**BỘ GIÁO DỤC & ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**KHOA CÔNG NGHỆ THÔNG TIN**

--🙢🕮🙠--

**A picture containing text, device, gauge

Description automatically generated**

**BÁO CÁO CÁ NHÂN**

**CÁC CÔNG NGHỆ LẬP TRÌNH HIỆN ĐẠI**

**ĐỀ TÀI**

**TÌM HIỂU FLUTTER**

**Nhóm 17**

3120410471 – Trịnh Hùng Thái

**Giảng viên phụ trách ThS. PHẠM THI VƯƠNG**

**TP.Hồ Chí Minh, tháng 5 năm 2024**

**CÔNG VIỆC ĐÃ LÀM**

1. Tìm hiểu về widget Image trong Flutter
2. Tìm hiểu về widget Card trong Flutter
3. Tìm hiểu về widget Tabbar trong Flutter
4. Tìm hiểu về widget Drawer trong Flutter
5. Tìm hiểu về widget list trong Flutter
6. Tìm hiểu về widget GridView trong Flutter
7. Tìm hiểu về widget Checkbox trong Flutter
8. Tìm hiểu về widget radio button trong Flutter
9. Tìm hiểu về widget Progress Bar trong Flutter
10. Xây dựng ứng dụng Flutter cơ bản: Trò chơi TicTacToe

# Tìm hiểu về widget Image trong Flutter

Widget Image là một trong những widget cơ bản nhất trong Flutter, được sử dụng để hiển thị hình ảnh. Widget này có thể hiển thị hình ảnh từ nhiều nguồn khác nhau, bao gồm:

1. Tài nguyên assets: Hình ảnh được lưu trữ trong thư mục assets của dự án Flutter.
2. Mạng: Hình ảnh được tải từ internet thông qua URL.
3. Bộ nhớ: Hình ảnh được lưu trữ trong bộ nhớ của thiết bị.

**Ưu điểm:**

* Dễ sử dụng.
* Hỗ trợ nhiều định dạng hình ảnh phổ biến.
* Hiệu suất cao.
* Có thể tùy chỉnh nhiều tính năng như kích thước, vị trí, màu sắc, v.v.

**Nhược điểm:**

* Không hỗ trợ hiển thị ảnh động.
* Khó khăn trong việc xử lý các hình ảnh phức tạp.

Widget Image có cấu trúc khá đơn giản, bao gồm các thuộc tính sau:

* **image**: Thuộc tính này bắt buộc, chỉ định nguồn hình ảnh.
* **height**: Chiều cao của hình ảnh.
* **width**: Chiều rộng của hình ảnh.
* **fit**: Cách thức hình ảnh được điều chỉnh cho phù hợp với kích thước của widget.
* **alignment**: Vị trí của hình ảnh trong widget.
* **color**: Màu sắc được sử dụng để tô màu cho hình ảnh (tùy chọn).
* **filterQuality**: Chất lượng của bộ lọc hình ảnh (tùy chọn).

# Tìm hiểu về widget Card trong Flutter

Card là một widget trong Flutter được sử dụng để hiển thị nội dung một cách đơn giản và trực quan. Nó có các góc bo tròn và đổ bóng, tạo hiệu ứng 3D giúp nội dung nổi bật hơn.

**Ưu điểm:**

* Hiển thị nội dung trực quan và thu hút
* Dễ sử dụng và tùy chỉnh
* Có thể sử dụng cho nhiều mục đích khác nhau

**Nhược điểm:**

* Có thể chiếm nhiều không gian nếu không được thiết kế hợp lý

Chúng ta có thể tùy chỉnh Card bằng cách sử dụng các thuộc tính. Một số thuộc tính cần thiết được đưa ra dưới đây:

|  |  |
| --- | --- |
| **Tên thuộc tính** | **Mô tả** |
| borderOnForeground | Nó được sử dụng để vẽ đường viền phía trước của một child. Theo mặc định, nó là true. Nếu nó là false, nó đã vẽ đường viền phía sau đứa trẻ. |
| color | Nó được sử dụng để tô màu nền của Card. |
| elevation | Nó kiểm soát kích thước bóng bên dưới Card. Giá trị độ cao lớn hơn làm cho khoảng cách bóng lớn hơn. |
| margin | Nó được sử dụng để tùy chỉnh không gian bên ngoài của Card. |
| shape | Nó được sử dụng để chỉ định hình dạng của Card. |
| shadowColor | Nó được sử dụng để vẽ bóng của Card. |
| clipBehavior | Nó được sử dụng để kẹp nội dung của Card. |

