

Creation de code de vue : Main Activity

1. Présentation générale

La classe **MainActivity** :

- Est une activité Android principale qui gère l'interface utilisateur.
 - Utilise la classe **Contrôle** comme intermédiaire pour gérer les données.
 - Permet à l'utilisateur de saisir ses données (poids, taille, âge, sexe) et de calculer son IMG (Indice de Masse Grasse).
-

2. Initialisation des composants graphiques

Méthode : **init()**

- **Description** : Lie les éléments de l'interface utilisateur (UI) à leurs références dans le code, et configure les comportements initiaux.
- **Étapes clés** :
 1. **Association des widgets** :
 - Les EditText pour le poids, la taille et l'âge.
 - Le RadioButton pour le sexe.
 - Les ImageView et TextView pour afficher les résultats.
 2. **Création du contrôleur** :
 - Instance de **Contrôle** initialisée avec le contexte de l'application.
 3. **Ajout de comportement** :
 - Appel de `ecouteCalcul()` pour définir un gestionnaire d'événements sur le bouton de calcul.

```

/**
 * initialisation des liens avec les objets graphiques
 */
private void init(){ 1 usage
    PoidsText = (EditText) findViewById(R.id.PoidsText) ;
    TailleText = (EditText) findViewById(R.id.TailleText);
    AgeText = (EditText) findViewById(R.id.AgeText) ;
    radioHomme = (RadioButton) findViewById(R.id.radioHomme) ;
    bodyShape = (ImageView) findViewById(R.id.bodyShape) ;
    observation = (TextView) findViewById(R.id.observation) ;
    this.control= Controle.getInstance( contexte: this);
    ecouteCalcul();
}

```

3. Gestion de l'événement clic sur le bouton

Méthode : **ecouteCalcul()**

- **Description** : Définit le comportement du bouton lorsque l'utilisateur clique dessus.
- **Étapes clés** :
 1. **Récupération des données saisies** :
 - Les valeurs de poids, taille, âge sont récupérées à partir des champs de saisie.
 - Une vérification (try-catch) empêche les erreurs liées à des saisies incorrectes.
 2. **Validation des données** :
 - Vérifie que les champs ne contiennent pas de valeurs nulles ou invalides.
 - Affiche un message d'erreur avec Toast en cas de saisie incorrecte.
 3. **Calcul des résultats** :
 - Si les données sont valides, la méthode afficherResultat() est appelée avec les valeurs saisies.

```

/**
 * Ecoute evenement sur bouton calcul
 */
private void ecouteCalcul(){ 1 usage
    ((Button) findViewById(R.id.calculIMC)).setOnClickListener(new View.OnClickListener(){
        public void onClick(View v){
            // Affiche un message temporaire "Calcul en cours..." lors du clic
            //Toast.makeText(MainActivity.this, "Calcul en cours...", Toast.LENGTH_SHORT).show();
            //Log.d("BoutonTest", "Le bouton calculIMC a été cliqué.*****");
            Integer poids=0;
            Integer taille=0;
            Integer age=0;
            Integer sexe=0;
            // recuperer les donnees saisie
            try {
                poids=Integer.parseInt(PoidsText.getText().toString());
                taille=Integer.parseInt(TailleText.getText().toString());
                age=Integer.parseInt(AgeText.getText().toString());
            } catch (Exception e){}
            if (radioHomme.isChecked()){sexe=1;}
            // controle des donnees saisie
            if (poids*taille*age ==0)
                {Toast.makeText( context: MainActivity.this, text: "saisie incorrect", Toast.LENGTH_SHORT).show();}
            else {
                afficherResultat(poids,taille,age,sexe);
            }
        }
    });
}

```

4. Affichage des résultats

Méthode : **afficherResultat()**

- **Description** : Gère l'affichage des résultats (IMG, message, et image).
- **Étapes clés** :
 1. **Création du profil** :
 - Appelle la méthode `creerProfile()` du contrôleur avec les données saisies.
 2. **Récupération des résultats** :
 - L'IMG et le message sont récupérés depuis le contrôleur.
 3. **Affichage des données** :
 - En fonction du message (normal, trop maigre, ou trop de graisse), sélectionne une image correspondante et ajuste la couleur du texte.
 - Le message final et l'IMG sont affichés dans le champ observation

```

private void afficherResultat(Integer poids,Integer taille, Integer age, Integer sexe)
    this.control.creerProfile(poids, taille, age, sexe, contexte: this);
    float img = this.control.getImg();
    String message = this.control.getMessage();
    if (message.equalsIgnoreCase( anotherString: "normal")) {
        if (sexe == 1) {
            bodyShape.setImageResource(R.drawable.muscled);
            observation.setTextColor(Color.GREEN);
        } else {
            bodyShape.setImageResource(R.drawable.athletof);
            observation.setTextColor(Color.GREEN);
        }
    } else if (message.equalsIgnoreCase( anotherString: "trop maigre")) {
        if (sexe == 1) {
            bodyShape.setImageResource(R.drawable.skinny);
            observation.setTextColor(Color.BLUE);
        } else {
            bodyShape.setImageResource(R.drawable.skinnyg);
            observation.setTextColor(Color.BLUE);
        }
    } else if (message.equalsIgnoreCase( anotherString: "trop de graisse")) {
        if (sexe == 1) {
            bodyShape.setImageResource(R.drawable.fat);
            observation.setTextColor(Color.RED);
        } else {
            bodyShape.setImageResource(R.drawable.fatgirl);
            observation.setTextColor(Color.RED);
        }
    }

    observation.setText(String.format("%180.01f", img) + " : " + message);
}

```

Testant maintenant l'application avec des valeurs variées :



