(leftX, y),
  rightPoint = Offset(rightX, y),
)
```



```
data class Entry(
  val message: Message,
  val lineCoordinates: LineCoordinates,
)
```

```
val leftX = (AvatarSize.width / 2f) - (lineWidth / 2f)
val rightX = leftX + lineWidth

val y = AvatarSize.height / 2f

val lineCoordinates = LineCoordinates(
  leftPoint = Offset(leftX, y),
  rightPoint = Offset(rightX, y),
)
```

```
data class Entry(
  val message: Message,
  val lineCoordinates: LineCoordinates,
)
```

```
val leftX = (AvatarSize.width / 2f) - (lineWidth / 2f)
val rightX = leftX + lineWidth

val y = AvatarSize.height / 2f

val direction = if (message.index % 2 == 0) 1f else -1f
val horizontalShift = randomBetween(MinShift, MaxShift) * direction

val lineCoordinates = LineCoordinates(
  leftPoint = Offset(leftX + horizontalShift, y),
  rightPoint = Offset(leftX + horizontalShift, y),
)
```

```
data class Entry(
  val message: Message,
  val lineCoordinates: LineCoordinates,
)


val leftX = (AvatarSize.width / 2f) - (lineWidth / 2f)
val rightX = leftX + lineWidth

val y = AvatarSize.height / 2f

val direction = if (message.index % 2 == 0) 1f else -1f
val horizontalShift = randomBetween(MinShift, MaxShift) * direction

val lineCoordinates = LineCoordinates(
  leftPoint = Offset(leftX + horizontalShift, y),
  rightPoint = Offset(leftX + horizontalShift, y),
)
```


```
val listState = rememberLazyListState()

LazyColumn(state = listState)
```


```
val listState = rememberLazyListState()
val transcriptState = rememberTranscriptState()
val entries = transcriptState.entries

LazyColumn(state = listState)
```


```
val listState = rememberLazyListState()
val transcriptState = rememberTranscriptState()
val entries = transcriptState.entries

LazyColumn(state = listState) {
  items(entries) { entry →
    Entry(entry)
  }
}
```


```
val listState = rememberLazyListState()
val transcriptState = rememberTranscriptState()
val entries = transcriptState.entries

BackgroundLine(listState, entries)

LazyColumn(state = listState) {
  items(entries) { entry →
    Entry(entry)
  }
}
```


```
@Composable
private fun BackgroundLine(
  entries: List<Entry>,
  listState: LazyListState,
) {

  val visibleItemInfos = listState.layoutInfo.visibleItemsInfo
}


interface LazyListItemInfo {

  /**
   * The main axis offset of the item in pixels.
   * It is relative to the start of the lazy list container.
   */
  val offset: Int

}
```


```kotlin
@Composable
private fun BackgroundLine(
  entries: List<Entry>,
  listState: LazyListState,
) {

  val visibleItemInfos = listState.layoutInfo.visibleItemsInfo
}


interface LazyListItemInfo {

  /**
   * The main axis offset of the item in pixels.
   * It is relative to the start of the lazy list container.
   */
  val offset: Int

}
```


```kotlin
@Composable
private fun BackgroundLine(
  entries: List<Entry>,
  listState: LazyListState,
) {

  val visibleItemInfos = listState.layoutInfo.visibleItemsInfo

  Canvas(modifier = Modifier.fillMaxSize()) {
    for (info in visibleItemInfos) {
      drawPath(getPoints(info, entries[info.index]), Color.Black)
    }
  }
}
```

3:00 3G

IM

We have to find them tomorrow for sure. This is the only lead we have right now.

Yes. It is highly likely that this part-time solicitor is somehow related to the mafia.

If we tail him, he may lead us straight back to his boss.

He talked to Iida and Nishiyama over at Central Street, right?

Indeed, it seems that is where our target waits. But then... who should be the one to go?

Morgana, I choose you.

That's not a bad ide
nine lives, right? Mo
spare one for this.

Next

3:00 3G

IM

We have to find them tomorrow for sure. This is the only lead we have right now.

Yes. It is highly likely that this part-time solicitor is somehow related to the mafia.

If we tail him, he may lead us straight back to his boss.

He talked to Iida and Nishiyama over at Central Street, right?

Indeed, it seems that is where our target waits. But then... who should be the one to go?

Morgana, I choose you.

Next

Total height

Overlap height


3:00
3G
IM
We have to find them tomorrow for sure. This is the only lead we have right now.
Yes. It is highly likely that this part-time solicitor is somehow related to the mafia.
If we tail him, he may lead us straight back to his boss.
He talked to Iida and Nishiyama over at Central Street, right?
Indeed, it seems that is where our target waits. But then... who should be the one to go?
Morgana, I choose you.
Next

3:00

3G

IM

We have to find them tomorrow for sure. This is the only lead we have right now.

Yes. It is highly likely that this part-time solicitor is somehow related to the mafia.

If we tail him, he may lead us straight back to his boss.

He talked to Iida and Nishiyama over at Central Street, right?

Indeed, it seems that is where our target waits. But then... who should be the one to go?

Morgana, I choose you.

Next


```
fun Modifier.drawConnectingLine(entry1: Entry, entry2: Entry?): Modifier {

}
```


```
fun Modifier.drawConnectingLine(entry1: Entry, entry2: Entry?): Modifier {
  if (entry2 == null) return this

  return drawWithCache {

    onDrawBehind {

    }
  }
}
```

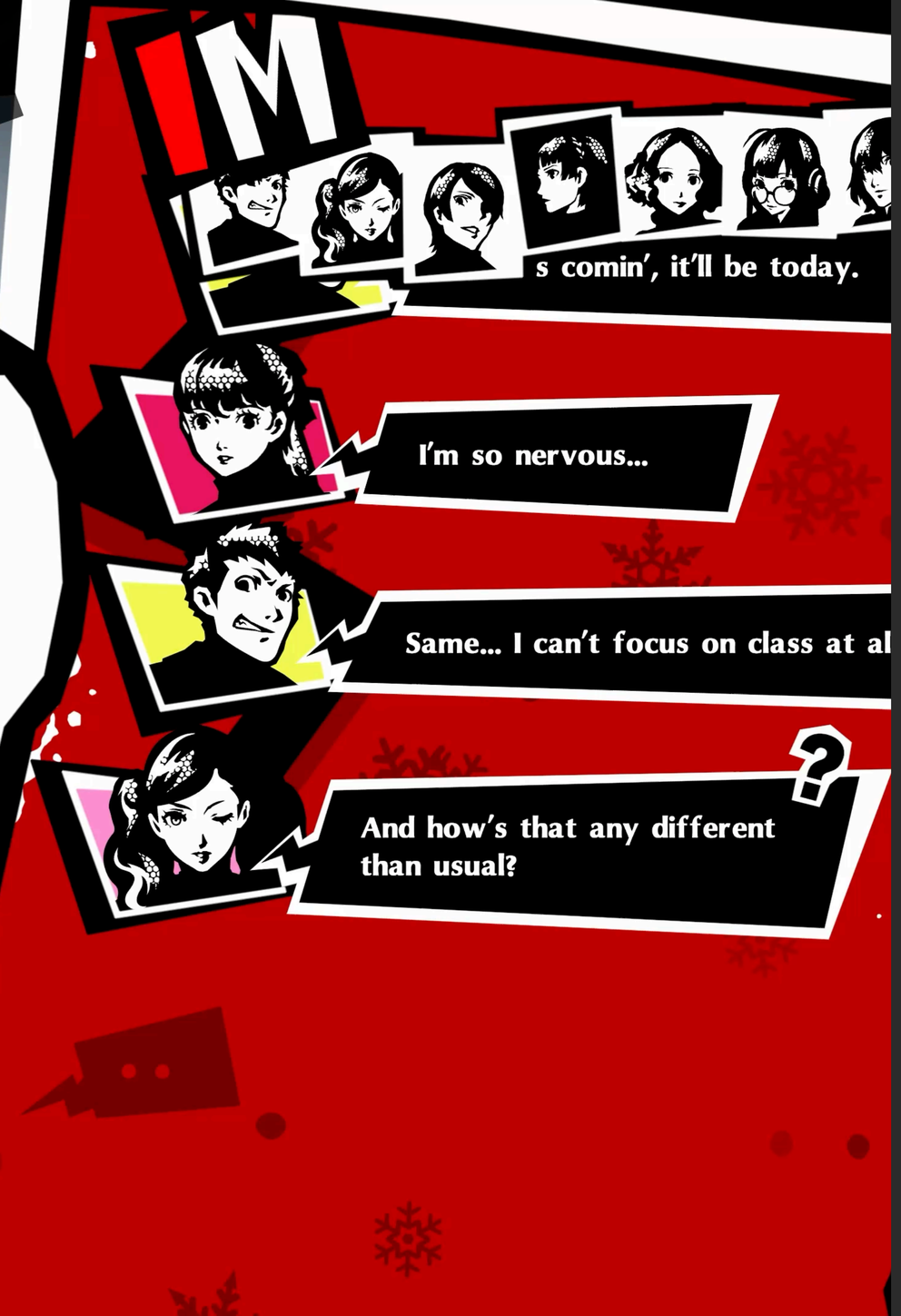

```
fun Modifier.drawConnectingLine(entry1: Entry, entry2: Entry?): Modifier {
  if (entry2 == null) return this

