

EditCalc Version 3

TRUONG Jacky, MOITY Anthony, BOUSBAINE Rabah

Livrables

Lognes, mars 2020

Centre GRETA MTE 77 Lognes
- Espace DIDEROT
95-97 boulevard du Segrais 77185 Lognes

Table des matières

Introduction.....	Page 3
Compréhension du besoin.....	Page 3
Diagramme de cas d'utilisation.....	Page 3
Diagramme de classes (généré).....	Page 4
Diagramme de séquence de conception.....	Page 5
Choix du thème.....	Page 6
Maquettes.....	Page 7
Organisation.....	Page 11
Ressources.....	Page 11
Planning.....	Page 12
Tableau de bord.....	Page 13
Présentation technique.....	Page 14
Création et disposition des éléments.....	Page 14
Programmation des éléments.....	Page 14
Test fonctionnel.....	Page 15
Descriptif du code.....	Page 16
Mémoires.....	Page 27
Mémoire sur les bases de données, serveurs et outils utilisés.....	Page 27
Mémoire sur l'intégration continue.....	Page 34
Mémoire sur les outils de déploiement.....	Page 37
Manuels.....	Page 47
Manuel d'installation.....	Page 47
Manuel utilisateur.....	Page 55
Bilan.....	Page 64
Tableau des évolutions.....	Page 65
Conclusion.....	Page 65

Introduction

L'Editeur dénommé « EditCALC » propose des solutions éducatives pour les écoles, et souhaite proposer une application pour aider les élèves de « CP1 » à compter jusqu'à 10 à travers une interface Homme-Machine. Notre mission sera de créer cette application.

Compréhension du besoin

Après la proposition d'une deuxième version, le client n'est pas satisfait de l'utilisation proposée de cette calculette bi fonction.

Le but du développement de cette troisième version est de pouvoir afficher le nom de l'utilisateur ayant fait le plus haut score. Son score sera également affiché. Le but de cette évolution est d'identifier le meilleur utilisateur dans un ensemble (Ex : Une salle de classe) et éventuellement cet utilisateur. Cette évolution permet également de motiver les autres utilisateurs, déterminés à voir à leur tour, leur nom sur la calculatrice.

Diagramme de cas d'utilisation

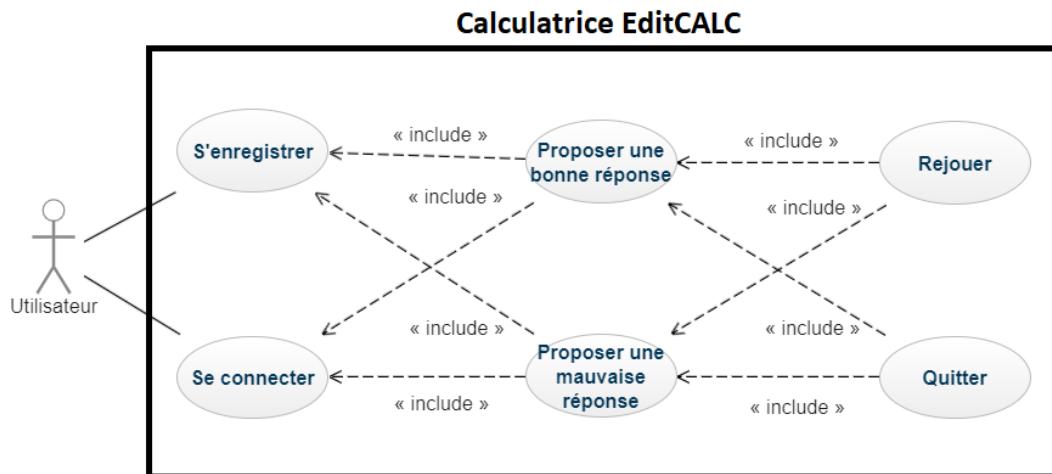


Diagramme de classes (généré par easyUML)

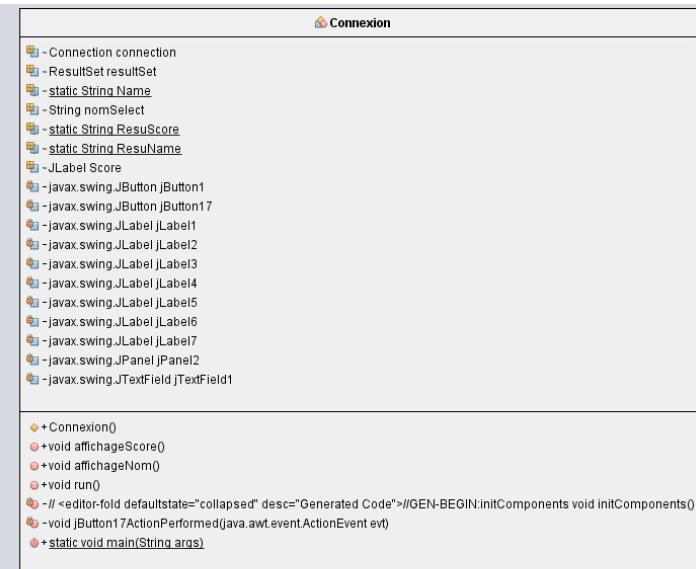
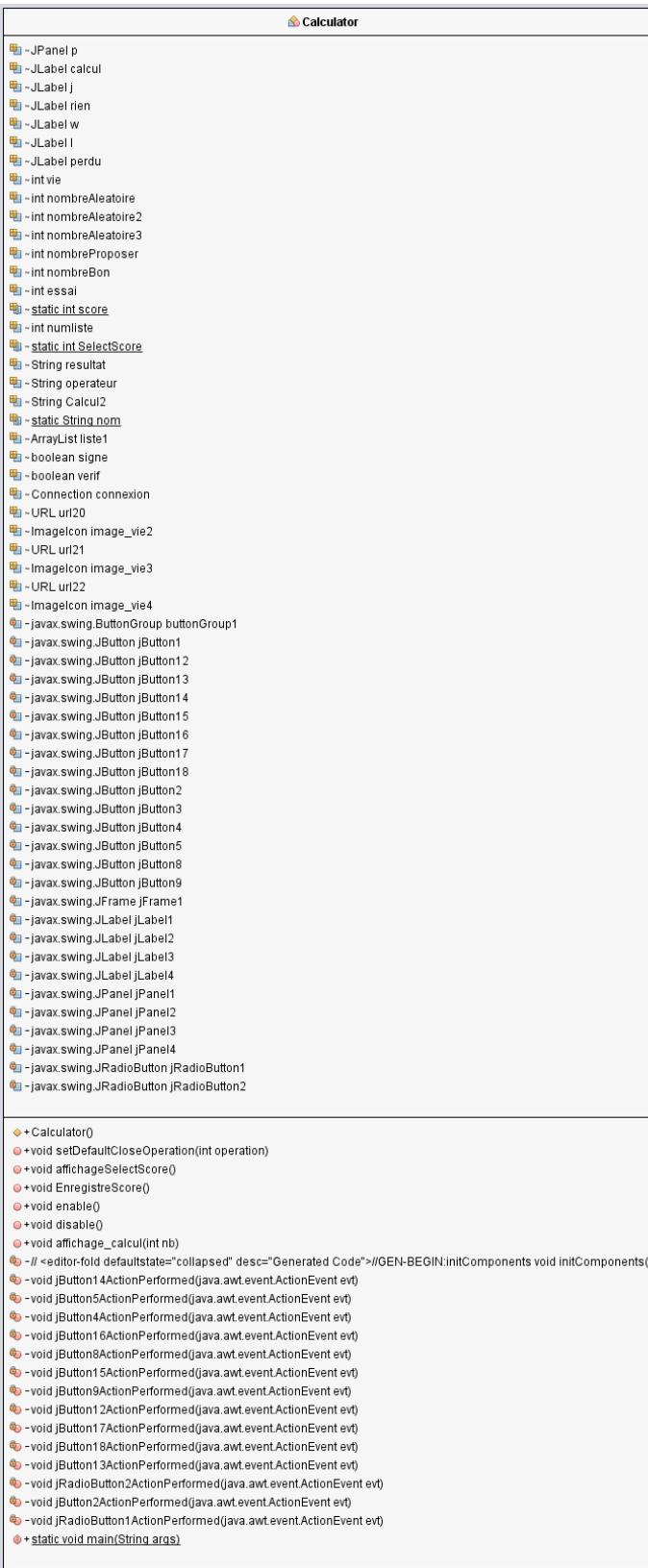
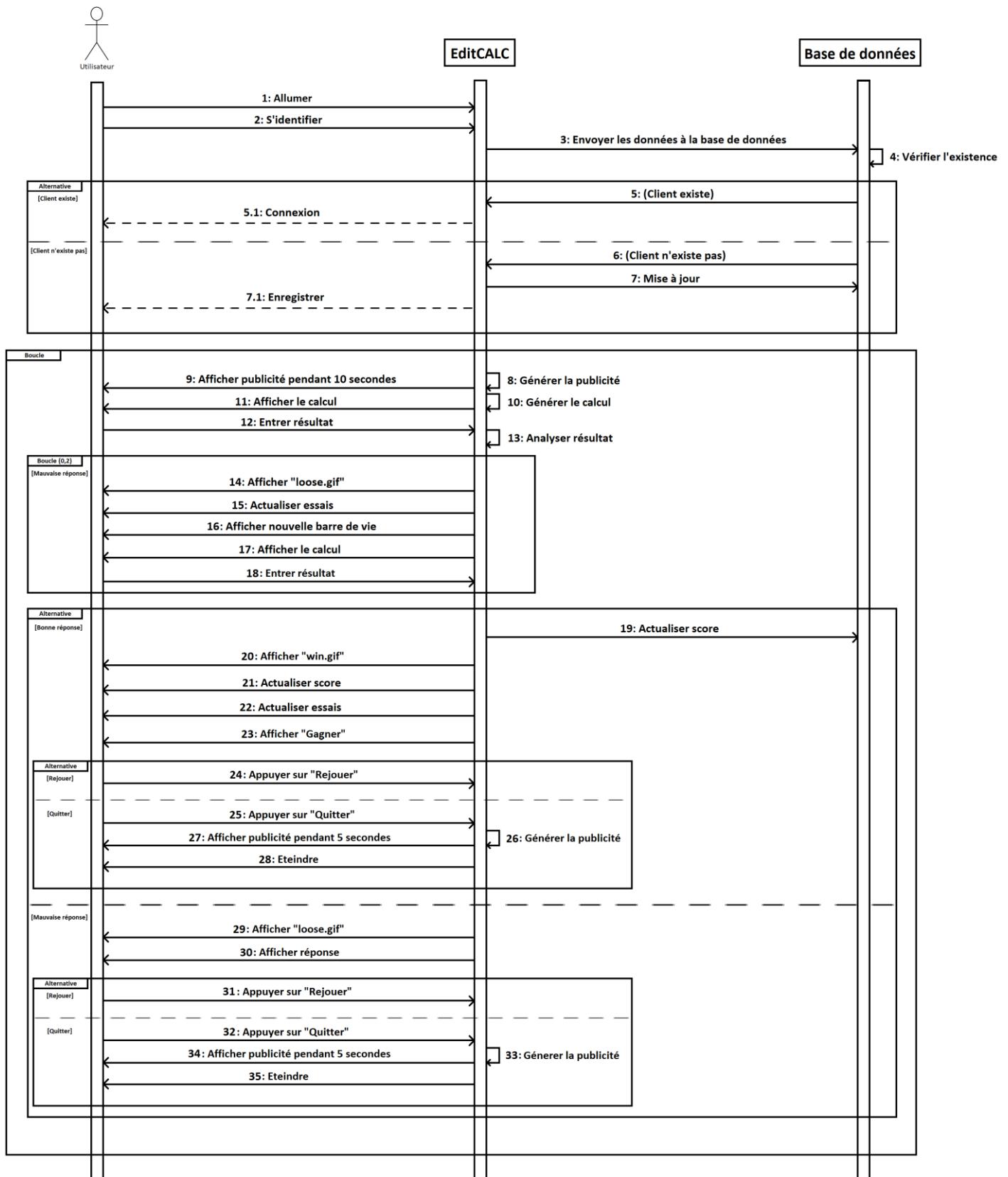


Diagramme de séquence de conception



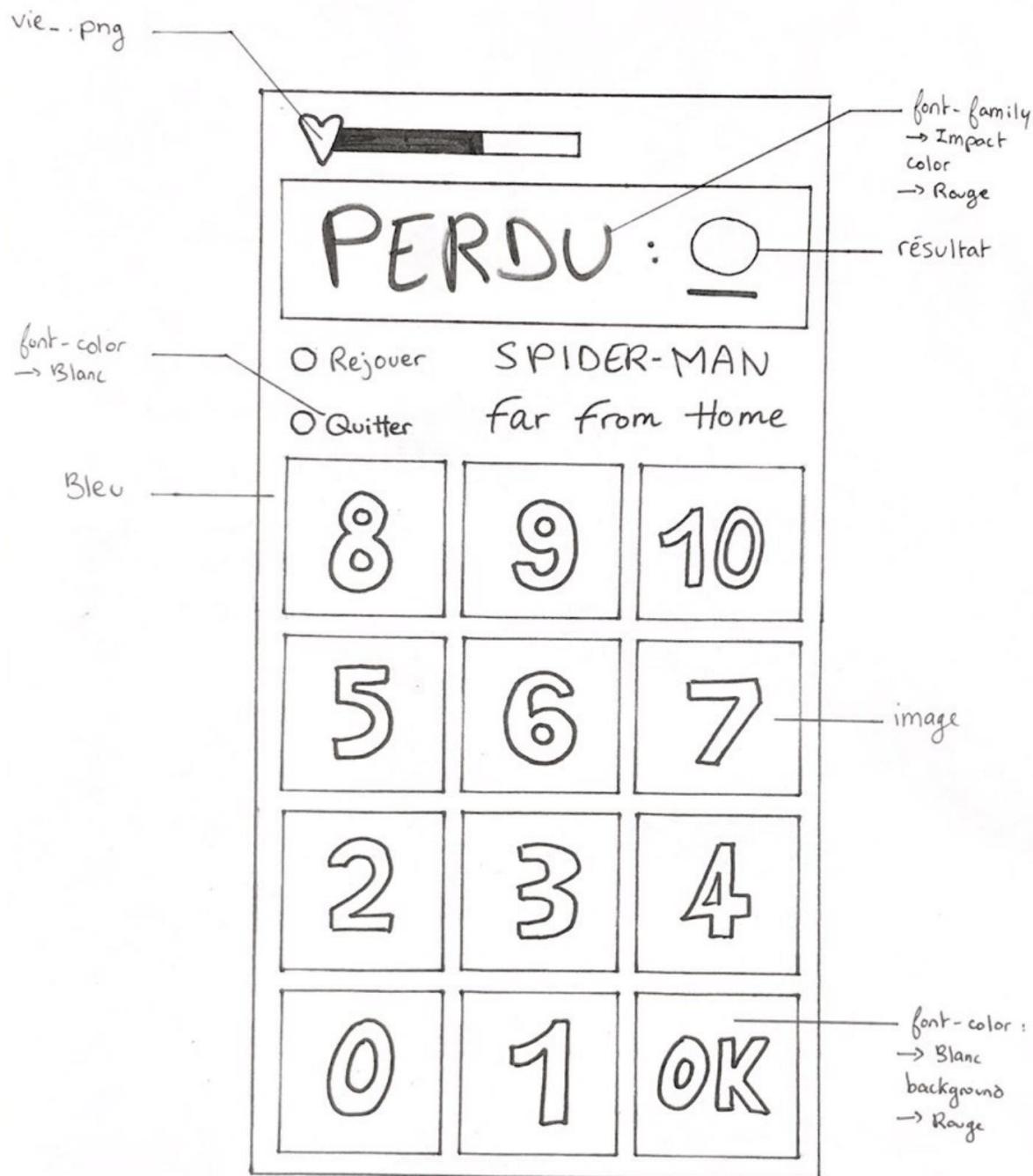
Choix du thème

Nous avons opté pour un thème Spiderman, car selon nous, Spiderman est le héro préféré de tous les enfants. Le choix de ce thème permet de capter l'attention des enfants, ce qui pourrait les encourager à travailler plus à travers cette application.