# Tìm hiểu về widget Tabbar trong Flutter

TabBar là một widget quan trọng trong Flutter giúp tạo giao diện, cho phép người dùng chuyển đổi giữa các nội dung khác nhau trong ứng dụng. TabBar thường được đặt ở đầu hoặc cuối màn hình, hiển thị các tab với tiêu đề và biểu tượng. Khi người dùng nhấp vào một tab, nội dung tương ứng sẽ được hiển thị.

TabBar bao gồm các thành phần chính sau:

* **TabBar:** Widget hiển thị các tab.
* **Tab:** Widget đại diện cho một tab riêng lẻ.
* **TabBarView:** Widget hiển thị nội dung tương ứng với tab được chọn.
* **TabController:** Widget quản lý trạng thái của TabBar, bao gồm tab nào đang được chọn.

# Tìm hiểu về widget Drawer trong Flutter

Drawer là một lựa chọn thay thế cho các tab vì đôi khi các ứng dụng dành cho thiết bị di động không có đủ không gian để hỗ trợ các tab. Drawer là một màn hình bên vô hình. Đây là một menu trượt bên trái thường chứa các liên kết quan trọng trong ứng dụng và chiếm một nửa màn hình khi hiển thị.

Drawer bao gồm các thành phần chính sau:

* **Drawer**: Widget hiển thị nội dung menu.
* **DrawerHeader:** Widget hiển thị phần đầu của menu, thường bao gồm logo, tên ứng dụng hoặc thông tin người dùng.
* **DrawerItem:** Widget đại diện cho một mục trong menu.
* **ListView:** Widget hiển thị danh sách các mục menu.

# Tìm hiểu về widget list trong Flutter

Widget ListView là một widget cơ bản trong Flutter, được sử dụng để hiển thị danh sách các widget con. Nó có thể hiển thị danh sách các widget theo chiều dọc hoặc chiều ngang, với khả năng cuộn nếu danh sách dài hơn màn hình. Cấu trúc của List như sau:

|  |
| --- |
| ListView(  children: <Widget>[  // Danh sách các widget con  ],  ) |

Tham số quan trọng:

* **children:** Danh sách các widget con được hiển thị trong danh sách.

**Ưu điểm:**

* Dễ sử dụng và triển khai.
* Hiệu quả và có thể xử lý danh sách lớn.
* Hỗ trợ nhiều loại widget con khác nhau.
* Có thể tùy chỉnh giao diện của danh sách.

**Nhược điểm:**

* Có thể gặp vấn đề về hiệu suất nếu danh sách quá lớn.
* Khó khăn trong việc tùy chỉnh giao diện phức tạp.

**Một số loại ListView khác:**

1. ListView: Hiển thị danh sách các widget theo chiều dọc.
2. GridView: Hiển thị danh sách các widget theo dạng lưới.
3. CustomScrollView: Cho phép tùy chỉnh chi tiết cách thức cuộn của danh sách.

# Tìm hiểu về widget GridView trong Flutter

GridView là một widget trong Flutter giúp hiển thị danh sách các item dưới dạng lưới 2D. Nó tương tự như ListView nhưng hiển thị các item theo hàng và cột thay vì chỉ theo một chiều. Nó có thể được sử dụng để tạo ra các bố cục như danh sách ảnh, danh sách sản phẩm, v.v.

**Ưu điểm:**

* Hiển thị danh sách các item một cách trực quan và dễ nhìn.
* Có thể tùy chỉnh bố cục của lưới, chẳng hạn như số lượng cột, khoảng cách giữa các item, v.v.
* Hỗ trợ cuộn trang để hiển thị nhiều item hơn.

**Nhược điểm:**

* Có thể phức tạp hơn so với ListView khi tùy chỉnh bố cục.
* Hiệu suất có thể bị ảnh hưởng nếu có quá nhiều item trong danh sách.