  return drawWithCache {
    val linePath = Path()
    val topLeft = entry1.lineCoordinates.leftPoint
    val topRight = entry1.lineCoordinates.rightPoint

    val bottomLeft = entry2.lineCoordinates.leftPoint + bottomOffset
    val bottomRight = entry2.lineCoordinates.rightPoint + bottomOffset

    onDrawBehind {

    }
  }
}
```


```
fun Modifier.drawConnectingLine(entry1: Entry, entry2: Entry?): Modifier {
  if (entry2 == null) return this

  return drawWithCache {
    val linePath = Path()
    val topLeft = entry1.lineCoordinates.leftPoint
    val topRight = entry1.lineCoordinates.rightPoint

    val bottomLeft = entry2.lineCoordinates.leftPoint + bottomOffset
    val bottomRight = entry2.lineCoordinates.rightPoint + bottomOffset

    onDrawBehind {
      with(linePath) {
        rewind()
        moveTo(topLeft.x, topLeft.y)
        lineTo(topRight.x, topRight.y)
        lineTo(bottomRight.x, bottomRight.y)
        lineTo(bottomLeft.x, bottomLeft.y)
        close()
      }

      drawPath(linePath, Color.Black)
    }
  }
}
```

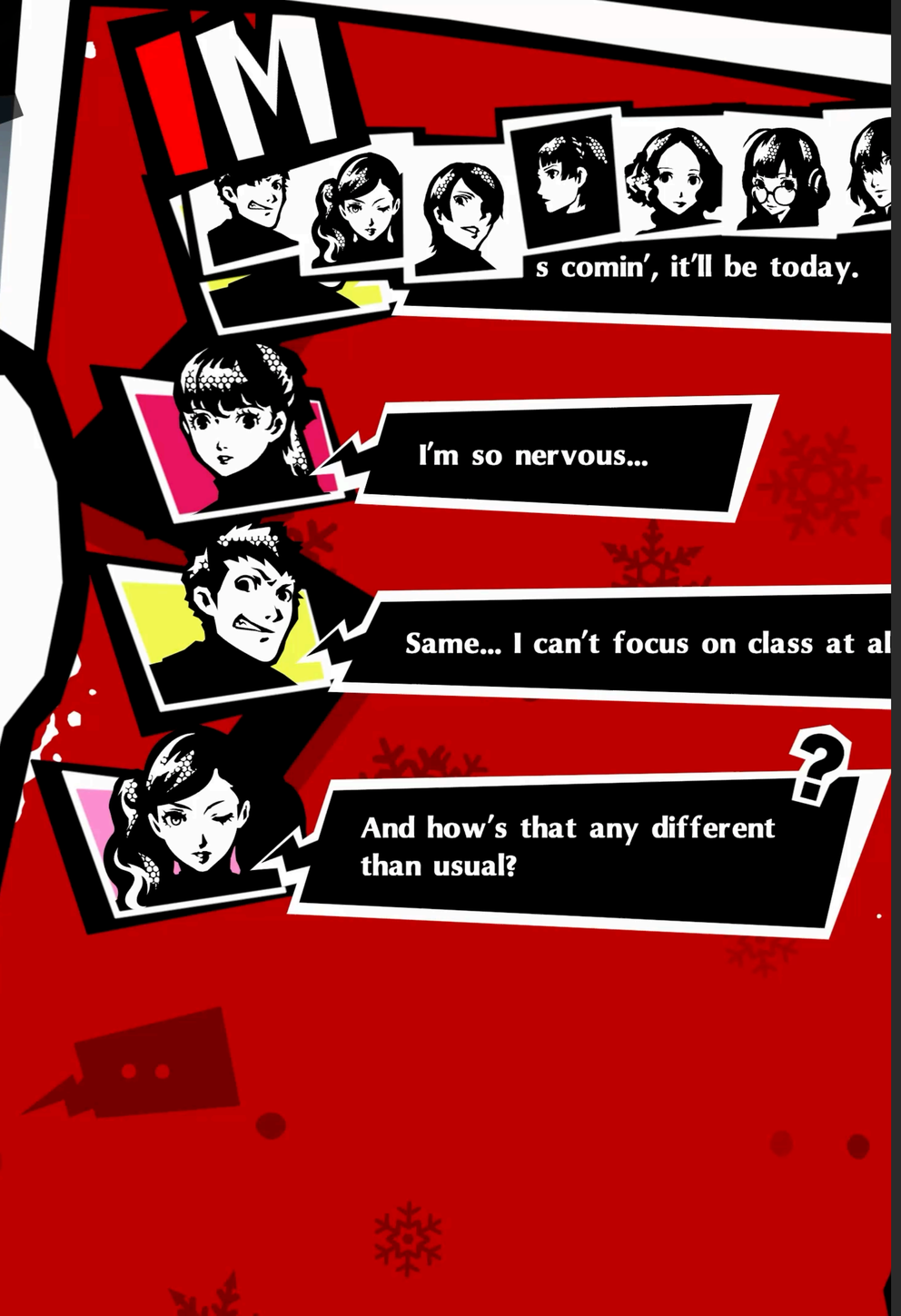

```
    val bottomRight = entry2.lineCoordinates.rightPoint + bottomOffset

    val shadowPaint = Paint().apply {
      color = Color.Black
      alpha = 0.5f
      asFrameworkPaint().maskFilter = BlurMaskFilter(4.dp.toPx(), NORMAL)
    }

    onDrawBehind {
      with(linePath) {
        rewind()
        moveTo(topLeft.x, topLeft.y)
        lineTo(topRight.x, topRight.y)
        lineTo(bottomRight.x, bottomRight.y)
        lineTo(bottomLeft.x, bottomLeft.y)
        close()
      }

      translate(top = 16.dp.toPx()) {
        drawIntoCanvas {
          it.drawPath(linePath, shadowPaint)
        }
      }