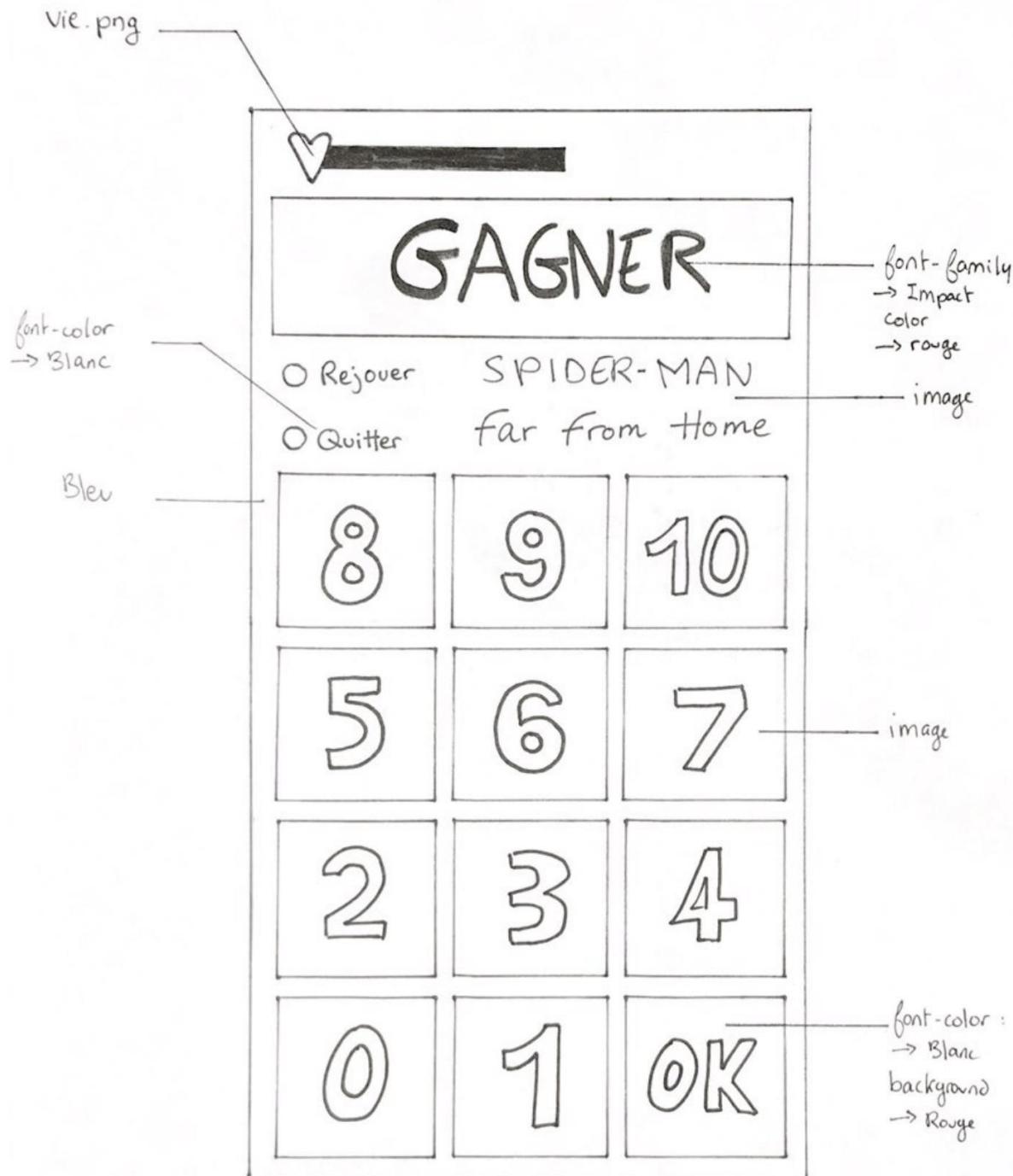


Maquettes

Avant de développer la version 3, nous devions dans un premier temps concevoir des maquettes afin de valider et tester certains aspects de la calculatrice.

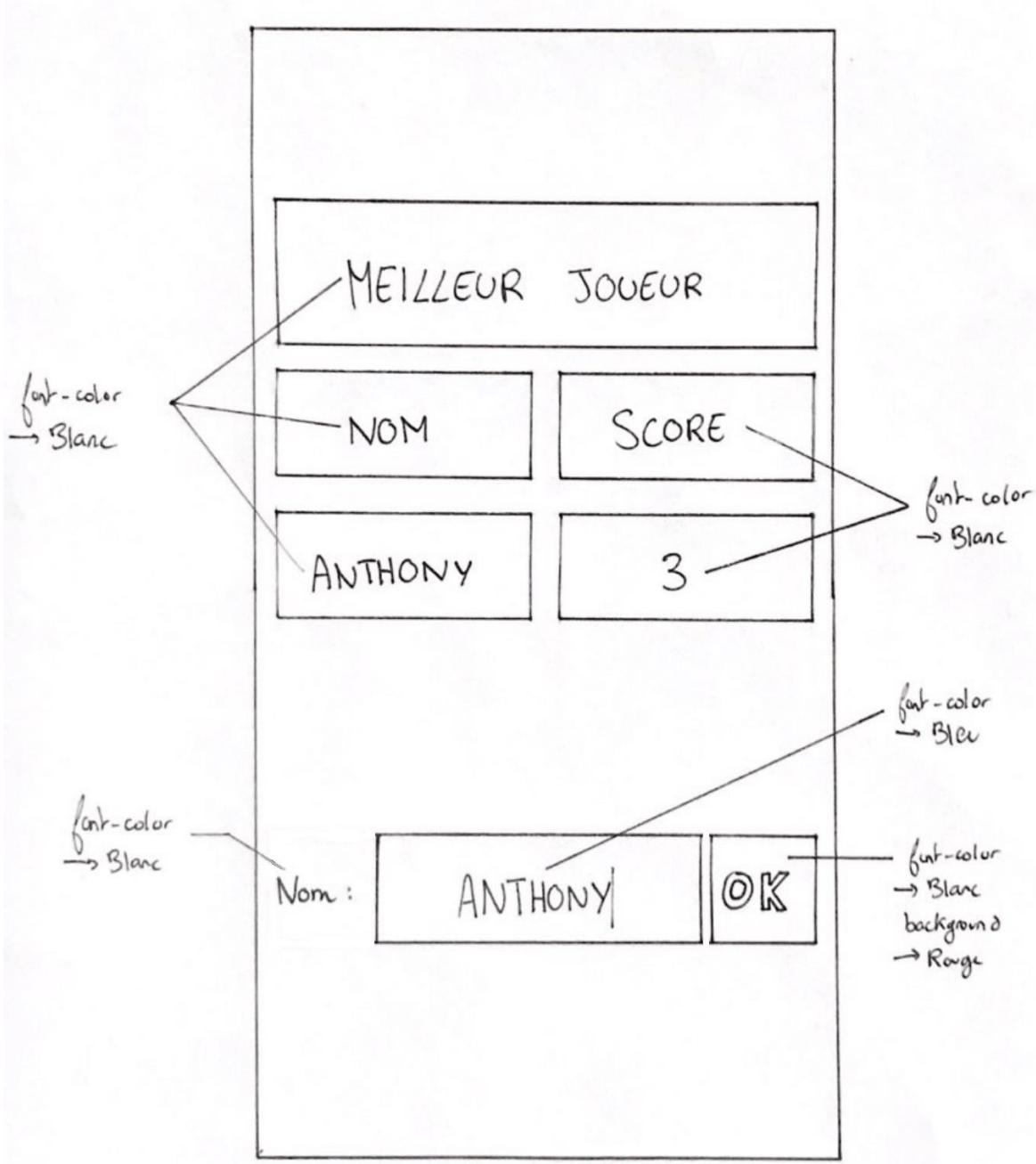


■ Maquette V2.1 (Perdu + réponse)

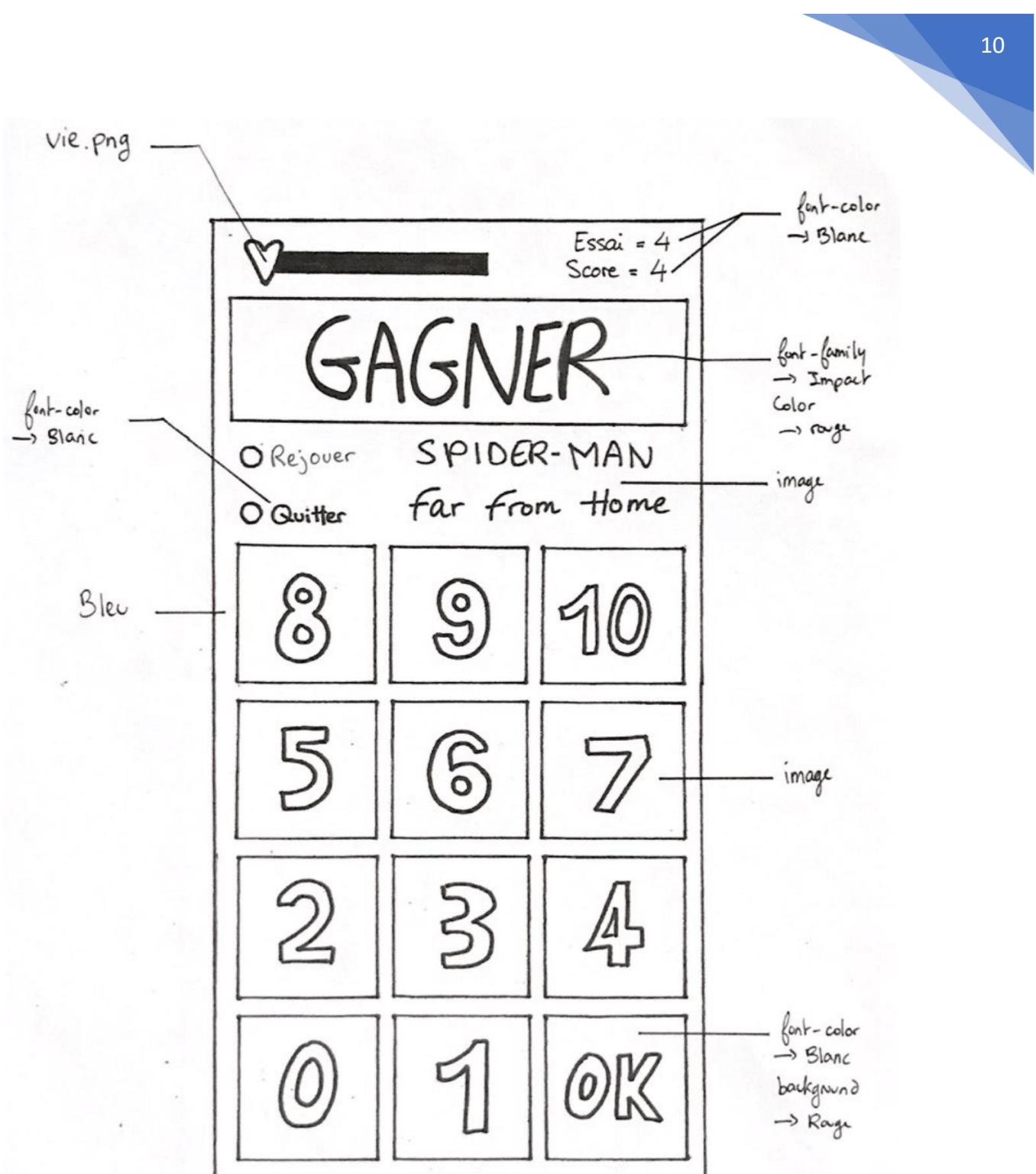


Maquette V2.2 (Gagner)

Maquettes finales



■ Maquette V3.1 (Interface d'enregistrement + connexion + meilleur joueur)



Maquette V3.2 (Interface joueur +
essai/score)

Organisation

Avant de se lancer dans le développement de la version 3, nous devons savoir quelles ressources utilisées (ressources logicielles et humaines), et définir un planning.

Ressources logicielles

- ❖ Netbeans IDE 8.2 + Swing : Nous avons tapé le code sur Netbeans puis pour l'interface graphique, nous avons utilisé les composants Swing
- ❖ Discord : Conçu initialement pour les « gamers », nous avons utilisé cette plateforme de communication pour s'envoyer des blocs de code, parler via un chat vocal, et travailler en partage d'écran
- ❖ Trello : C'est un logiciel de gestion de projet en ligne, fonctionnant avec des cartes et des colonnes. Nous l'avons utilisé pour nous assigner des tâches et voir l'avancé de chacun sans pour autant passer par Discord.

Ressources humaines

- ❖ Jacky
- ❖ Anthony
- ❖ Rabah
- ❖ Les professeurs du GRETA
- ❖ Les contacts tiers

Planning

Du 20 mars à aujourd'hui							
Jours	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche
Disponibilité	Available	Available	Available	Available	Available	Unavailable	Unavailable

 Disponible

 Indisponible

Tableau de bord

Période concernée	Du : 20-mars		du		27-mars		Réalisé	Remarques
	Livrables et jalons	% d'avancement	Avancement	Livrable	Date de fin prévue	Replanifié		
Création d'une base de données (name-score)	5%	—	—	—	21-mars	X	/	
Ajout d'une [frame] pour l'enregistrement ou la connexion	15%	—	—	—	21-mars	X	/	
Requêtes SQL (affichage du score, insertion du joueur dans la bdd, mise à jour du meilleur score)	35%	3	3	3	22-mars	23-mars	X	Des difficultés comme prévu
Récupération du score dans la bdd en cas de connection	40%	2	2	2	23-mars	X	/	
Affichage du nom et du score du meilleur joueur dans la nouvelle [frame] et dans l'interface principale	50%	2	3	3	24-mars	25-mars	X	Problèmes de mise en relation
Modélisation et documentation de la bdd	60%	2	2	2	22-mars	X	/	
Documentation des outils et serveurs utilisés	70%	3	2	2	24-mars	29-mars	X	
Documentation technique	80%	2	1	1	26-mars	X	/	
Présentation issue de la veille	85%	1	1	1	/	X	/	Identique à celle livrée pour la version 1
Présentation powerpoint du projet	95%	3	—	—	26-mars	28-mars	X	/
Démonstration du fonctionnement lors de la livraison	100%	3	—	—	27-mars	31-mars	X	/
Commentaires :								
Réalisé								
Avenir								
Point dur et solution								
Risque								
Cotation Risque								
Criticité								
Moyenne								
Faible								

Avancement	Valeur à saisir	Valeur à saisir	Livrable
En avance	1	1	Pas de difficultés
Conforme au prévu	2	2	Quelques difficultés
En retard	3	3	Grandes difficultés

Requêtes SQL + Affichage des valeurs => Mettre rapidement en place l'interface graphique et se concentrer uniquement sur la base de données

Moyenne

Criticité
Moyenne

Risque
Faible

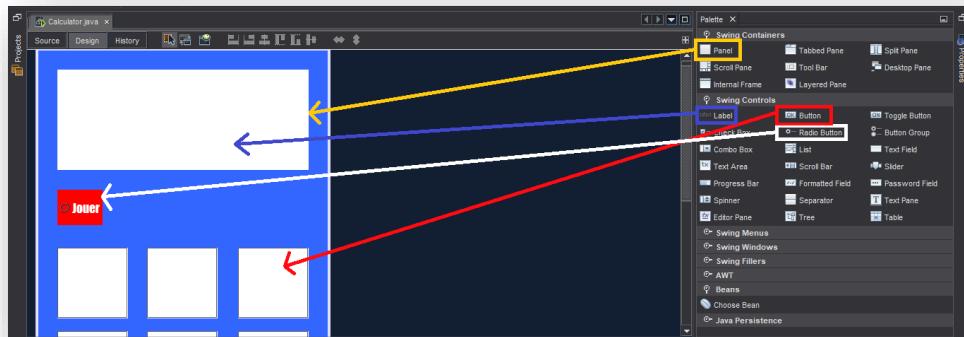
Présentation technique

Dans la conception de cette version 2 de la calculatrice, nous avons utilisé les composants Swing, afin de créer une interface Homme-machine entre les utilisateurs et la calculatrice. Nous pourrons ainsi gérer le code source de la calculatrice ainsi que son design.