GridView được cấu tạo từ các thành phần sau:

* **children**: Danh sách các widget sẽ được hiển thị trong GridView.
* **gridDelegate**: Một đối tượng xác định cách thức sắp xếp các item trong GridView.
* **scrollDirection**: Hướng cuộn của GridView (theo chiều ngang hoặc chiều dọc).
* **shrinkWrap**: Thu hẹp kích thước của GridView để vừa với nội dung.

# Tìm hiểu về widget Checkbox trong Flutter

Checkbox là một widget Material Design cho phép người dùng chọn một hoặc nhiều tùy chọn từ một danh sách. Nó hiển thị một hộp vuông nhỏ với dấu kiểm bên trong khi được chọn.

Cấu trúc của một Widget CheckBox được mô tả như sau:

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| value | Nó được sử dụng cho dù Checkbox có được chọn hay không. |
| onChanged | Nó sẽ được gọi khi giá trị được thay đổi. |
| Tristate | Nó là false, theo mặc định. Giá trị của nó cũng có thể là true, false hoặc null. |
| activeColor | Nó chỉ định màu của Checkbox đã chọn. |
| checkColor | Nó chỉ định màu của biểu tượng kiểm tra khi chúng được chọn. |
| materialTapTargetSize | Nó được sử dụng để định cấu hình kích thước của mục tiêu chạm. |

# Tìm hiểu về widget radio button trong Flutter

RadioButton là một widget material design cho phép người dùng chọn một tùy chọn duy nhất từ một danh sách các lựa chọn. Widget này thường được sử dụng trong các biểu mẫu để thu thập thông tin từ người dùng.

RadioButton là một widget stateless, nghĩa là nó không có trạng thái riêng. Widget này nhận một số thuộc tính để tùy chỉnh giao diện và chức năng:

* **groupValue**: Nó được sử dụng để chỉ định mục hiện được chọn cho nhóm nút radio.
* **title**: Nó được sử dụng để chỉ định nhãn nút radio.
* **value**: Nó chỉ định giá trị trái, được biểu thị bằng một nút radio.
* **onChanged**: Nó sẽ được gọi bất cứ khi nào người dùng chọn nút radio.

# Tìm hiểu về widget Progress Bar trong Flutter

Progress bar (Thanh tiến trình) là một phần tử điều khiển đồ họa được sử dụng để hiển thị tiến trình của một tác vụ như tải xuống, tải lên, cài đặt, truyền tệp, v.v. Trong phần này, chúng ta sẽ hiểu cách hiển thị progress bar trong một ứng dụng rung.

Flutter có thể hiển thị progress bar với sự trợ giúp của hai widget con, được đưa ra dưới đây:

* **LinearProgressIndicator**
* **RoundProgressIndicator**

Cấu trúc của widget ProgressBar trong Flutter khá đơn giản. Ta có thể sử dụng widget **LinearProgressIndicator** hoặc **CircularProgressIndicator** để tạo ProgressBar cho giao diện của mình:

|  |
| --- |
| LinearProgressIndicator(  value: 0.5, // Giá trị tiến trình (từ 0.0 đến 1.0)  backgroundColor: Colors.grey, // Màu nền  valueColor: Colors.blue, // Màu phần được tô màu  )  CircularProgressIndicator(  value: 0.5, // Giá trị tiến trình (từ 0.0 đến 1.0)  backgroundColor: Colors.grey, // Màu nền  valueColor: Colors.blue, // Màu phần được tô màu  ) |

# Xây dựng ứng dụng Flutter cơ bản: Trò chơi TicTacToe

TicTacToe (còn gọi là XOX) là một trò chơi kinh điển, đơn giản nhưng đầy thú vị. Với Flutter, bạn có thể dễ dàng tạo ra một phiên bản trò chơi này với giao diện đẹp mắt và logic chơi hoàn chỉnh. Trò chơi dành cho hai người chơi, mỗi người chơi lần lượt đánh dấu X hoặc O vào các ô trên bảng 3x3. Người chơi đầu tiên tạo ra một hàng, cột hoặc đường chéo gồm 3 dấu hiệu của mình sẽ chiến thắng. Nếu tất cả các ô được đánh dấu mà không có người chiến thắng, trò chơi kết thúc hòa.

A black cell phone with a game on screen

Description automatically generated

Màn hình chính trò chơi

Màn hình thông báo

Trình bày từ trang 175 đến trang 187 trong cuốn báo cáo