      drawPath(linePath, Color.Black)
    }
  }
```


3:00
3G
We have to find them tomorrow for sure. This is the only lead we have right now.
Yes. It is highly likely that this part-time solicitor is somehow related to the mafia.
If we tail him, he may lead us straight back to his boss.
He talked to Iida and Nishiyama over at Central Street, right?
Indeed, it seems that is where our target waits. But then... who should be the one to go?
Morgana, I choose you.
That's not a bad ide
nine lives, right? Mo
spare one for this.
Next

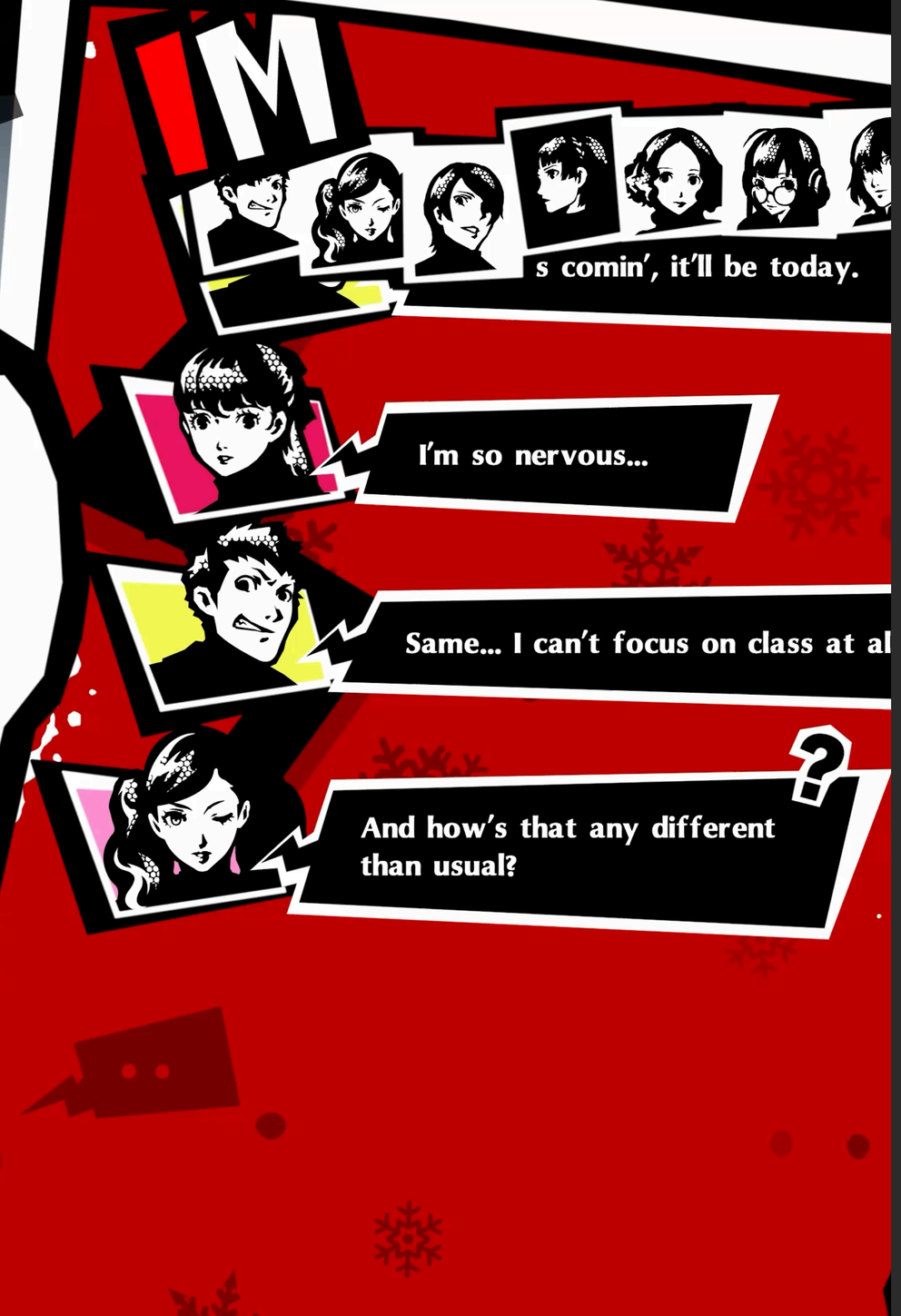
Animations



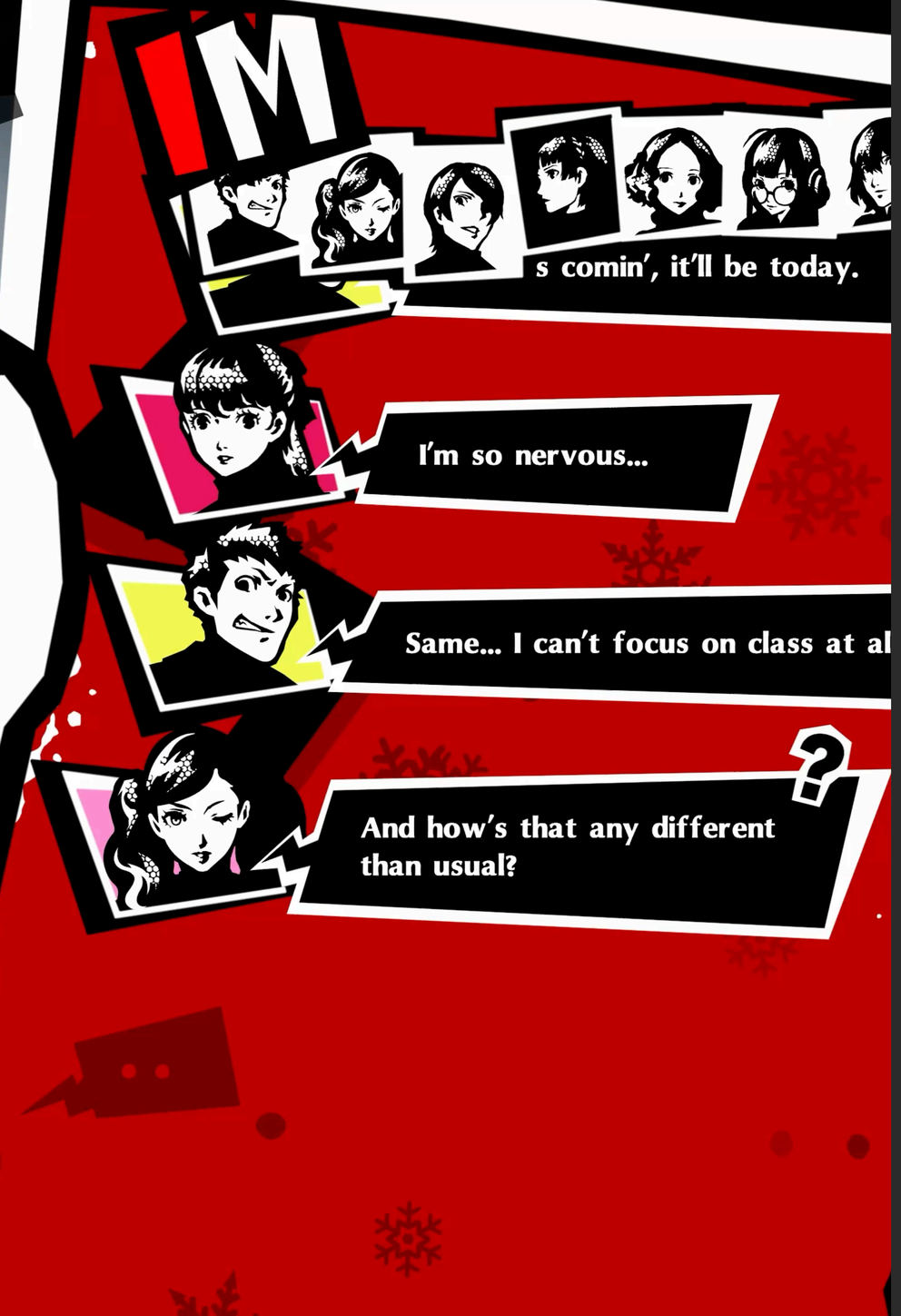