La création et la disposition des éléments

Les composants Swing permettent de créer les éléments graphiques.

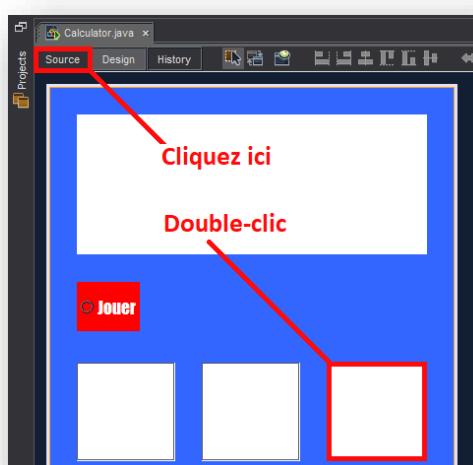
Les éléments sont disposés manuellement, et leurs apparences peuvent être modifiées depuis la palette Swing.



La programmation des éléments

Disposer les éléments ne suffit pas, il faut les lier, leur attribuer des méthodes...etc

Pour cela, nous devons aller dans « Source » ou bien faire un double-clic sur l'élément qu'on veut programmer



Descriptif des tests

TEST FONCTIONNEL V3				
N°	ACTIONS	ATTENDU	PROBLÈME RENCONTRÉ	RESULTATS
1	Exécuter la calculatrice	Ouverture de la calculatrice	L'application ne se lance pas à cause de problèmes lié à la base de donnée.	OK
2	Affichage de l'interface du meilleur joueur (nom, score) et insérer un nouveau joueur où se connecte directement	Affichage du joueur ayant fait le meilleur score. Insérer un nouvel utilisateur.	Problème requête SQL Affichage des deux valeurs sur l'interface	OK
3	Si c'est un joueur qui existe déjà dans la base de donnée, il continue avec le dernier score enregistré sinon, il commence à 0	Nouveau joueur : Commence à 0 Joueur déjà enregistré : Continue avec son score	Affichage de nom et score sur l'interface graphique	OK
4	Appuyer sur le bouton « JOUER »	La calculatrice génère une publicité pendant 10 secondes et génère un calcul aléatoire	Problème trouvé en V1 mais pas en V2 ni en V3	OK
5	Appuyer sur un chiffre	Génère le nombre d'image demandé et appuie sur « OK » pour valider	Problème trouvé en V1 mais pas en V2 ni en V3	OK
6	Système de vie	L'utilisateur perd de la vie à chaque mauvaise réponse	Problème trouvé en V2 mais pas en V3	OK
7	GIF : GAGNÉ et PERDU	Affichage de GIF lorsque l'utilisateur gagne ou perd	Problème trouvé en V2 mais pas en V3	OK
8	Résultat final (gagné)	L'utilisateur doit entrer la bonne réponse avant 3 choix. Cela affichera le GIF gagnant puis affichera « GAGNÉ »	Problème trouvé en V2 mais pas en V3	OK
9	Résultat final (perdu)	L'utilisateur doit entrer la bonne réponse avant 3 choix. Si ce n'est pas le cas, cela affichera le GIF perdant puis affichera « PERDU » en donnant la bonne réponse	Problème trouvé en V2 mais pas en V3	OK
10	Bouton rejouer et quitter	Après avoir connu son résultat (GAGNÉ ou PERDU), le système propose à l'utilisateur de rejouer ou de quitter	Problème trouvé en V2 mais pas en V3	OK
11	La calculatrice se ferme	5 secondes après avoir quitté. La calculatrice régénère une pub puis se ferme	La pub ne s'affiche pas	OK

Descriptif du code

Calculatrice :

```
package calculatrice;

import java.awt.Font;
import java.awt.GridLayout;
import java.net.URL;

import java.util.concurrent.Executors;
import java.util.concurrent.ScheduledExecutorService;
import java.util.concurrent.TimeUnit;

import javax.swing.ImageIcon;
import javax.swing.JLabel;
import java.util.ArrayList;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

import java.sql.Connection;
import javax.swing.JPanel;

/*
 * @author rabah
 */
public class Calculator extends javax.swing.JFrame {

    /**
     * Creates new form Calculator
     */

    @Override
    public void setDefaultCloseOperation(int operation) {
        super.setDefaultCloseOperation(operation);
    }

    JPanel p = new JPanel();
    JLabel calcul = new JLabel();
    JLabel j = new JLabel();
    JLabel rien = new JLabel();
    JLabel w = new JLabel();
    JLabel l = new JLabel();
    JLabel perdu = new JLabel();
```

```

int vie=4;
int nombreAleatoire; //on initialise une variable nombreAleatoire qui est un nombre aléatoire entre 0 et 9
int nombreAleatoire2; // on créer une nouvelle variable nombreAleatoire2
int nombreAleatoire3 = (int) (Math.random() * (2));
int nombrePropose;
int nombreBon;
int essaie=0;
static int score; //On initialise en static les variables qui nous serviront pour les requête sql car si on ne le fait pas SQL ne les accepte pas
int numliste;
static int SelectScore;

String resultat;
String operateur;
String Calcul;
static String nom;

ArrayList listel = new ArrayList();
boolean signe;
boolean verif;
Connection connexion =null;

URL url120 = ClassLoader.getSystemClassLoader().getResource("images/vie2.png");
ImageIcon image_vie2 = new ImageIcon(new ImageIcon(url120).getImage().getScaledInstance(225, 45, 225));

URL url121 = ClassLoader.getSystemClassLoader().getResource("images/vie4.png");
ImageIcon image_vie3 = new ImageIcon(new ImageIcon(url121).getImage().getScaledInstance(225, 45, 225));

URL url122 = ClassLoader.getSystemClassLoader().getResource("images/vie3.png");
ImageIcon image_vie4 = new ImageIcon(new ImageIcon(url122).getImage().getScaledInstance(225, 45, 225));

public Calculator() throws SQLException {
    initComponents();
    this.setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE); //pour ne plus pouvoir fermer la fenêtre
    affichageSelectScore(); //On exécute la requête qui sélectionne le score du joueur par rapport à son Nom

    URL url1 = ClassLoader.getSystemClassLoader().getResource("images/0.jpg"); //on enregistre l'image qui est sur le bouton 0
    ImageIcon icon = new ImageIcon(new ImageIcon(url1).getImage().getScaledInstance(95, 95, 95)); //on créer un icone et on met l'image dans l'icone et on défini la taille
    jButton12.setIcon(icon); //on ajoute l'icône au bouton

    URL url2 = ClassLoader.getSystemClassLoader().getResource("images/1.jpg");
    ImageIcon icon1 = new ImageIcon (new ImageIcon(url2).getImage().getScaledInstance(95, 95, 95));
    jButton5.setIcon(icon1);

    URL url3 = ClassLoader.getSystemClassLoader().getResource("images/2.jpg");
    ImageIcon icon2 = new ImageIcon(new ImageIcon(url3).getImage().getScaledInstance(95, 95, 95));
    jButton6.setIcon(icon2);

    URL url4 = ClassLoader.getSystemClassLoader().getResource("images/3.jpg");
    ImageIcon icon3 = new ImageIcon(new ImageIcon(url4).getImage().getScaledInstance(95, 95, 95));
    jButton16.setIcon(icon3);

    URL url5 = ClassLoader.getSystemClassLoader().getResource("images/4.jpg");
    ImageIcon icon4 = new ImageIcon(new ImageIcon(url5).getImage().getScaledInstance(95, 95, 95));

    jButton5.setIcon(icon4);

    URL url6 = ClassLoader.getSystemClassLoader().getResource("images/5.jpg");
    ImageIcon icon5 = new ImageIcon(new ImageIcon(url6).getImage().getScaledInstance(95, 95, 95));

    jButton4.setIcon(icon5);

    URL url7 = ClassLoader.getSystemClassLoader().getResource("images/6.jpg");
    ImageIcon icon6 = new ImageIcon(new ImageIcon(url7).getImage().getScaledInstance(95, 95, 95));

    jButton15.setIcon(icon6);

    URL url8 = ClassLoader.getSystemClassLoader().getResource("images/7.jpg");
    ImageIcon icon7 = new ImageIcon(new ImageIcon(url8).getImage().getScaledInstance(95, 95, 95));
    jButton14.setIcon(icon7);

    URL url9 = ClassLoader.getSystemClassLoader().getResource("images/8.jpg");
    ImageIcon icon8 = new ImageIcon(new ImageIcon(url9).getImage().getScaledInstance(95, 95, 95));
    jButton13.setIcon(icon8);

    URL url10 = ClassLoader.getSystemClassLoader().getResource("images/9.jpg");
    ImageIcon icon9 = new ImageIcon(new ImageIcon(url10).getImage().getScaledInstance(95, 95, 95));
    jButton18.setIcon(icon9);

    URL url11 = ClassLoader.getSystemClassLoader().getResource("images/banniere.png");
    ImageIcon iconLogo = new ImageIcon(new ImageIcon(url11).getImage().getScaledInstance(277, 97, 277));
    jLabel1.setIcon(iconLogo);
}

```

```

jPanel3.setLayout(new GridLayout(1,1)); //on initialise une grille de 1 sur 1 pour le JPanel qui affiche le résultat
jPanel3.setSize(358,93);

JLabel Hello = new JLabel(" "); //nouveau JLabel
Hello.setFont(new Font("Impact", Font.PLAIN, 72)); //on initialise la police à 110

jPanel3.add(Hello); //ajout du JLabel dans le JPanel qui affiche le résultat

calcul.setSize(358,93);
calcul.setFont(new Font("Impact", Font.PLAIN, 72));
calcul.setForeground(java.awt.Color.red);
calcul.setHorizontalAlignment(JLabel.CENTER);

perdu.setSize(358,93);
perdu.setFont(new Font("Impact", Font.PLAIN, 72));
perdu.setForeground(java.awt.Color.red);
perdu.setHorizontalAlignment(JLabel.CENTER);

j.setSize(412,743);

p.setSize(412,743);

w.setSize(412,713);

rien.setSize(358,93);
rien.setFont(new Font("Impact", Font.PLAIN, 72));
rien.setForeground(java.awt.Color.red);
rien.setHorizontalAlignment(JLabel.CENTER);

l.setSize(412,743);

jLabel2.setSize(225,45);

}

p.setSize(412,743);

w.setSize(412,743);

rien.setSize(358,93);
rien.setFont(new Font("Impact", Font.PLAIN, 72));
rien.setHorizontalAlignment(JLabel.CENTER);

l.setSize(412,743);

jLabel2.setSize(225,45);

listel.add(1);
listel.add(2);
listel.add(3);
listel.add(4);
listel.add(5);
listel.add(6);
listel.add(7);
listel.add(8);
listel.add(9);
listel.add(10);

score=SelectScore; //On initialise le score du joueur afficher comme le Score qui est sélectionner par la requête
}
final public void affichageSelectScore() throws SQLException
{
connexion = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/calculatrice", "root", ""); //Connexion à la base de données
nom=Connexion.getName(); //On récupère le nom entré dans le champs de la première frame et on l'a met dans une variable appeler nom

//Préparation de la requête qui sélectionne le score du joueur
try(PreparedStatement stat = connexion.prepareStatement("SELECT Score FROM users WHERE Nom='"+nom+"'"))
{
    ResultSet resu=stat.executeQuery(); //On execute la requête et on l'a met dans un variable resu
    resu.next(); //On charge le résultat
    SelectScore= resu.getInt(1); //On met le résultat dans une variable SelectScore
    jLabel4.setText("Score = "+SelectScore); //On écrit dans le Label du Score, le score sélectionner par la requête
}
}

```

```
}

final public void EnregistreScore() throws SQLException
{
    connexion = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/calculatrice", "root", ""); //Connexion à la base de données
    nom=Connexion.Name;//On récupère le nom entré dans le champs de la première frame et on l'a met dans une variable appeler nom

    //Préparation de la requête qui met à jour le score du joueur dans la bdd quand il gagne
    try(PreparedStatement statement = connexion.prepareStatement("UPDATE users SET Score=? WHERE Nom=?"))
    {
        statement.setInt(1, score);//Le premier ? qui est dans la requête est définie par le Score du joueur
        statement.setString(2, nom);//Le deuxième ? qui est dans la requête est définie par le Nom du joueur
        int resultatset=statement.executeUpdate(); //On met l'exécution de la requête dans une variable de type int car on a le résultat qui est un chiffre ou un nombre
    }
}

@Override
public void enable(){ //fonction pour débloquer tout les boutons
    jPanel2.setEnabled(true);

    jRadioButton2.setEnabled(true);

    jButton4.setEnabled(true);
    jButton5.setEnabled(true);
    jButton6.setEnabled(true);
    jButton7.setEnabled(true);
    jButton11.setEnabled(true);
    jButton13.setEnabled(true);
    jButton14.setEnabled(true);
    jButton15.setEnabled(true);
    jButton16.setEnabled(true);
    jButton17.setEnabled(true);
    jButton111.setEnabled(true);
    jButton2.setEnabled(true);

}

@Override
public void disable(){ //Fonction pour bloquer tout les boutons
    jPanel1.setEnabled(false);

    jRadioButton2.setEnabled(false);

    jButton4.setEnabled(false);
    jButton5.setEnabled(false);
    jButton6.setEnabled(false);
    jButton7.setEnabled(false);
    jButton11.setEnabled(false);
    jButton13.setEnabled(false);
    jButton14.setEnabled(false);
    jButton15.setEnabled(false);
    jButton16.setEnabled(false);
    jButton111.setEnabled(false);
    jButton2.setEnabled(false);
}
```

```

}

public void affichageCalcul(int nb)
{ //Fonction pour afficher le résultat avec des images

for(int i=1;i<=nb;i++) //Boucle selon nb = nombre sur lequel on a cliqué
{
    URL url = ClassLoader.getSystemClassLoader().getResource("images/spiderman.png");//chargement de l'image
    ImageIcon test = new ImageIcon(new ImageIcon(url).getImage());//nouvelle icon et initialisation de l'image dans l'icon

    jPanel3.add(new JLabel(test));//nouveau Label dans le JPanel qui sert à afficher le résultat
}
jPanel3.repaint();//remplace le JPanel vide par les images
jPanel3.validate();//validation du remplacement
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
Generated Code

private void jButton14ActionPerformed(java.awt.event.ActionEvent evt) {
    jPanel3.removeAll();

    affichageCalcul(7);

    nombreProposer=7;
}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    jPanel3.removeAll();

    affichageCalcul(4);

    nombreProposer=4;
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

    jPanel3.removeAll();

    affichageCalcul(2);

    nombreProposer=2;
}

private void jButton15ActionPerformed(java.awt.event.ActionEvent evt) {
    jButton17.setEnabled(true);

    jPanel3.removeAll();

    affichageCalcul(6);

    nombreProposer=6;
}

private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {
    jButton17.setEnabled(true);

    jPanel3.removeAll();

    affichageCalcul(1);

    nombreProposer=1;
}

private void jButton12ActionPerformed(java.awt.event.ActionEvent evt) {
    jButton17.setEnabled(true);

    jPanel3.removeAll();//efface tout ce qu'il y a dans le JPanel

    affichageCalcul(0);

    jPanel3.add(jLabel1); //nouveau JLabel dans le JPanel qui sert à l'écriture du résultat
    jPanel3.repaint(); //remplacement du JPanel vide par l'image
    jPanel3.validate(); //validation du changement

    nombreProposer=0;
}

private void jButton17ActionPerformed(java.awt.event.ActionEvent evt) {

    essai=essai+1;
    jLabel3.removeAll();
    jLabel3.setText("Essai = "+essai);
    jRadioButton1.setEnabled(true);
}