```
data class Entry(
  val message: Message,
  val lineCoordinates: LineCoordinates,
)
```



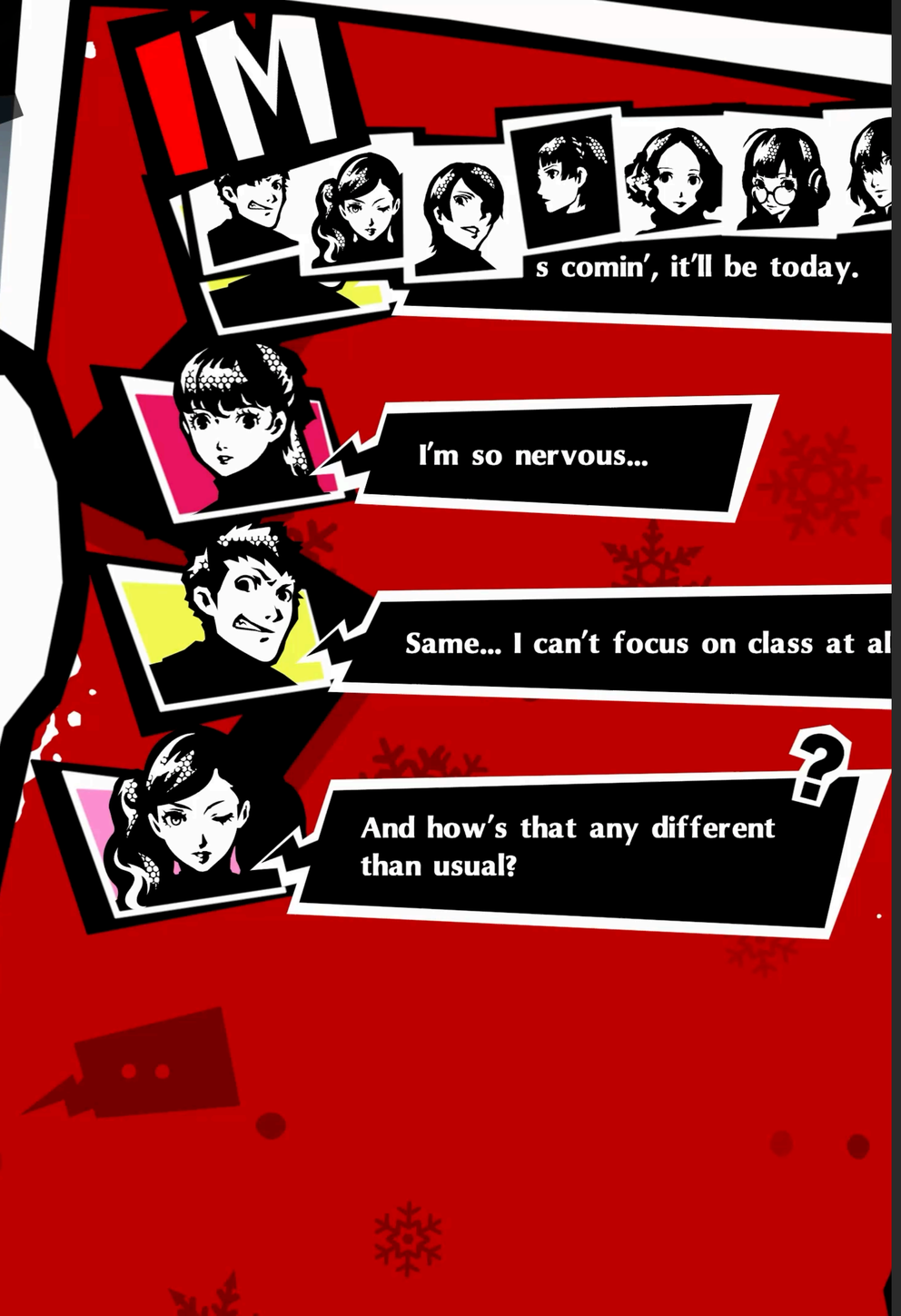

```
data class Entry(
  val message: Message,
  val lineCoordinates: LineCoordinates,

  val lineProgress: State<Float>,
)
```



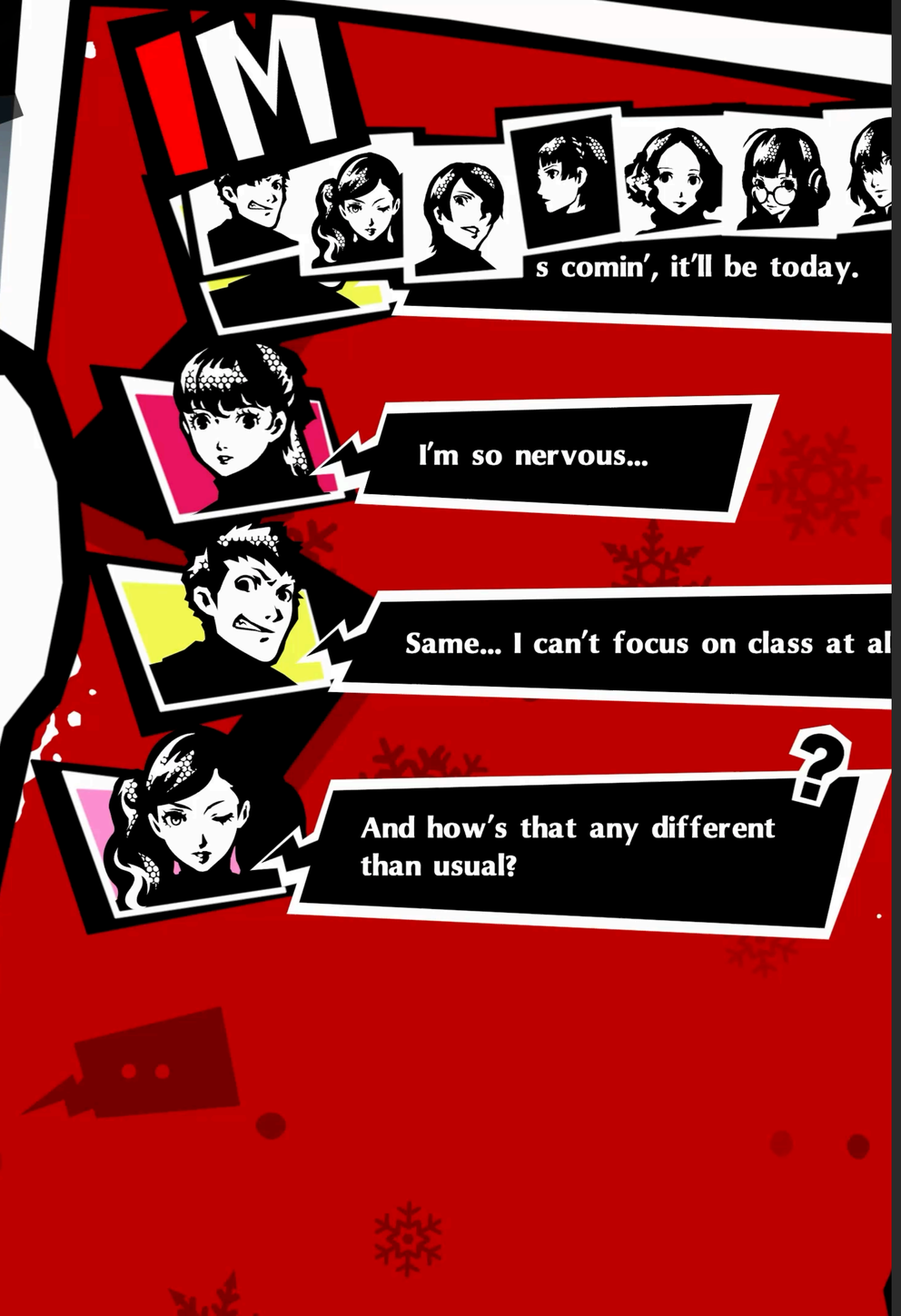

```
data class Entry(
  val message: Message,
  val lineCoordinates: LineCoordinates,

  val lineProgress: State<Float>,
  val avatarBackgroundScale: State<Float>,
  val avatarForegroundScale: State<Float>,
)
```




```
data class Entry(
  val message: Message,
  val lineCoordinates: LineCoordinates,

  val lineProgress: State<Float>,
  val avatarBackgroundScale: State<Float>,
  val avatarForegroundScale: State<Float>,
  val messageHorizontalScale: State<Float>,
  val messageVerticalScale: State<Float>,
  val messageTextAlpha: State<Float>,
)
```



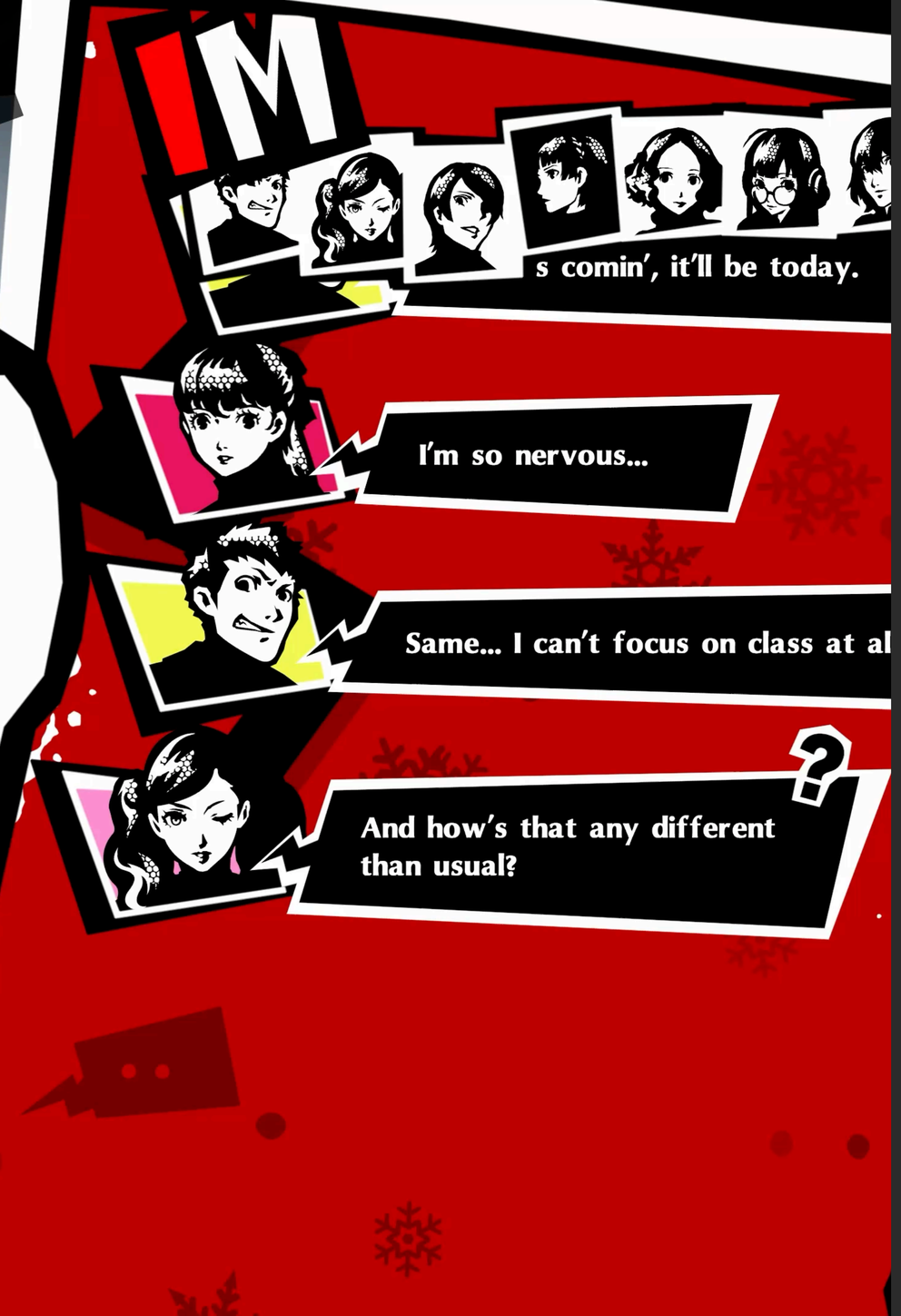

```
avatarBackgroundScale = Animatable(initialValue = 0.6f)
```



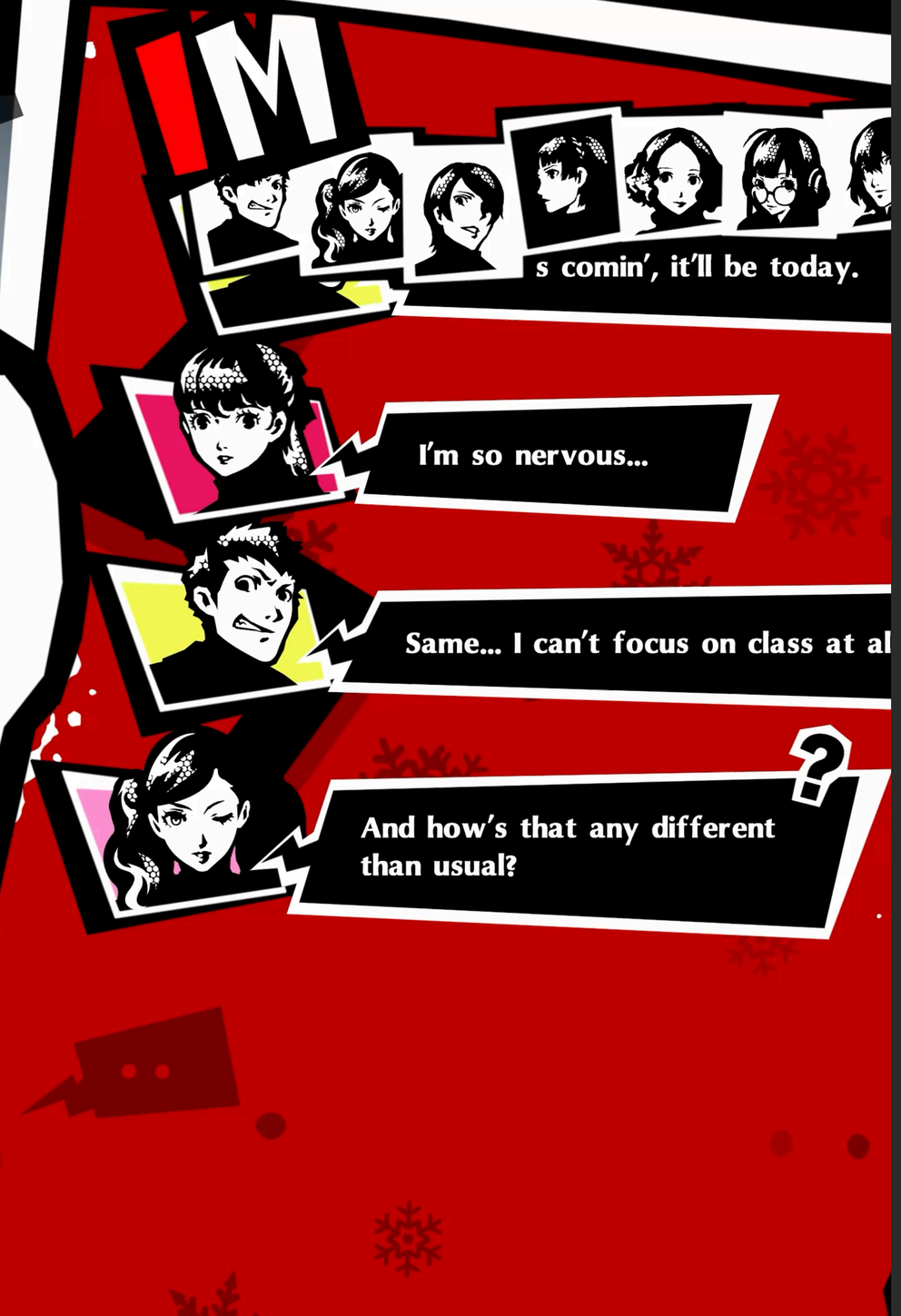

```
avatarBackgroundScale = Animatable(initialValue = 0.6f)
  .apply {
    coroutineScope.launch {
      animateTo(
        targetValue = 1f,
        animationSpec = tween(
          durationMillis = 300,
          easing = EaseOutBack,
        ),
      )
    }
  }
```

```
avatarBackgroundScale: Animatable<Float, AnimationVector1D> = Animatable(initialValue = 0.6f)
  .apply {
    coroutineScope.launch {
      animateTo(
        targetValue = 1f,
        animationSpec = tween(
          durationMillis = 300,
          easing = EaseOutBack,
        ),
      )
    }
  }
```

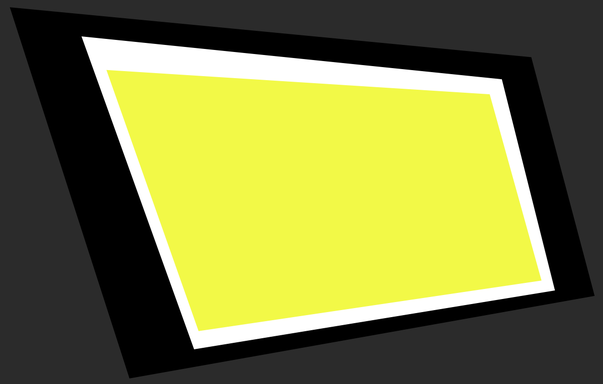
```
avatarBackgroundScale: State<Float> = Animatable(initialValue = 0.6f)
  .apply {
    coroutineScope.launch {
      animateTo(
        targetValue = 1f,
        animationSpec = tween(
          durationMillis = 300,
          easing = EaseOutBack,
        ),
      )
    }
  }
  .asState()
```



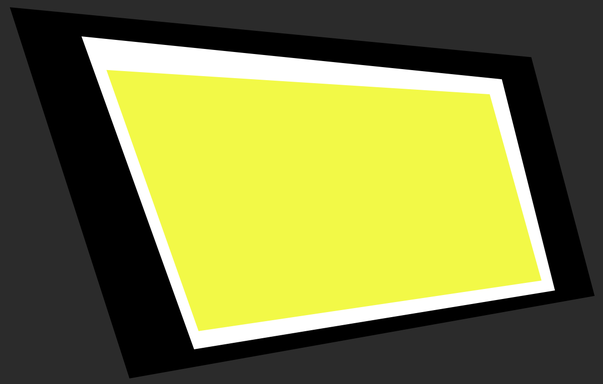