```

```

if(vie==false)
{
    if(nombreAleatoire==nombreAleatoire|| nombreAleatoire==nombreProposer || nombreAleatoire==nombreProposer)
    {

        p.add(e);
        this.setContentPane(p);

        ScheduledExecutorService s = Executors.newSingleThreadScheduledExecutor(); //fonction qui permet aux tâches de s'exécuter avec du délais

        s.schedule(new Runnable()
        {//on initialise le timer à 0
            @Override
            public void run()
            {
                disable();

                jRadioButton1.setEnabled(true);
                jRadioButton1.setText("Rejouer");

                jRadioButton2.setEnabled(true);

                calcul.setText("GAGNER");

                score = score+1;
                jLabel14.removeAll();
                jLabel14.setText("Score = "+score);

                jPanel3.removeAll();

                jPanel3.add(calcul);

                jPanel2.setVisible(true); //met visible le JPanel qui affiche la calculatrice
                w.setVisible(false); //rend invisible l'image de la pub
                try {
                    EnregistreScore(); //On exécute la requête qui enregistre le score quand le joueur gagne
                } catch (SQLException ex) {
                    Logger.getLogger(Calculator.class.getName()).log(Level.SEVERE, null, ex);
                }
            }
        },(long) 2.8, TimeUnit.SECONDS); //timer en seconde, initialiser à 10 secondes
        this.setVisible(true); //on met la JFrame en visible
    }
}
else
{
    vie=vie-1;

    p.add(l);

    this.setContentPane(p);

    ScheduledExecutorService s = Executors.newSingleThreadScheduledExecutor(); //fonction qui permet aux tâches de s'exécuter avec du délais

    s.schedule(new Runnable()
    {//on initialise le timer à 0
        @Override
        public void run()
        {
            if(vie==3)
            {
                calcul.setText(Calcul2);

                jPanel3.removeAll();

                jPanel3.add(calcul); //on ajoute le JLabel calcul dans le JPanel qui sert à l'écriture
                jPanel3.repaint(); //on remplace
                jPanel3.validate(); //on valide le remplacement

                jPanel2.setVisible(true); //met visible le JPanel qui affiche la calculatrice
                l.setVisible(false); //rend invisible l'image de la pub

                jLabel2.removeAll();

                jLabel2.setIcon(image_vie2);
                jLabel2.repaint();
                jLabel2.validate();

            }
            if(vie==2)
            {
                calcul.setText(Calcul2); //on initialise la police à 100

                jPanel3.removeAll();
            }
        }
    },(long) 2.8, TimeUnit.SECONDS);
}
}

```

```

        }

        if(vie==1)
        {
            if(nombreAleatoire>=nombreAleatoire2)
            {
                nombreBon=(nombreAleatoire-nombreAleatoire2);
            }
            else
            {
                nombreBon=(nombreAleatoire2-nombreAleatoire);
            }

            resultat=""+nombreBon;
            perdu.setText(resultat);
            disable();

            jRadioButton2.setText("Rejouer");
            jRadioButton2.setEnabled(true);
            jRadioButton1.setEnabled(true);

            jPanel13.removeAll();

            jPanel13.add(perdu); //on ajout le JLabel calcul dans le JPanel qui sert à l'écriture
            jPanel13.repaint(); //on remplace
            jPanel13.validate(); //on valide le remplacement

            jPanel2.setVisible(true); //met visible le JPanel qui affiche la calculatrice
            l.setVisible(false); //rend invisible l'image de la pub

            jLabel2.removeAll();

            jLabel2.setIcon(image_vie4);
            jLabel2.repaint();
            jLabel2.validate();
        }
    }

    public void run()
    {
        nombreAleatoire = (int) (Math.random() * ( 10 ));

        if(nombreAleatoire == 9)//on test si nombreAleatoire est égale à 9
        {
            nombreAleatoire2 = (int) (Math.random() * ( 2 ));//si il est égale à 9 on initialise nombreAleatoire entre 0 et 1
        }
        else if(nombreAleatoire == 8)
        {
            nombreAleatoire2 = (int) (Math.random() * ( 3 ));
        }
        else if(nombreAleatoire == 7)
        {
            nombreAleatoire2 = (int) (Math.random() * ( 4 ));
        }
        else if(nombreAleatoire == 6)
        {
            nombreAleatoire2 = (int) (Math.random() * ( 5 ));
        }
        else if(nombreAleatoire == 5)
        {
            nombreAleatoire2 = (int) (Math.random() * ( 6 ));
        }
        else if(nombreAleatoire == 4)
        {
            nombreAleatoire2 = (int) (Math.random() * ( 7 ));
        }
        else if(nombreAleatoire == 3)
        {
            nombreAleatoire2 = (int) (Math.random() * ( 8 ));
        }
        else if(nombreAleatoire == 2)
        {
            nombreAleatoire2 = (int) (Math.random() * ( 9 ));
        }
        else
        {
            nombreAleatoire2 = (int) (Math.random() * ( 10 ));
        }
    }
}

```

```

jPanel3.removeAll(); //effacer tout ce qu'il y a dans le JPanel

if(nombreAleatoire3<1){
    //on ajoute le calcul dans un nouveau JLabel
    if(nombreAleatoire<nombreAleatoire2)
    {
        operateur="-";
        signe=false;
        Calcul2=(nombreAleatoire2+operateur+nombreAleatoire+" = ");
    }
    else
    {
        Operateur="+-";
        signe=false;
        Calcul2=(nombreAleatoire+Operateur+nombreAleatoire2+" = ");
    }
}
else{
    operateur="+";
    signe=true;
    Calcul2=(nombreAleatoire+operateur+nombreAleatoire2+" = ");//on ajoute le calcul dans un nouveau JLabel
}

calcul.setText(Calcul2);

jPanel3.add(calcul); //on ajoute le JLabel calcul dans le JPanel qui sert à l'écriture
jPanel3.repaint(); //on remplace
jPanel3.validate(); //on valide le remplacement

jPanel3.setVisible(true); //met visible le JPanel qui affiche la calculatrice
j.setVisible(false); //rend invisible l'image de la pub

}

},10, TimeUnit.SECONDS); //timer de 10 secondes

this.setVisible(true); //on met le JFrame en visible

j.setVisible(true);
jPanel2.setVisible(false); //on rend invisible l'affichage de la calculatrice

```

```

private void jRadioButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    disable(); //désactivation de tout les boutons

    p.add(j);
    this.setContentPane(p);

    ScheduledExecutorService s = Executors.newSingleThreadScheduledExecutor(); //fonction qui permet aux tâches de s'exécuter avec du délais

    s.schedule(new Runnable()
    { //on initialise le timer à 0
        @Override
        public void run()
        {
            System.exit(0); //fermeture de l'application
        }
    },5, TimeUnit.SECONDS); //timer en seconde, initialiser à 10 secondes

    this.setVisible(true); //on met le JFrame en visible

    j.setVisible(true);

    p.add(jPanel2); //on ajoute le JPanel de la pub dans le JPanel de la calculatrice

    jPanel2.setVisible(false);
}

/**
 * @param args the command line arguments
 */
public static void main(String args[])
{
    /* Set the Nimbus look and feel */
    //
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html 

    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(Fenetre.class.getName()).log(Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(Fenetre.class.getName()).log(Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(Fenetre.class.getName()).log(Level.SEVERE, null, ex);
    } catch (UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(Fenetre.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```


Connexion et enregistrement :

```

package calculatrice;
import java.awt.Component;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.sql.ResultSet;
import java.util.logging.Level;
import java.util.logging.Logger;

import javax.swing.JLabel;
import javax.swing.JTextArea;
import javax.swing.JTextField;
/*
 * @author 33666
 */
public class Connexion extends javax.swing.JFrame {
    Connection connection = null;
    ResultSet resultSet = null;

    static String Name;//On initialise en static les variables qui nous serviront pour les requête sql car si on ne le fait pas SQL ne les accepte pas
    String nomSelect;
    static String ResuScore ;
    static String ResuName ;

    JLabel Score = new JLabel();

    public Connexion() throws SQLException{
        initComponents();
        affichageScore();
        affichageNom();
        jLabel7.removeAll();
        jLabel7.setText(ResuScore);
        jLabel6.removeAll();
        jLabel6.setText(ResuName);
    }

    final public void affichageScore() throws SQLException
    {
        connection = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/calculatrice", "root", ""); //Connexion à la base de données

        //Préparation de la requête qui sélectionne le Score le plus haut qui l'y a dans la base de données
        try(PreparedStatement stat = connection.prepareStatement("SELECT MAX(Score) FROM users")){
            {
                ResultSet resu=stat.executeQuery(); //On met l'exécution de la requête dans une variable resu
                resu.next(); //On charge le résultat
                ResuScore=resu.getString(1); //On inscrit le Score dans une variable ResuScore
            }
        }
        final public void affichageNom() throws SQLException
        {
            connection = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/calculatrice", "root", ""); //Connexion à la base de données

            //Préparation de la requête qui sélectionne le Nom du joueur qui a le score le plus haut
            try(PreparedStatement statem = connection.prepareStatement("SELECT Nom FROM users WHERE Score = '"+ResuScore+"'")){
                {
                    ResultSet result=statem.executeQuery(); //On met l'exécution de la requête dans une variable result
                    result.next(); //On charge le résultat
                    ResuName=result.getString(1); //On inscrit le Nom dans une variable ResuName
                }
            }
        }

        public void run() throws SQLException{
            affichageScore(); //On exécute la méthode qui affiche le Score du meilleur joueur
        }

        /** This method is called from within the constructor to
         * initialize the form.
         * WARNING: Do NOT modify this code. The content of this method is
         * always regenerated by the Form Editor.
         */
        @SuppressWarnings("unchecked")
        // Generated Code
    }
}

```

```

private void jButton17ActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        Name = jTextField1.getText(); //On inscrit le résultat du champs Nom dans une variable Name
        connection = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/calculatrice", "root", ""); //Connexion à la base de données
        //Préparation de la requête qui insert dans la base de données le nom du joueur qui s'inscrit si et seulement si il n'est pas déjà inscrit dans celle ci
        try(PreparedStatement stm = connection.prepareStatement("INSERT INTO users(Nom) SELECT ? WHERE NOT EXISTS (SELECT Nom FROM users WHERE Nom=?);")){
            ////
            stm.setString(1, Name); //Le premier ? qui est dans la requête est définie par le Nom du joueur rentrer dans le jTextField
            stm.setString(2, Name); //Le deuxième ? qui est dans la requête est définie par le Nom du joueur rentrer dans le jTextField
            int update =stm.executeUpdate(); //On met l'exécution de la requête dans une variable de type int car on a le résultat qui est un chiffre ou un nombre
        }
    }

    } catch (SQLException e) {
        System.err.format("SQL State: %s\n%s", e.getSQLState(), e.getMessage());
    } catch (Exception e) {
        e.printStackTrace();
    }
    this.setVisible(false); //On fait en sorte de mettre non visible la Frame de connexion quand on appuie sur OK quand on a renseigner le champ pour se connecter ou s'inscrire
    try {
        Calculator c = new Calculator(); //Après avoir renseignez son Nom ceci fait en sorte de créer une nouvelle calculatrice
        c.setVisible(true); //On la rend visible grâce à cette ligne
    } catch (SQLException ex) {
        Logger.getLogger(Connexion.class.getName()).log(Level.SEVERE, null, ex);
    }
}