```
avatarBackgroundScale = Animatable(initialValue = 0.6f)
  .apply {
    // ...
  }

avatarForegroundScale = Animatable(initialValue = 0.0f)
  .apply {
    coroutineScope.launch {
      delay(160L)
      snapTo(0.8f)
      animateTo(
        targetValue = 1f,
        animationSpec = tween(
          durationMillis = 150,
          easing = EaseOutBack,
        ),
      )
    }
  }
  .asState()
```




```
avatarBackgroundScale = Animatable(initialValue = 0.6f)
  .apply {
    // ...
  }

avatarForegroundScale = Animatable(initialValue = 0.0f)
  .apply {
    coroutineScope.launch {
      delay(160L)
      snapTo(0.8f)
      animateTo(
        targetValue = 1f,
        animationSpec = tween(
          durationMillis = 150,
          easing = EaseOutBack,
        ),
      )
    }
  }
  .asState()
```



```
Box(
  modifier = Modifier
    .size(AvatarSize)
    .drawBehind {
      // Previous drawing code
    }
) {

}
```

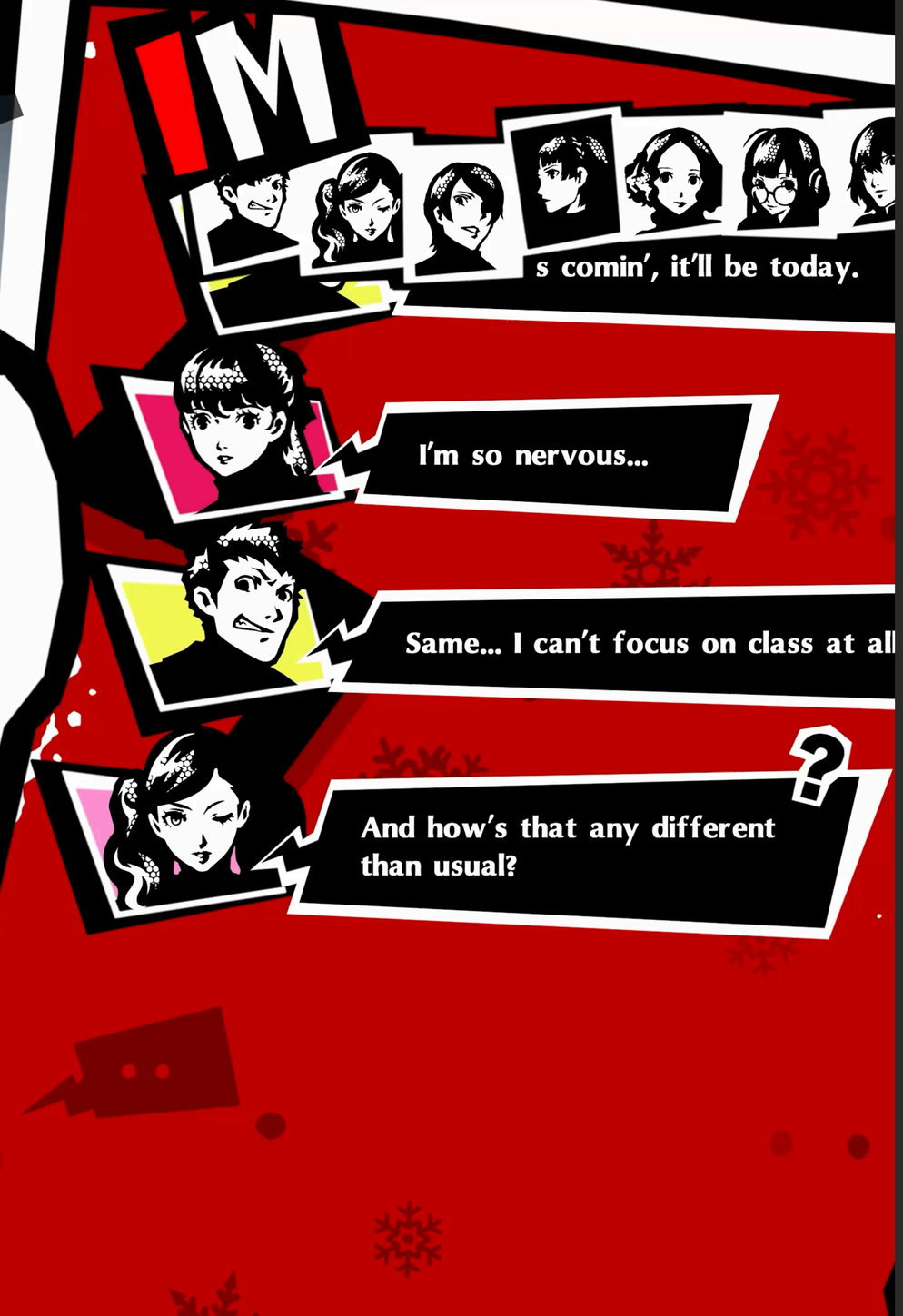


```
Box(
  modifier = Modifier
    .size(AvatarSize)
    .scale(entry.avatarBackgroundScale.value)
    .drawBehind {
      // Previous drawing code
    }
) {

}
```

```
Box(
  modifier = Modifier
    .size(AvatarSize)
    .scale(entry.avatarBackgroundScale.value)
    .drawBehind {
      // Previous drawing code
    }
) {
  Image(
    painter = painterResource(entry.message.sender.image),
  )
}
```

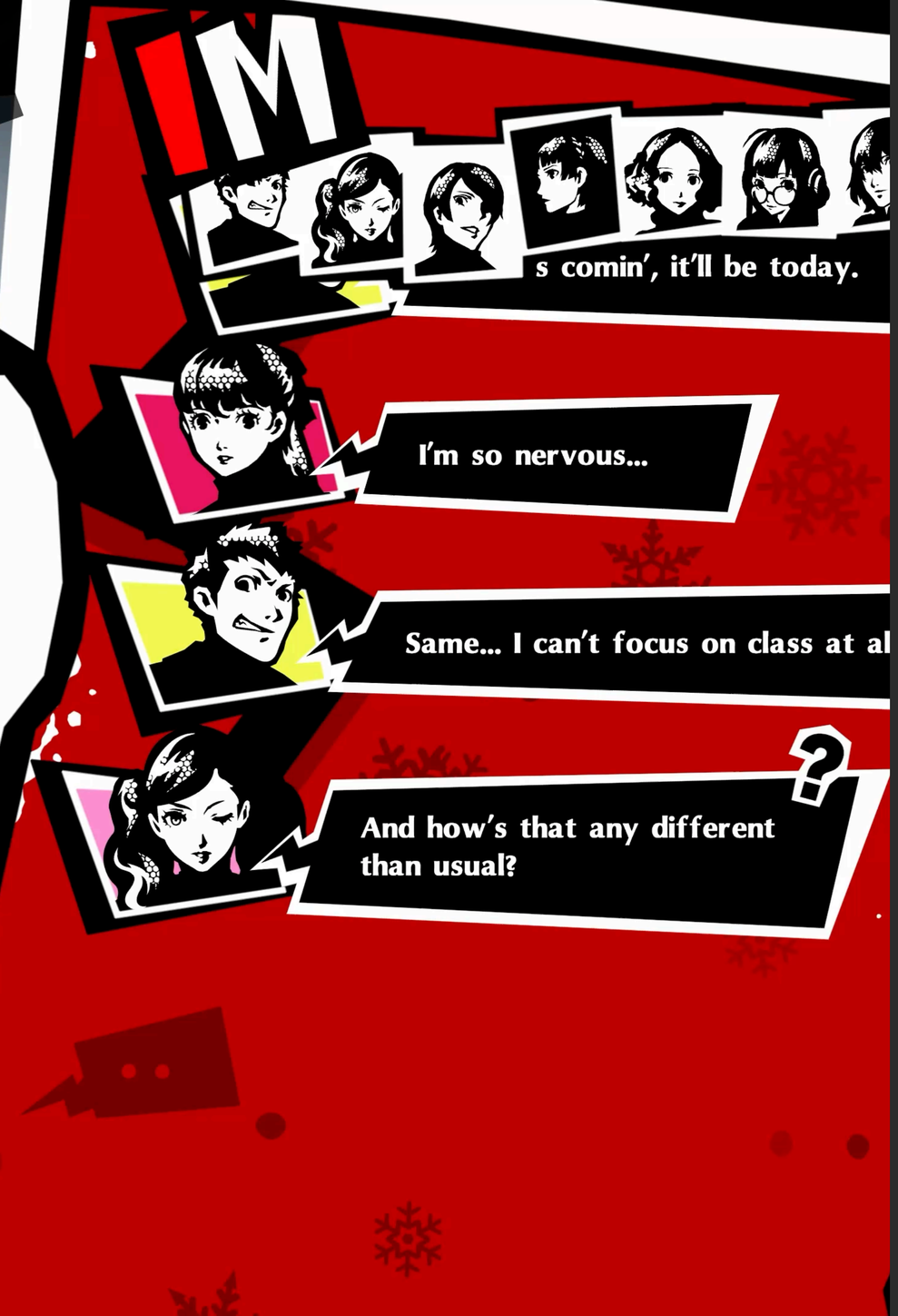
```
Box(
  modifier = Modifier
    .size(AvatarSize)
    .scale(entry.avatarBackgroundScale.value)
    .drawBehind {
      // Previous drawing code
    }
) {
  Image(
    painter = painterResource(entry.message.sender.image),
    modifier = Modifier.graphicsLayer {
      transformOrigin = TransformOrigin(
        pivotFractionX = 0.5f,
        pivotFractionY = 1.15f,
      ),
      scaleX = entry.avatarForegroundScale.value
      scaleY = entry.avatarForegroundScale.value
    }
  )
}
```