public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    // Look and feel setting code (optional)
    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        @Override
        public void run() {
            try {
                Connexion connection;
                connection = new Connexion(); //On crée une nouvelle Frame de connexion
                connection.setVisible(true); //On la rend visible avec cette requête
            } catch (SQLException ex) {
                Logger.getLogger(Connexion.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton jButton11;
private javax.swing.JButton jButton12;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JPanel jPanel1;
private javax.swing.JTextField jTextField1;
// End of variables declaration
}

```

Mémoires

Base de données, serveurs et outils utilisés SGBD

En informatique, un système de gestion de base de données (abr. SGBD) est un logiciel système servant à stocker, à manipuler ou gérer, et à partager des informations dans une base de données, en garantissant la qualité, la pérennité et la confidentialité des informations, tout en cachant la complexité des opérations.

Un SGBD permet d'inscrire, de retrouver, de modifier, de trier, de transformer ou d'imprimer les informations de la base de données. Il permet d'effectuer des comptes rendus des informations enregistrées et comporte des mécanismes pour assurer la cohérence des informations, éviter des pertes d'informations dues à des pannes, assurer la confidentialité et permettre son utilisation par d'autres logiciels. Selon le modèle, le SGBD peut comporter une simple interface graphique jusqu'à des langages de programmation sophistiqués.

Ils sont utilisés pour de nombreuses applications informatiques, notamment les guichets automatiques bancaires, les logiciels de réservation, les bibliothèques numériques, les logiciels d'inventaire, les progiciels de gestion intégrés ou la plupart des blogs et sites web.

Les SGBD sont souvent utilisés par d'autres logiciels ainsi que les administrateurs ou les développeurs. Ils peuvent être sous forme de composant logiciel, de serveur, de logiciel applicatif ou d'environnement de programmation.

But visé :

Les SGBD sont les logiciels intermédiaires entre les utilisateurs et les bases de données. Une base de données est un magasin de données composé de plusieurs fichiers manipulés exclusivement par le SGBD. Ce dernier cache la complexité de manipulation des structures de la base de données en mettant à disposition une vue synthétique du contenu.

L'ensemble SGBD et base de données est destiné à permettre le stockage de données d'une manière offrant de nombreux avantages par rapport à un enregistrement conventionnel dans des fichiers. Il permet d'obtenir et de modifier rapidement des données, de les partager entre plusieurs usagers.

A l'aide du SGBD plusieurs usagers et plusieurs logiciels peuvent accéder simultanément aux données. Le SGBD effectue les vérifications pour assurer qu'aucune personne non autorisée n'ait accès à des données confidentielles contenues dans la base de données, il arbitre les collisions lorsqu'il y a plusieurs modifications simultanées de la même information et comporte des mécanismes en vue d'éviter des pertes de données à la suite d'une panne. Le SGBD effectue également sur demande des vérifications pour assurer que les données introduites soient correctes (valeurs dans les limites admises, format correct) et que les données soient cohérentes par rapport à ce qui se trouve déjà dans la base de données.

Fonctionnalités :

Un SGBD permet d'enregistrer des données, puis de les rechercher, de les modifier et de créer automatiquement des comptes rendus (anglais report) du contenu de la base de données. Il permet de spécifier les types de données, la structure des données contenues dans la base de données, ainsi que des règles de cohérence telles que l'absence de redondance.

Les caractéristiques des données enregistrées dans la base de données, ainsi que les relations, les règles de cohérence et les listes de contrôle d'accès sont enregistrées dans un catalogue qui se trouve à l'intérieur de la base de données et manipulé par le SGBD.

Voici une liste non-exhaustive des SGBD les plus connus, triés par catégories :

- **Système propriétaire** : Oracle Database, Microsoft SQL Server
- **Système libre** MySQL, PostgreSQL, MariaDB
- **Orienté objet** : ZODB, db4o
- **Embarqué** : SQLite
- **NoSQL** : Cassandra, Redis, MongoDB
- **Autre système** : Access, OpenOffice.org Base

WAMP

WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PhpMyAdmin pour gérer plus facilement vos bases de données.



DÉMARRER AVEC WAMPSERVER



Comme vous allez le voir, WampServer s'installe facilement et son utilisation très intuitive permet de le configurer très rapidement (sans toucher aux fichiers de configuration).

INSTALLATION

- ➊ Double-cliquez sur le fichier téléchargé et laissez vous guider. Tout est géré par l'installateur de WampServer. Par défaut, WampServer est livré avec les toutes dernières versions de Apache, MySQL et PHP.
- ➋ Une fois installé, vous pourrez ajouter manuellement des versions supplémentaires d'Apache, PHP ou MySQL (Uniquement compilées VC9, VC10 ou VC11). Les explications pour le faire vous seront données sur le forum.
- ➌ Chaque version de Apache, MySQL et PHP dispose de sa propre configuration et de ses propres fichiers (données pour MySQL).

Utilisez WampServer :

- ➍ Création automatique d'un répertoire « www » lors de l'installation (typiquement c:\wamp\www).
- ➎ Créez un sous répertoire pour votre projet et déposez-y vos fichiers PHP
- ➏ Cliquez sur le lien « Localhost » du menu de WampServer ou ouvrez votre navigateur préféré, et allez à l'adresse <http://localhost>

FONCTIONNALITÉS

WampServer dispose également d'un « TrayIcon » vous permettant de gérer et configurer simplement vos serveurs, sans toucher aux fichiers de configuration.

Clic gauche sur l'icône de WampServer, vous pouvez notamment :

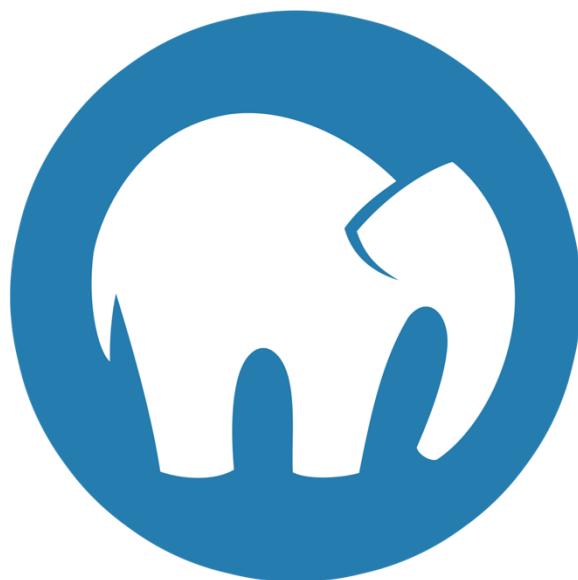
- ➊ Gérer les services de Apache et MySQL
- ➋ passer en mode online/offline (accessible à tous ou limité à localhost)
- ➌ Installer et changer de version de Apache, MySQL et PHP
- ➍ Gérer les paramètres de configuration de vos serveurs
- ➎ Accéder à vos logs
- ➏ Accéder aux fichiers de configuration
- ➐ Créer des alias...et bien plus encore

Clic droit :

- ➑ Changer la langue du menu de WampServer et accéder directement à cette page

MAMP

MAMP est un environnement de serveur local gratuit qui peut être installé en quelques clics sous MacOs et Windows. MAMP fournit tous les outils nécessaires pour exécuter WordPress sur un PC de bureau à des fins de test ou de développement, par exemple. Avec l'aide de leur serveur DNS local NAMO, il est même possible de tester facilement les projets sur des appareils mobiles. Peu importe si vous préférez Apache ou Nginx ou si vous voulez travailler avec PHP, Python, Perl ou Ruby.



LAMP

LAMP est un acronyme désignant un ensemble de logiciels libres permettant de construire des serveurs de sites web. L'acronyme original se réfère aux logiciels suivants :

- « Linux », le système d'exploitation (GNU/Linux) ;
- « Apache », le serveur Web ;
- « MySQL ou MariaDB », le serveur de base de données ;
- À l'origine¹, « PHP », « Perl » ou « Python », les langages de script.

Même si les auteurs de chacun de ces programmes ne se sont pas coordonnés pour construire des plates-formes LAMP, cette combinaison de logiciels s'est popularisée du fait du faible coût de l'ensemble et de la présence de tous ces composants dans la plupart des distributions GNU/Linux.

Cet acronyme a été inventé par Michael Kunze qui l'a utilisé pour la première fois en 1998 dans le magazine allemand c't. L'article en question voulait démontrer qu'un ensemble de logiciels libres pouvait concurrencer les offres commerciales disponibles.



Manuel d'installation LAMP à voir [ici](#)

XAMPP

XAMPP est un ensemble de logiciels permettant de mettre en place un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres (X (cross) Apache MariaDB Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus.

Il est distribué avec différentes bibliothèques logicielles qui élargissent la palette des services de façon notable : OpenSSL, Expat (parseur XML) PNG, SQLite, zlib... ainsi que différents modules Perl et Tomcat. Nombre de ces extensions étant inutiles aux débutants, une version allégée — version lite — est en conséquence aussi proposée.

Officiellement, XAMPP permet de configurer un serveur de test local avant la mise en œuvre d'un site internet, et son usage n'est pas recommandé pour un serveur dit de production.



Manuel d'installation XAMPP à voir [ici](#)

PhpMyAdmin

PhpMyAdmin (PMA) est une application Web de gestion pour les systèmes de gestion de base de données MySQL réalisée principalement en PHP et distribuée sous licence GNU GPL.

Il s'agit de l'une des plus célèbres interfaces pour gérer une base de données MySQL sur un serveur PHP. De nombreux hébergeurs, gratuits comme payants, le proposent ce qui évite à l'utilisateur d'avoir à l'installer.

Cette interface pratique permet d'exécuter, très facilement et sans grandes connaissances en bases de données, des requêtes comme les créations de table de données, insertions, mises à jour, suppressions et modifications de structure de la base de données, ainsi que l'attribution et la révocation de droits et l'import/export. Ce système permet de sauvegarder commodément une base de données sous forme de fichier .SQL et d'y transférer ses données, même sans connaître SQL.

Les requêtes SQL restent possibles, ce qui permet de les tester interactivement lors de la création d'un site pour les utiliser ensuite en batch (c'est-à-dire en différé) une fois au point.



Intégration continue d'une application

Introduction

Cet outil intervient dans le but de gagner du temps par l'automatisation de tâches récurrentes. Il est préférable de scripter l'ensemble de ces tâches et de laisser un outil les exécuter afin de ne pas créer d'erreurs lors du déploiement et des tests du projet informatique. Ce processus est généralement sans risque, car la plupart des outils fournissent une visualisation claire des actions en cours, et des éventuelles erreurs. C'est donc un outil qui s'intègre parfaitement dans une chaîne de déploiement continue ou d'intégration continue, les deux cas ayant pour but d'accélérer le processus de mise en production du projet informatique.

Si vous ne lancez vos tests qu'occasionnellement, l'utilisation d'un tel outil va vous permettre de rendre la chose systématique afin de rendre un projet de qualité supérieur. Une fois le changement effectué sur votre stack technique vous ne vous en passerez plus.

Tout comme la continuous delivery, l'intégration continue est avant tout privilégiée dans un environnement de développement logiciel agile. L'objectif de cette approche moderne est de progresser par étapes afin de concevoir le processus de développement plus efficacement et de pouvoir réagir aux modifications avec flexibilité. L'intégration continue a été évoquée pour la toute première fois dans la description de la méthode agile de l'Extreme Programming de Kent Beck. Mais a priori, l'idée d'une intégration continue serait antérieure. Elle intervient par exemple déjà dans la méthode Booch.

La continuous integration (en français « intégration continue ») est une technique de développement de logiciel agile. Pour ce type d'intégration, les développeurs intègrent les fragments de code terminés régulièrement, parfois plusieurs fois par jour, dans l'application au lieu de les intégrer tous en même temps à la fin du projet.

L'intégration continue nous fournit une bonne solution lorsque l'entreprise travaille sur un vaste projet ou un client souhaite avoir un logiciel à la fois complet et complexe. Différentes équipes travaillent à la conception de pans de l'application et les développeurs se chargent de programmer les différentes fonctionnalités. Après un travail de plusieurs mois voire de plusieurs années, l'intégralité du travail doit être regroupée et c'est alors que les problèmes surviennent. Dans un tel cas, la détection et la correction des erreurs, le regroupement de tous les fragments de code peut prendre plusieurs mois pour finalement se rapprocher de la phase de test finale et du déploiement.

Dans le cadre de la continuous integration, l'intégration du nouveau code est effectuée de façon bien plus précoce et pas uniquement lorsque toutes les parties prenantes ont terminé leur sous-domaine. Au lieu de cela, les développeurs intègrent leur code terminé une ou plusieurs fois par jour dans la mainline, le code source qui est accessible par tous les programmeurs. Étant donné qu'il s'agit toujours dans ce cas de sections de code relativement courtes, l'intégration est elle aussi plutôt rapide. Seules quelques minutes sont

nécessaires à un développeur pour mettre le résultat de son travail à disposition du reste de l'équipe. Si l'on découvre alors une erreur, elle peut être immédiatement localisée et, dans le meilleur des cas, corrigée rapidement.

Quand et pourquoi l'utiliser ?

L'outil d'intégration continue, est-il en existe plusieurs avec chacun leurs avantages, va donc intervenir directement auprès des développeurs informatiques leur permettant d'automatiser des tâches récurrentes et répétitives. Bien configuré, cet outil va alors permettre la construction du projet puis la mise en place sur un environnement de préproduction. C'est sûr ce premier environnement que le projet va alors subir une batterie de tests qui peuvent être à la fois fonctionnels et techniques. Suivant l'outil mis en place il va soit effectuer les tests, ou tout simplement les déclencher en appelant un autre outil de test.