```
Box(
  modifier = Modifier
    .size(AvatarSize)
    .scale(entry.avatarBackgroundScale.value)
    .drawBehind {
      // Previous drawing code
    }
) {
  Image(
    painter = painterResource(entry.message.sender.image),
    modifier = Modifier.graphicsLayer {
      transformOrigin = TransformOrigin(
      ● pivotFractionX = 0.5f,
      ● pivotFractionY = 1.15f,
      ),
      scaleX = entry.avatarForegroundScale.value
      scaleY = entry.avatarForegroundScale.value
    }
  )
}
```

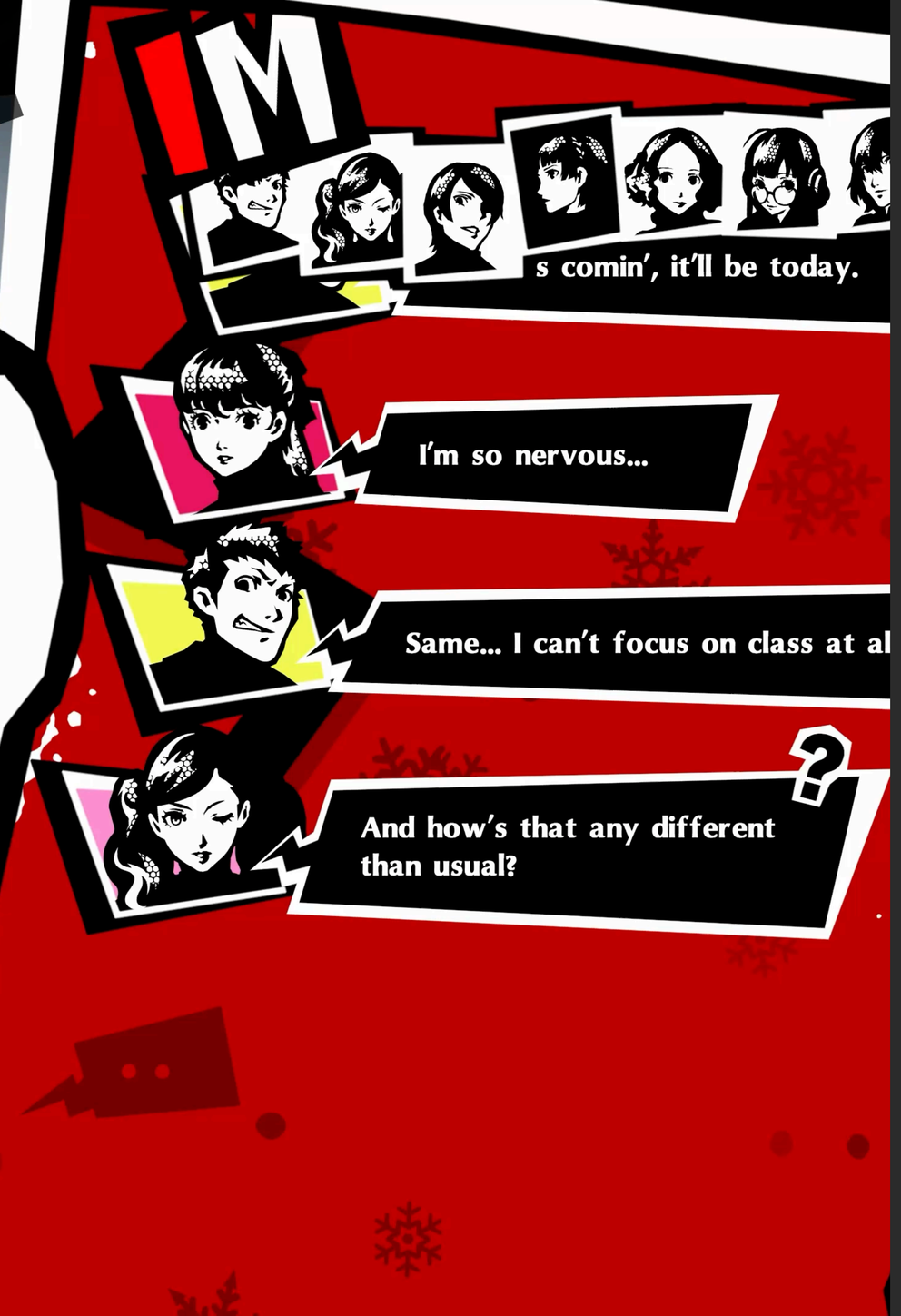


The line 線

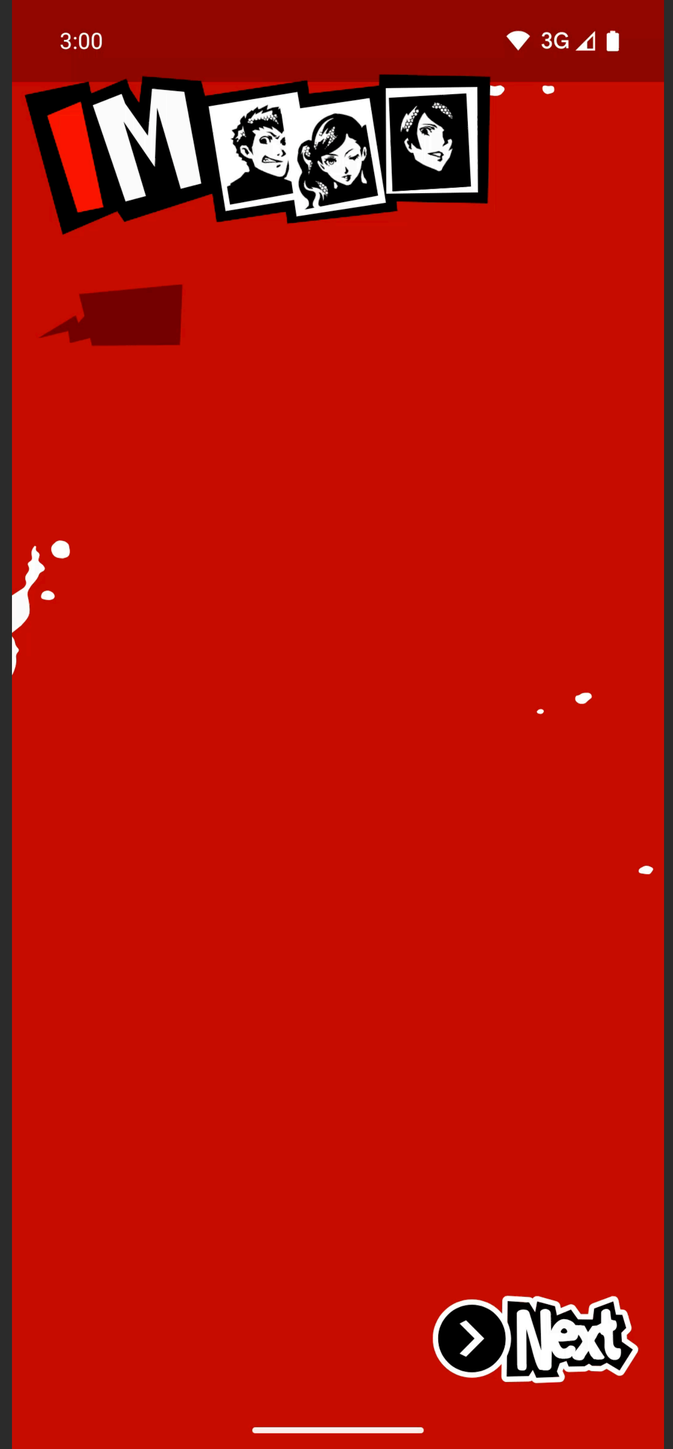
```
val bottomLeft = entry2.lineCoordinates.leftPoint
val bottomRight = entry2.lineCoordinates.rightPoint