C'est une fois le projet testé et validé, que l'outil d'intégration continue va alors pouvoir déplacer le projet sur l'environnement de production où il sera alors exploité à 100% par les utilisateurs. Il est d'ailleurs possible que sur cet environnement final, le projet informatique subisse également d'autres tests.

Concernant l'outil à proprement parlé, il en existe des dizaines et on généralement tous une particularité. Cela peut-être le langage supporté ou les options supplémentaires de l'outil comme une supervision de l'ensemble du process ou la mise en place de tests supplémentaire. Cet outil intervient dans le but de gagner du temps par l'automatisation de tâches récurrentes. Il est préférable de scripter l'ensemble de ces tâches et de laisser un outil les exécuter afin de ne pas créer d'erreurs lors du déploiement et des tests du projet informatique. Ce process est généralement sans risque, car la plupart des outils fournissent une visualisation claire des actions en cours, et des éventuelles erreurs. C'est donc un outil qui s'intègre parfaitement dans une chaîne de déploiement continue ou d'intégration continue, les deux cas ayant pour but d'accélérer le processus de mise en production du projet informatique. Si vous ne lancez vos tests qu'occasionnellement, l'utilisation d'un tel outil va vous permettre de rendre la chose systématique afin de rendre un projet de qualité supérieur. Une fois le changement effectué sur votre stack technique vous ne vous en passerez plus.

Quels sont les avantages et les inconvénients ?

Lors du travail quotidien, on constate souvent que l'intégration continue ne présente pas que des avantages en dépit de ses qualités. Si elle permet effectivement de faire l'économie d'une phase d'intégration longue et fastidieuse en fin de projet et de régler les problèmes de façon précoce, pour les équipes qui interviennent, le passage à l'intégration continue peut s'avérer très compliqué. Dans un tel cas, ce processus peut même demander plus de temps qu'il ne permet d'en économiser.

Avantages	Inconvénients
Possibilité de recherche précoce des erreurs	Conversion de processus habituels
Feedback permanent	Nécessite des serveurs et des environnements supplémentaires
Pas de surcharge contrairement à une seule grande intégration finale	Nécessité de mettre au point des processus de test adaptés
Enregistrement précis des modifications	Si plusieurs développeurs souhaitent intégrer leur code approximativement au même moment, des délais d'attente peuvent survenir
Disponibilité continue d'une version actuelle opérationnelle	
Nécessité d'un travail progressif	

Liste et comparatif des outils de déploiement

Introduction

Aujourd’hui, on trouve de nombreux outils d’intégration continue différents sur Internet. Ils ont tous vocation à aider les développeurs dans la mise en œuvre de l’intégration continue et y parviennent de différentes façons avec des fonctionnalités bien spécifiques. Mais les outils IC ne se distinguent pas uniquement par l’étendue de leurs fonctionnalités, on constate également de grandes différences en termes de prix et licence. Alors que bon nombre d’entre eux sont des logiciels open source disponibles gratuitement, certains fabricants proposent également des outils payants. Nous vous proposons un aperçu des programmes les plus appréciés et vous présentons leurs caractéristiques et leurs fonctionnalités.

❖ Jenkins



Jenkins

2019 Elections

- Governing Board
- Security Officer
- Events Officer

Le logiciel Jenkins est probablement l'un des outils IC les plus connus sur le marché. Depuis 2005 (à l'époque sous le nom de Hudson), le logiciel a été constamment amélioré. Aujourd'hui, ce logiciel programmé sous Java offre de nombreuses fonctionnalités et interfaces contribuant à faciliter non seulement l'intégration continue, mais aussi la livraison et le déploiement continus.

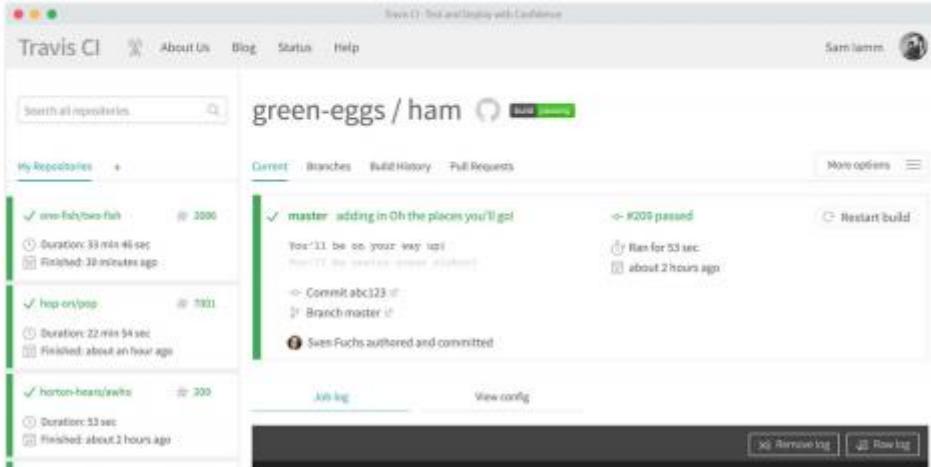
- Programmé sous Java
- Fonctionne dans un conteneur d'EJB
- Plus de 1 000 plugins
- Supporte également la livraison et le déploiement continus
- Peut-être combiné avec de nombreux systèmes de contrôle de version
- Contrôle via IGU (basée sur le Web), API REST ou commandes
- Hébergement sur le Cloud possible
- Gratuit
- Open source (licence MIT)

❖ Travis CI

Test and Deploy with Confidence

Easily sync your GitHub projects with Travis CI and you'll be testing your code in minutes!

 Sign Up



Les personnes travaillant avec GitHub préféreront certainement Travis CI, car cet outil CI fonctionne parfaitement avec ce système de gestion des versions populaire. Le logiciel est paramétrable à l'aide d'un simple fichier YAML hébergé dans le répertoire racine du projet de développement. GitHub informe Travis CI de chaque modification apportée dans le dépôt et maintient toujours le projet à jour.

- Programmé sous Ruby
- Fonctionne sur toutes les plateformes
- Fonctionne avec GitHub
- Configuration à l'aide d'un fichier YAML
- Gratuit pour les projets open source
- Pour les projets commerciaux, coût compris entre 69 \$ et 489 \$ par mois
- Open source (licence MIT)

❖ Bamboo



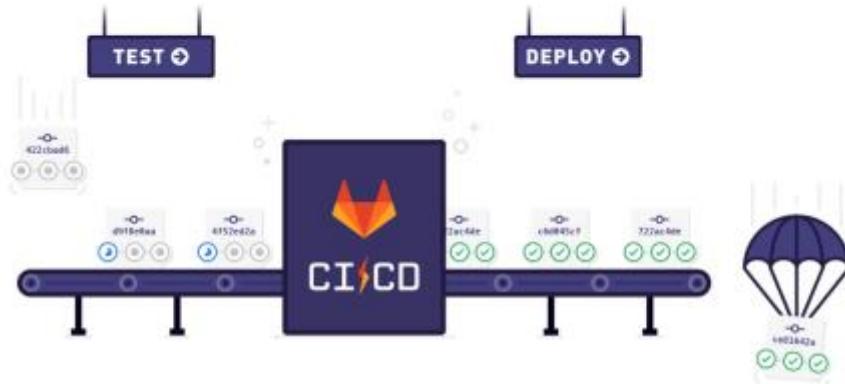
La société Atlassian, qui propose également aujourd’hui le service d’hébergement de fichiers Bitbucket, distribue depuis 2007 l’outil d’intégration continue Bamboo. À l’instar de Jenkins, Bamboo assiste les développeurs dans l’intégration mais offre également des fonctionnalités pour le déploiement et la gestion des versions. Le travail avec cet outil s’effectue via une interface en ligne simple.

- Programmé sous Java
- Fonctionne sur toutes les plateformes
- Intégration simple d’autres produits Atlassian
- Grande quantité d’extensions
- Plusieurs tests possibles en simultanée
- Communication via une interface Web et API REST
- Gratuit pour les projets open source, les organisations à but non lucratif et les classes scolaires
- Pour toute autre utilisation, coût unique entre 10 \$ et 110 000 \$ en fonction du nombre de serveurs utilisés

❖ GitLab CI

What Are The Advantages?

- **Integrated:** GitLab CI/CD is part of GitLab. You can use it for free on [GitLab.com](#)
- **Easy to learn:** See our [Quick Start](#) guide
- **Beautiful:** GitLab CI/CD offers the same great experience as GitLab. Familiar, easy to use, and beautiful.
- **Scalable:** Tests run distributed on separate machines of which you can add as many as you want
- **Faster results:** Each build can be split in multiple jobs that run in parallel on multiple machines
- **Continuous Delivery (CD):** multiple stages, manual deploys, environments, and variables
- **Open source:** CI/CD is included with both the open source GitLab Community Edition and the proprietary GitLab Enterprise Edition



GitLab CI est une composante du célèbre système de gestion des versions GitLab. En plus de l'intégration continue, GitLab offre un déploiement et une livraison continu. Tout comme pour Travis CI, la configuration de GitLab CI s'effectue via un fichier YAML. Par rapport à d'autres outils, le travail avec ce logiciel est également plus facile à d'autres égards.

- Composante de GitLab
- Programmé sous Ruby et Go
- Configuration à l'aide d'un fichier YAML
- Supporte également la livraison et le déploiement continu
- Open Core
- Auto-hébergement et hébergement sur le cloud disponible
- La version gratuite dispose uniquement de quelques fonctionnalités
- Le coût des autres versions est compris entre 4 \$ et 99 \$ par mois et par utilisateur

❖ Circle CI



L'outil d'intégration continue CircleCI fonctionne parfaitement avec GitHub et Bitbucket. Pour la phase de test, un conteneur ou une machine virtuelle sont utilisés. CircleCI accorde une grande importance à des processus de développement fluides, sans heurts, ce qui permet de mettre automatiquement à disposition des builds exempts d'erreur pour d'autres environnements.

- Configuration à l'aide d'un fichier YAML
- Supporte également le déploiement continu
- Auto-hébergement et hébergement sur le cloud disponible
- Fonctionne dans des conteneurs Docker, sous Linux VMs et MacOS VMs
- Gratuit pour un conteneur
- Autrement, coût compris entre 50 \$ et 3 150 \$ par mois

❖ CruiseControl



The screenshot shows the CruiseControl website. On the left is a sidebar with links: home, download, getting started, documentation, overview, config.xml, faq, wiki, contributing, developers, mailing lists, source repository, license, and Release: 2.8.4. The main content area has a header "CruiseControl". Below it is a paragraph about the tool's history and features, mentioning continuous integration, extensibility, and a web interface. A large screenshot of a web-based build status dashboard is displayed, showing a green bar at the top indicating a successful build ("ccn-testdrive passed (44 minutes ago)"), a list of recent builds on the left, and a "Latest Builds" section on the right with a red highlight on the most recent build ("about 17 hours ago"). At the bottom, there are links to Ant, NAnt, Maven, Phing, Bake, and Xcode, and a note about the BSD-style license.

L'outil CruiseControl fait partie des plus anciennes applications proposant une intégration continue. Cet outil a été introduit sur le marché dès 2001 et a été constamment amélioré depuis, notamment par Martin Fowler qui est l'un des pionniers dans le domaine de l'intégration continue. Outre un tableau de bord clair, les développeurs disposent également de nombreux plugins facilitant leur travail.

- Programmé sous Java
- Fonctionne sur toutes les plateformes
- Tableau de bord basé sur le Web
- Des versions pour Ruby (CruiseControl.rb) et .NET (CruiseControl.NET) sont disponibles
- Open source (licence BSD)
- Gratuit

❖ Codeship

Ship Smarter

Don't waste time and resources maintaining and scaling your local CI/CD infrastructure. Leave it to us!

Ship Faster

Customizable CI/CD automation speeds code changes into production. Build more, wait less!

Ship Better

Build and Test automation catches bugs before they reach your users. Ship your code with confidence!

L'outil IC Codeship appartient aujourd'hui à CloudBee qui dispose également de Jenkins dans son portefeuille. Ce programme est disponibles en deux versions : la version de base offre une interface Web facile d'utilisation alors que la version pro est configurée à l'aide de fichiers dans le dépôt. Les développeurs souhaitant travailler avec un conteneur Docker devront opter pour la version pro.

- Interface Web dans la version de base
- Fichiers de configuration dans le dépôt pour la version pro
- Docker supporté dans la version pro
- Gratuit pour 100 builds par mois en cas de pipeline test
- Coût compris entre 75 \$ et 1 500 \$ par mois

❖ TeamCity



Le logiciel TeamCity séduit avant tout par ses « gated commits » : grâce à ces derniers, TeamCity teste les modifications apportées au code avant même qu’elles ne soient insérées dans la mainline. Le code source est uniquement intégré au code base pour toute l’équipe lorsqu’il est exempt d’erreur. TeamCity effectue les tests de façon autonome en arrière-plan de telle sorte que les développeurs peuvent poursuivre leur travail dans l’intervalle.

- Programmé sous Java
- Fonctionne sur toutes les plateformes
- Gated Commits
- Gratuit pour 100 builds avec 3 agents de build
- Coût unique compris entre 299 € et 21 999 €
- 50 % de remise pour les start-ups et gratuit pour les projets open source

	déploiement continu supporté	hébergement sur le Cloud	licence	prix pour l'offre payante	version gratuite	particularité
Jenkins	✓	✓	MIT	-	✓	nombreux plugins
Travis CI	✗	✓	MIT	69-489 \$ par mois	✓	connexion directe à GitHub
Bamboo	✓	✓	propriétaire	coût unique de 10-110 000 \$	✓	
GitLab CI	✓	✓	MIT/EE	4-99 \$ par mois	✓	connexion directe avec d'autres produits Atlassian
Circle CI	✓	✓	propriétaire	50-3 150 \$ par mois	✓	utilisation simple
CruiseControl	✗	✗	BSD	-	✓	entièrement gratuit
Codeship	✓	✓	propriétaire	75-1 500 \$ par mois	✓	version de base et pro
TeamCity	✓	✗	propriétaire	coût unique de 299-21 999 €	✓	Gated Commits

Conclusion

Pour conclure, ce qu'il faut retenir de l'intégration continue c'est le test permanent du code tout au long du développement du projet. On retiendra que plus les bugs seront découverts tôt dans la phase de projet, moins le coût des correctifs sera important et plus grande sera la productivité. Néanmoins maintenir une plateforme d'intégration continue n'est pas toujours simple. Il faut savoir peser les pour et les contre : à savoir de détacher une ressource pour administrer l'outil et le faire évoluer en fonction des besoins.

Manuels

Manuel d'installation

Prérequis :

Avoir un serveur

Avoir un exécuteur de requête SQL

Trouver les fichiers :

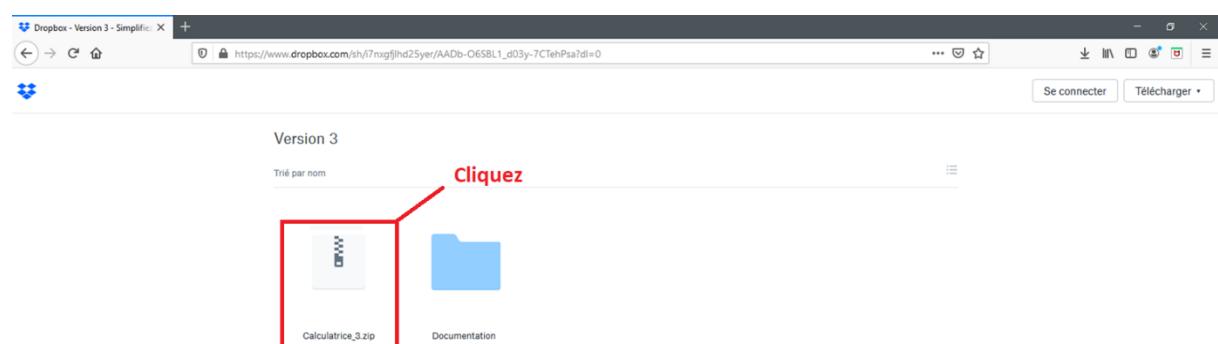
Entrez le lien suivant dans la barre de recherche :

https://www.dropbox.com/sh/i7nxgfjlhd25yer/AADb-O6SBL1_d03y-7CTehPsa?dl=0

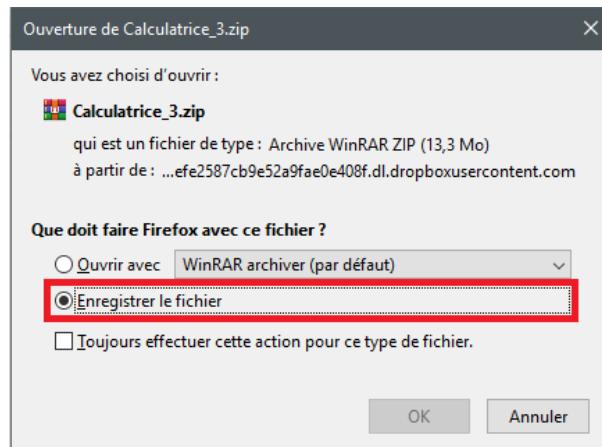
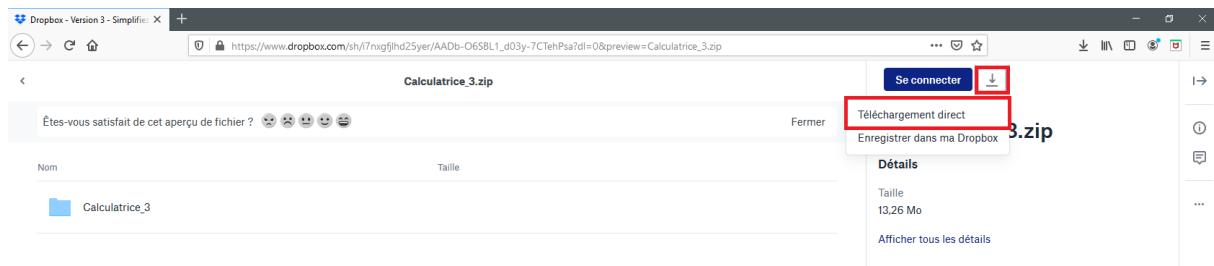
Vous arriverez ici :



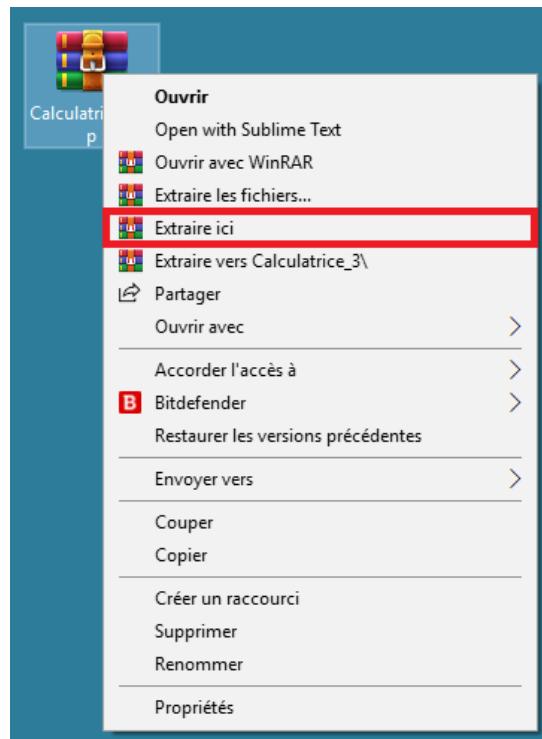
Cliquez sur le fichier Calculatrice_3.zip



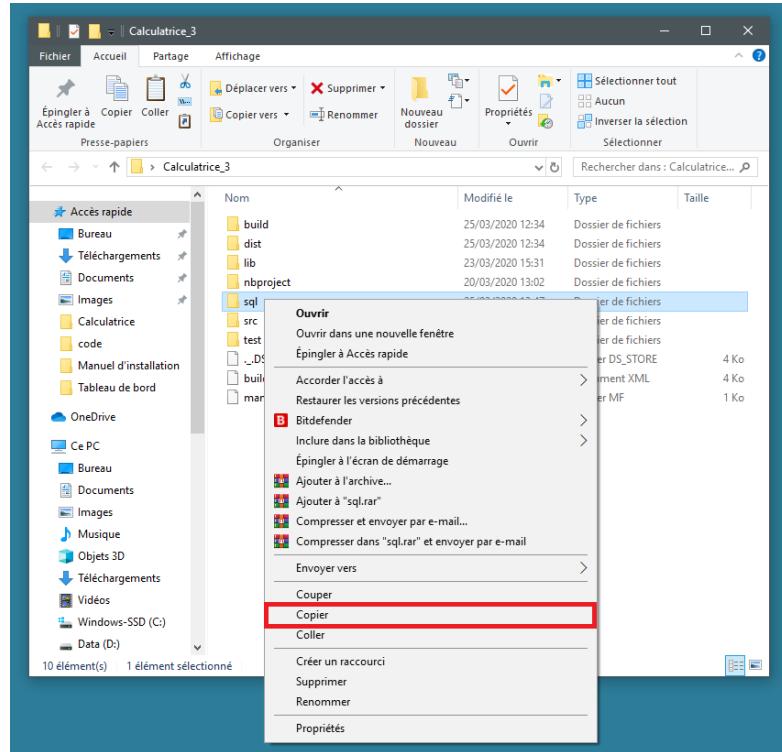
Télécharger le fichier en appuyant sur la flèche en haut à gauche puis « téléchargement direct »



Dézippez le dossier

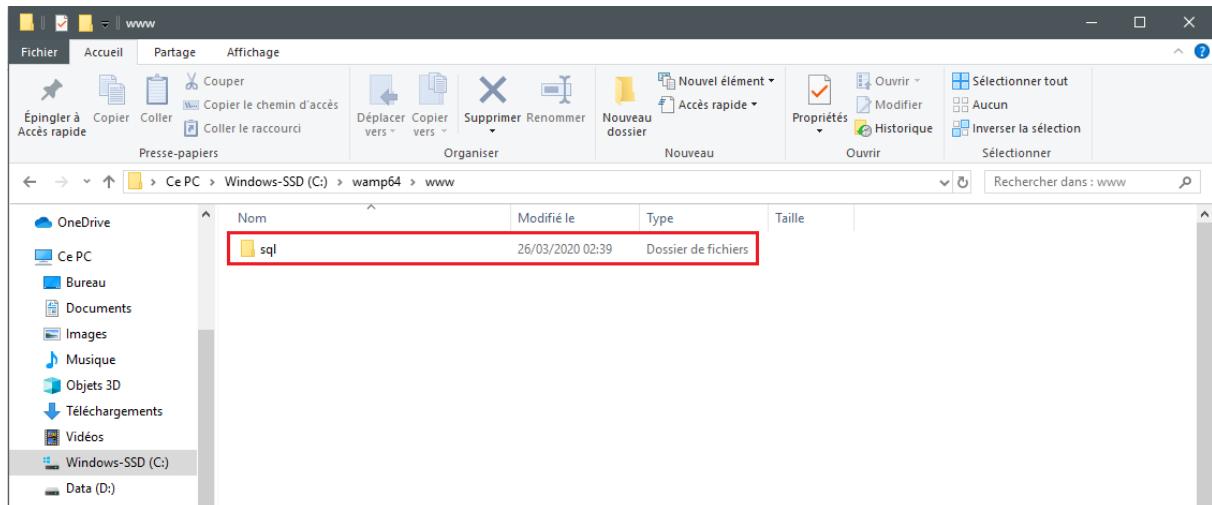


Ouvrez le dossier et copier tout le dossier « sql » (Ctrl+C ou clic-droit > copier)



Collez le dossier « sql » dans le dossier « www » de notre serveur local.

Chemin : C:\wamp64\www



Nous suivrons les commentaires présents dans chaque fichier .sql dans l'ordre suivant :

Version 3 > SQL

Trié par nom



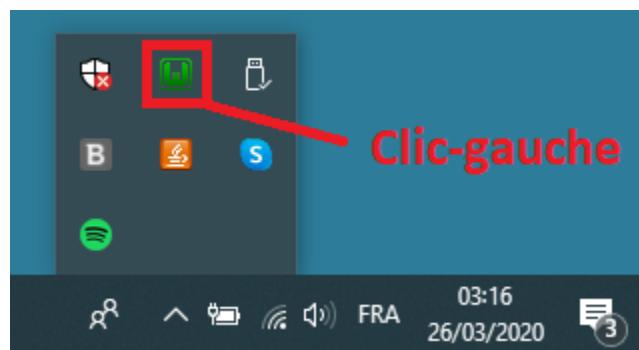
Creation_base.sql :

Cliquez sur le fichier

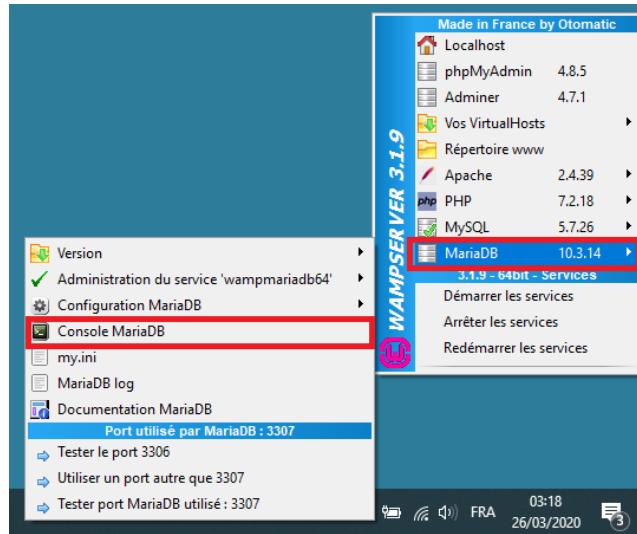
Le contenu suivant s'affichera :

```
1 CREATE database calculatrice CHARACTER SET 'utf8';
2
3 /*
4     Vous mettez le dossier sql dans dans votre dossier "www"
5     Vous lancez Wamp ou Mamp
6     Vous faites clique GAUCHE sur le petit logo quand il est en VERT
7     Vous allez dans la console MariaDB en allant sur le petit MariaDb et après vous allez pouvoir voir Console MariaDb
8     Puis vous vous connecter
9     Puis vous faites "SOURCE C:/wamp64/www/sql/Creation_base.sql"
10 */
```

« Vous faites clique GAUCHE sur le petit logo quand il est en VERT »



« Vous allez dans la console MariaDB en allant sur le petit MariaDb et après vous allez pouvoir voir Console MariaDb »



Entrez ensuite l'identifiant et le mot de passe

PS : L'identifiant par défaut sur Phpmayadmin est « root », il n'y a pas de mot de passe

Si vous n'avez pas configuré de mot de passe, appuyez juste sur la touche « Entrée » du clavier dans la console MariaDB

« Puis vous faites 'SOURCE C:/wamp64/www/sql/Creation_base.sql' »

```
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 57
Server version: 10.3.14-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> SOURCE C:/wamp64/www/sql/Creation_base.sql
Query OK, 1 row affected (0.010 sec)

MariaDB [(none)]>
```

Creation_table.sql :

```
1 CREATE TABLE users(
2     Nom VARCHAR(120) NOT NULL,
3     Score INT(11) DEFAULT 0,
4     PRIMARY KEY (Nom)
5 );
6
7 /*
8     Dès que vous avez créé la base vous faites "USE calculatrice;" 
9     Puis "SOURCE C:/wamp64/www/sql/Creation_table.sql"
10 */
```

« Dès que vous avez créé la base vous faites ‘USE calculatrice;’ »

```
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 57
Server version: 10.3.14-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> SOURCE C:/wamp64/www/sql/Creation_base.sql
Query OK, 1 row affected (0.010 sec)

MariaDB [(none)]> USE calculatrice;
Database changed
MariaDB [calculatrice]>
```

« Puis ‘SOURCE C:/wamp64/www/sql/Creation_table.sql’ »

```
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 57
Server version: 10.3.14-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> SOURCE C:/wamp64/www/sql/Creation_base.sql
Query OK, 1 row affected (0.010 sec)

MariaDB [(none)]> USE calculatrice;
Database changed
MariaDB [calculatrice]> SOURCE C:/wamp64/www/sql/Creation_table.sql
Query OK, 0 rows affected (0.030 sec)

MariaDB [calculatrice]>
```

Insertion_users.sql (initialisation d'un utilisateur test par défaut):

```
1 INSERT INTO users (Nom, Score) VALUES ('Anthony',0);
```

Exécutez la requête SQL suivante dans la console :

SOURCE C:/wamp64/www/sql/Insertion_users.sql

```
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 52
Server version: 10.3.14-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> SOURCE C:/wamp64/www/sql/Creation_base.sql
Query OK, 1 row affected (0.010 sec)

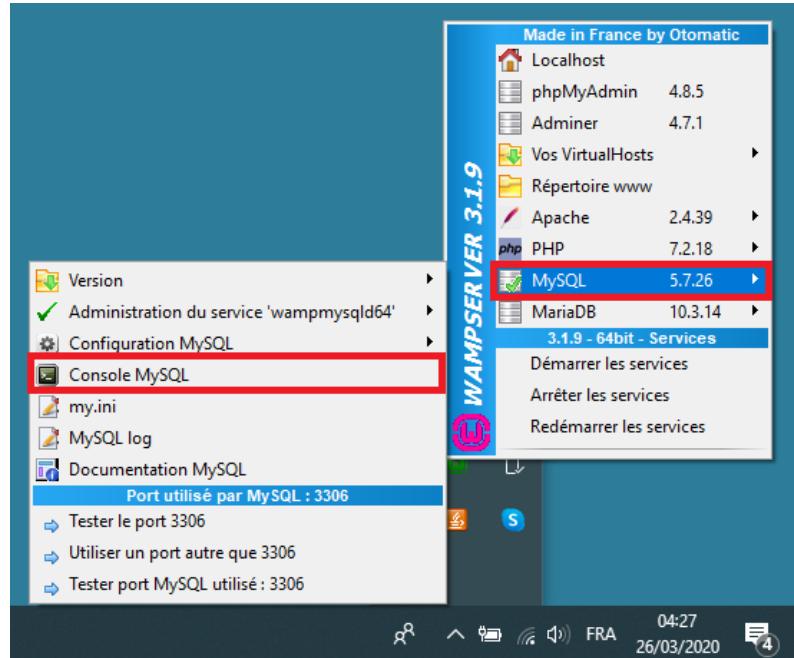
MariaDB [(none)]> USE calculatrice;
Database changed
MariaDB [calculatrice]> SOURCE C:/wamp64/www/sql/Creation_table.sql
Query OK, 0 rows affected (0.023 sec)

MariaDB [calculatrice]> SOURCE C:/wamp64/www/sql/Insertion_users.sql
Query OK, 1 row affected (0.016 sec)