onDrawBehind {
  with(linePath) {
    rewind()
    moveTo(topLeft.x, topLeft.y)
    lineTo(topRight.x, topRight.y)
    lineTo(bottomRight.x, bottomRight.y)
    lineTo(bottomLeft.x, bottomLeft.y)
    close()
  }

  drawPath(linePath, Color.Black)
}
```




```
onDrawBehind {
  val currentBottomLeft = lerp(
    start = topLeft,
    stop = bottomLeft,
    fraction = entry1.lineProgress.value,
  )
  val currentBottomRight = lerp(
    start = topRight,
    stop = bottomRight,
    fraction = entry1.lineProgress.value,
  )

  with(linePath) {
    rewind()
    moveTo(topLeft.x, topLeft.y)
    lineTo(topRight.x, topRight.y)
    lineTo(currentBottomRight.x, currentBottomRight.y)
    lineTo(currentBottomLeft.x, currentBottomLeft.y)
    close()
  }

  drawPath(linePath, Color.Black)
}
```




```
onDrawBehind {
  val currentBottomLeft: Offset = lerp(
    start = topLeft,
    stop = bottomLeft,
    fraction = entry1.lineProgress.value,
  )
  val currentBottomRight: Offset = lerp(
    start = topRight,
    stop = bottomRight,
    fraction = entry1.lineProgress.value,
  )

  with(linePath) {
    rewind()
    moveTo(topLeft.x, topLeft.y)
    lineTo(topRight.x, topRight.y)
    lineTo(currentBottomRight.x, currentBottomRight.y)
    lineTo(currentBottomLeft.x, currentBottomLeft.y)
    close()
  }

  drawPath(linePath, Color.Black)
}
```



Takeaways
テイクアウト

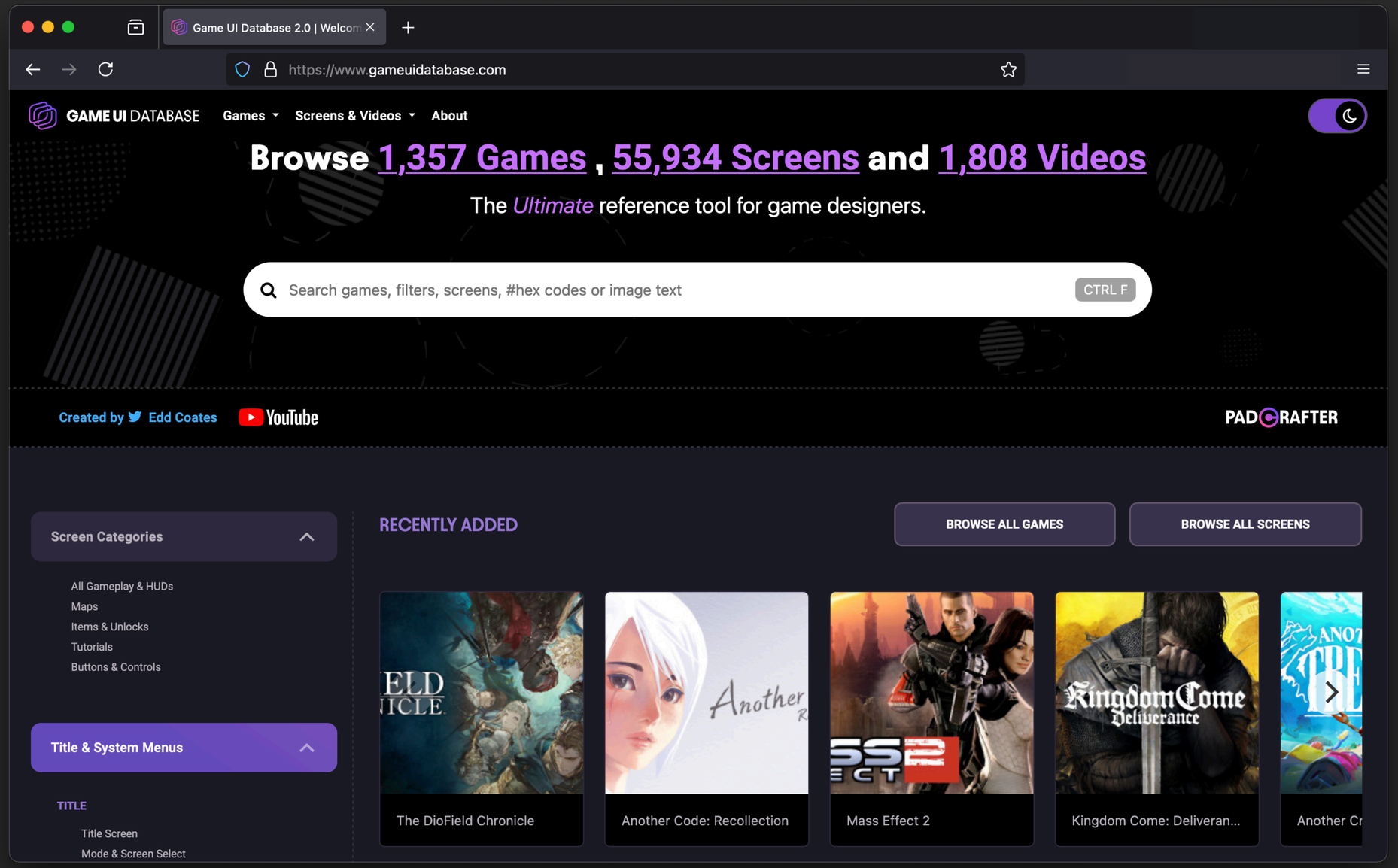
Compose UI facilitates creativity
Compose UIが独創性を高める

Some UIs should be boring, but not all of them
一部のUIは退屈であるべきだが、すべてではない

Maybe we lost some magic with all the consistency
一貫性が無くなったことで、マジックが失われてしまったのかもしれない。

Video games can be source of inspiration
ゲームは発想の源になる

gameuidatabase.com



3:00 3G

We have to find them tomorrow for sure. This is the only lead we have right now.

Yes. It is highly likely that this part-time solicitor is somehow related to the mafia.

If we tail him, he may lead us straight back to his boss.

He talked to Iida and Nishiyama over at Central Street, right?

...

Next

How to draw an owl

1.

2.

1. Draw some circles

2. Draw the rest of the f[redacted]king owl

Creative UIs with Compose

github.com/chris-horner/persona-im

chrishorner@androiddev.social

DroidKaigi 2024

Cash App