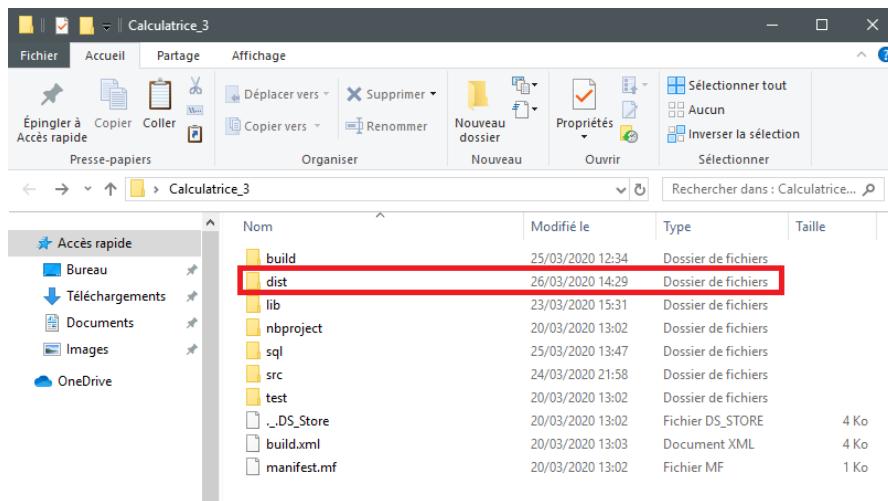
MariaDB [calculatrice]>
```

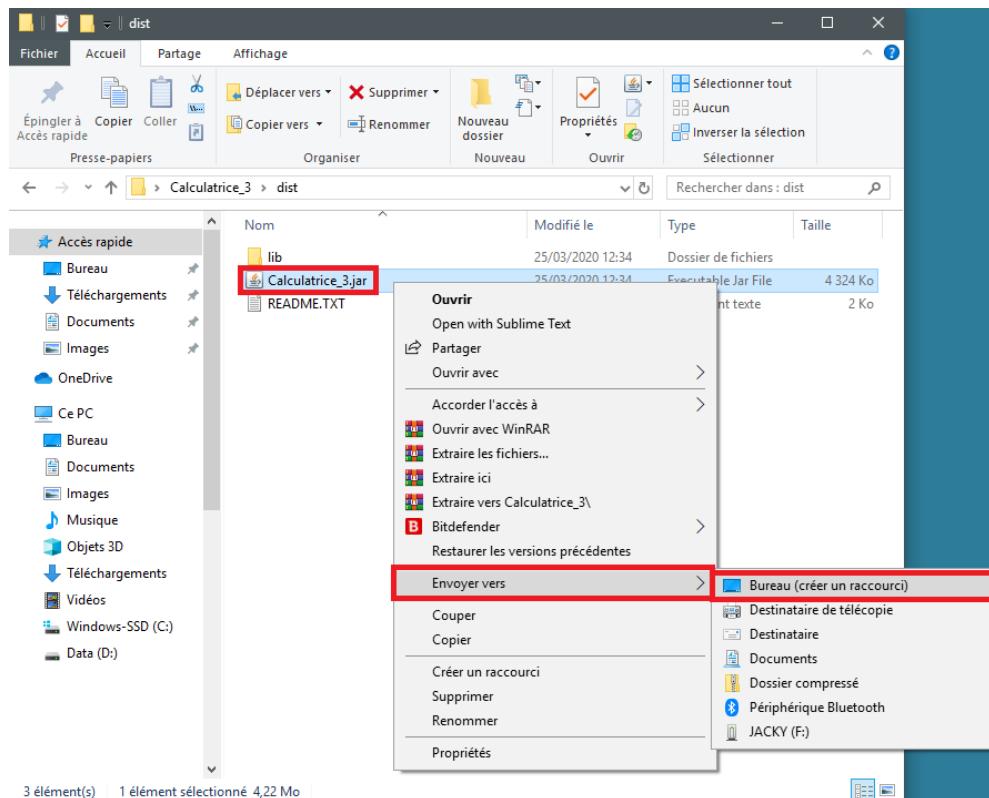
REPETER LES MEMES ETAPES MAIS AVEC MYSQL

SI LA CALCULATRICE NE S'ALLUME PAS



Revenez ensuite sur le dossier Calculatrice_3, allez dans dist et créez un raccourcie du fichier.jar





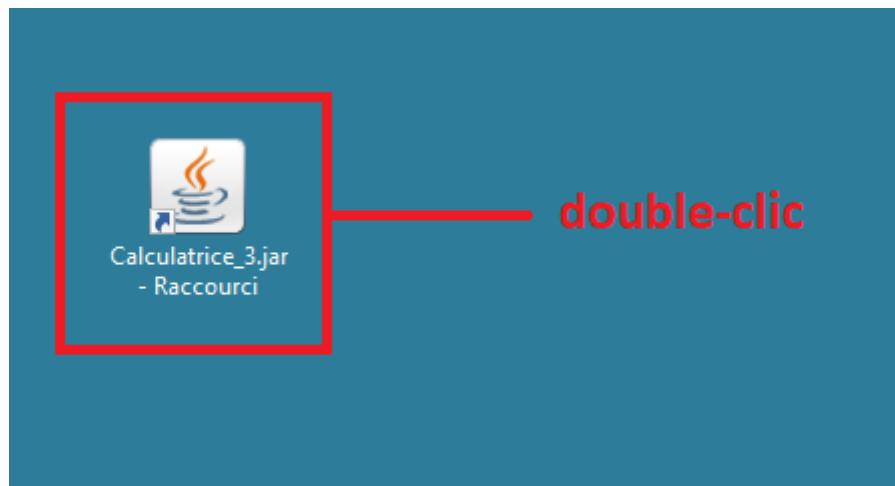
Le voilà sur le bureau et prêt à l'emploi



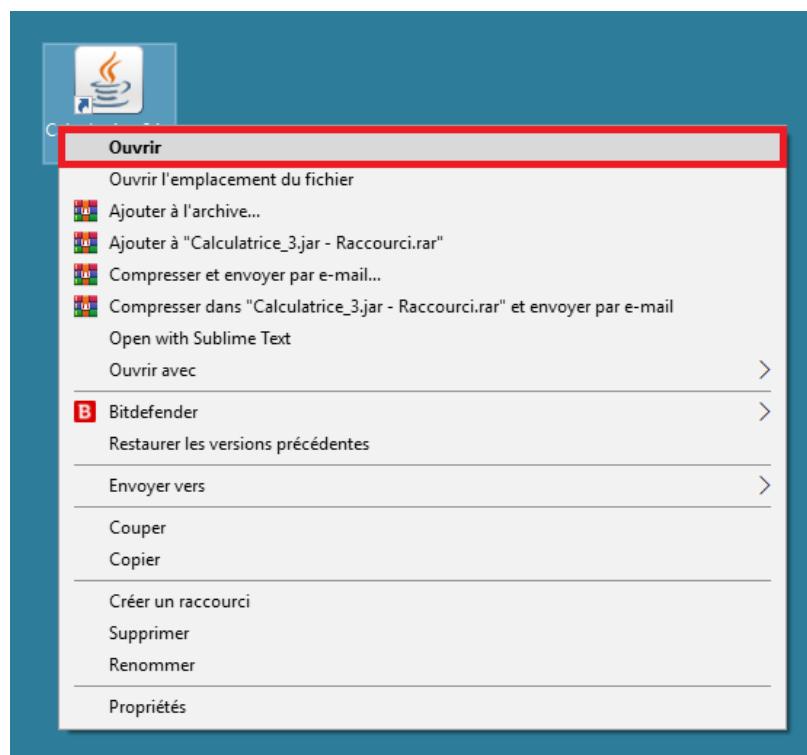
Manuel utilisateur

- **Allumer la calculatrice**

Allez sur le bureau, puis double-cliquer sur l'application, où bien faites un clic gauche puis appuyez sur «ouvrir»

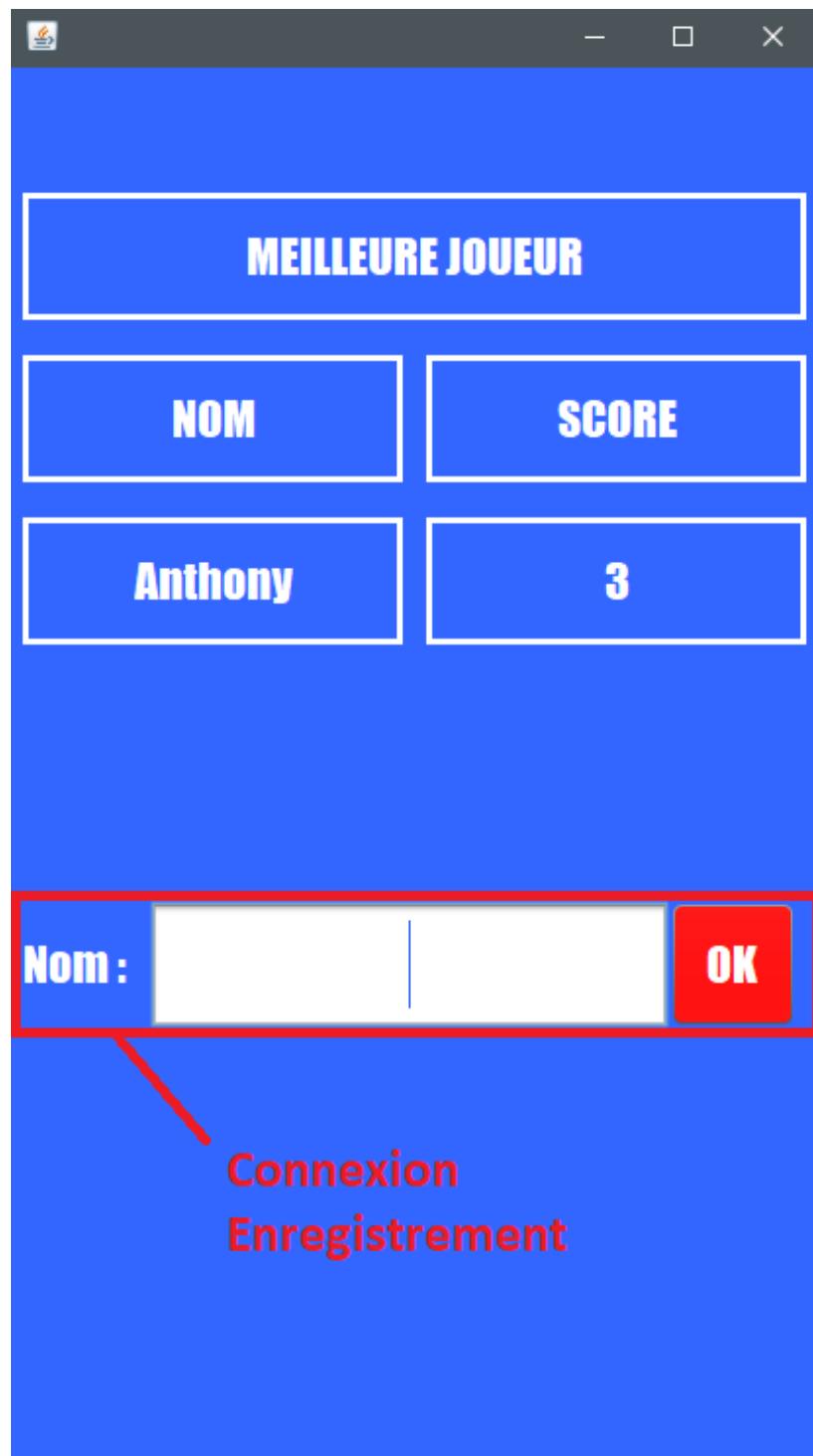


Ou

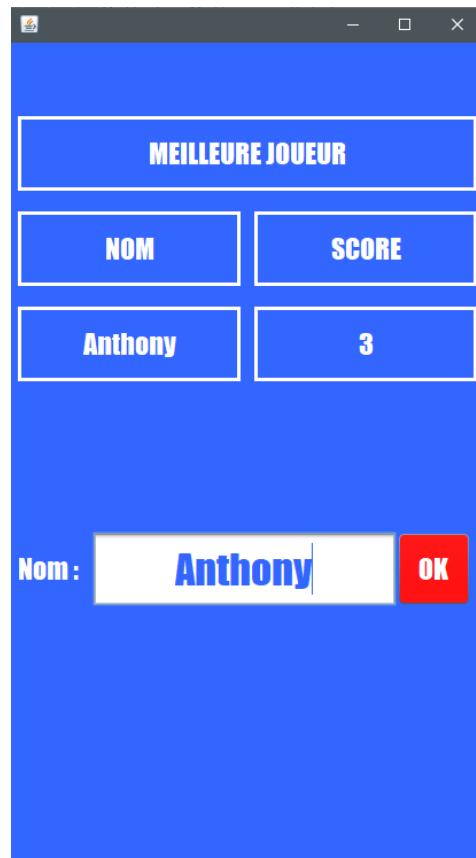


- **Une fois dans la calculatrice**

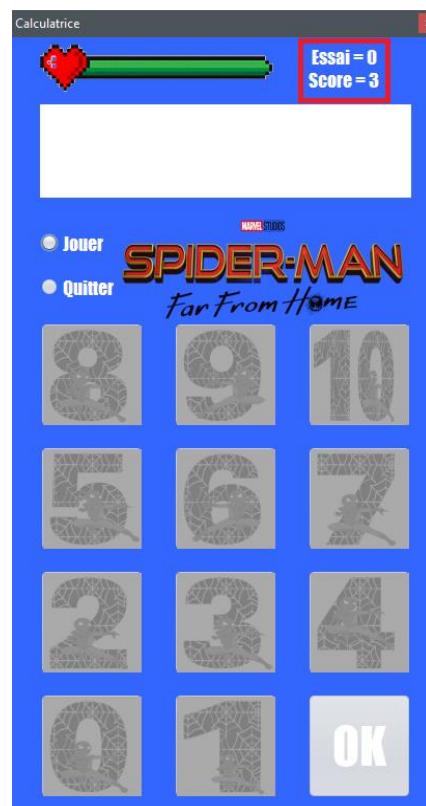
Vous arriverez sur cette interface : le nom et le score du joueur ayant répondu juste au plus grand nombre de calculs s'affichent. La barre de saisie permet de se connecter afin d'augmenter son score, ou d'enregistrer un nouveau joueur.



Nous allons augmenter le score de l'utilisateur « Anthony »



Appuyez sur « OK », vous arriverez sur une nouvelle interface



Le score affiché est le score du joueur « Anthony »
 « Essai » désigne le nombre d'essais dans une partie. Une bonne réponse comme une mauvaise réponse correspondent tous deux à un essai
 Appuyez sur « Jouer », une pub Spiderman s'affichera pendant 10 secondes



Après la pub, un calcul aléatoire sera généré, vous pourrez ensuite choisir votre réponse :

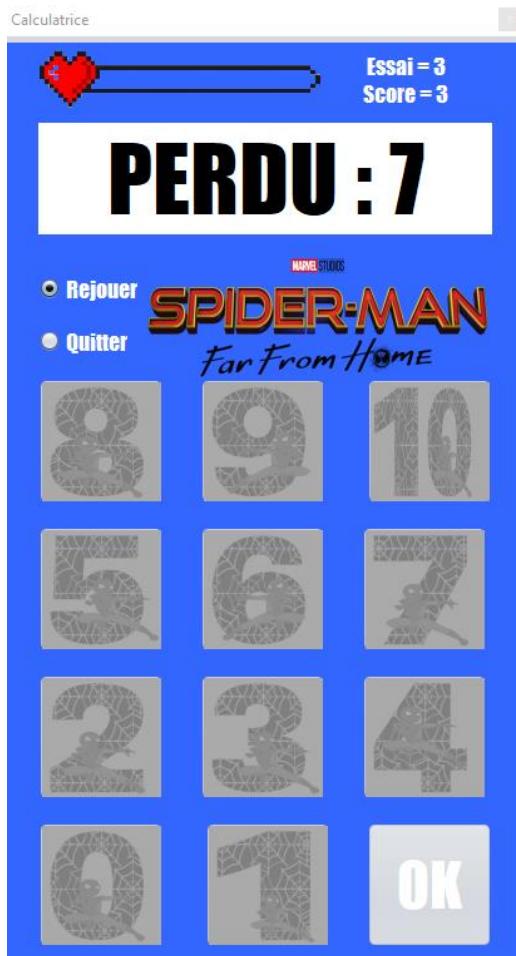
Two side-by-side screenshots of a math game interface. Both screens show a green progress bar with a heart icon at the start, labeled "Essai = 0 Score = 3".
 The left screen displays a math equation "7+0 =" in a white box. Below it is a grid of numbers from 0 to 10, each with a small spider-man logo on it. At the bottom right is a grey "OK" button.
 The right screen shows the same progress bar and grid. The "OK" button is now highlighted in orange, indicating it is the correct answer.
 Both screens include the "Jouer" and "Quitter" buttons at the bottom, and the "MARVEL STUDIOS SPIDER-MAN Far From Home" logo at the top.

Pour valider la réponse, appuyez sur le bouton « OK »

❖ Si vous ne répondez pas correctement



Si vous n'avez pas répondu correctement, vous aurez perdu une vie sur les trois qui vous sont proposées. Une fois que toutes les vies ont été épuisées, la calculatrice affichera la réponse du calcul.

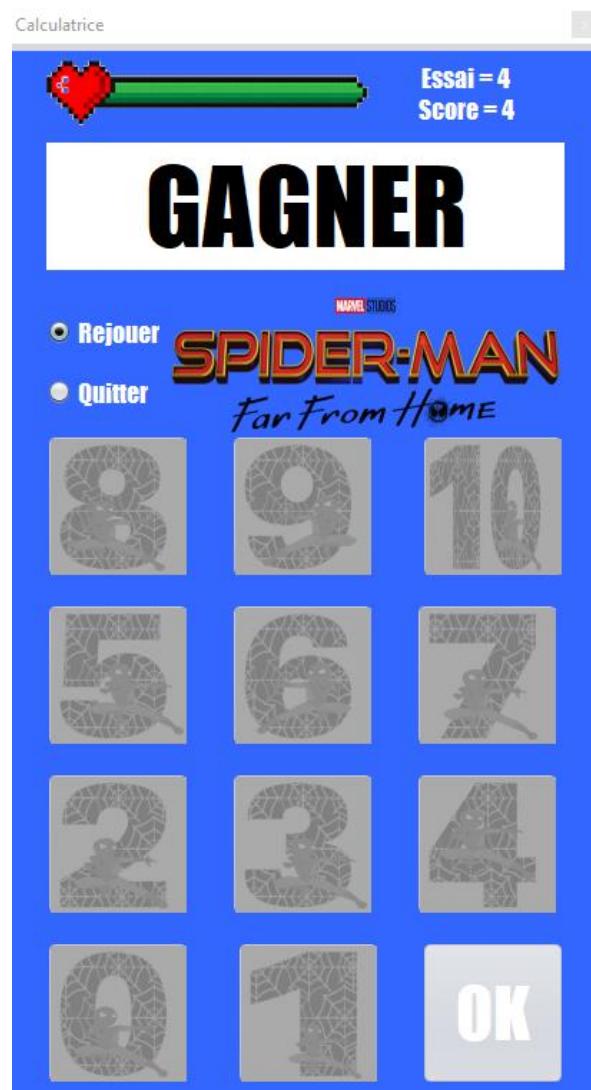


Comme vous pouvez le voir ici, le score n'a pas augmenté, et puisque nous avons donné 3 fois une mauvaise réponse, nous sommes actuellement à 3 essais pour cette partie.

-

Vous pourrez appuyer sur rejouer pour relancer la calculatrice ainsi qu'un nouveau calcul aléatoire, ou appuyer sur quitter pour l'éteindre

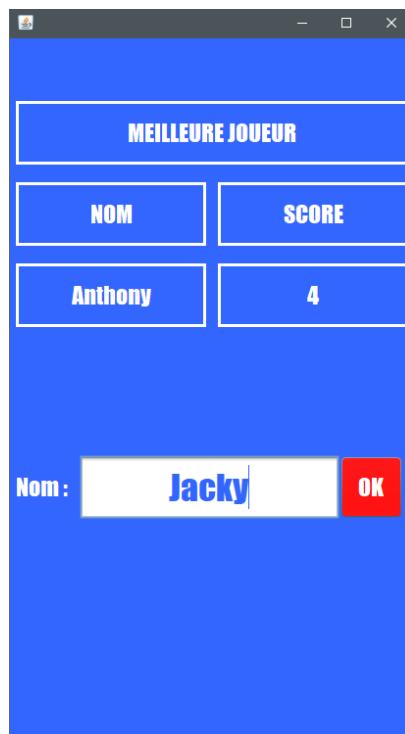
❖ Si vous avez gagné



Ici, nous avons donné une bonne réponse du premier coup, le score à augmenter de 1, les essais également

Vous pourrez appuyer sur rejouer pour relancer la calculatrice ainsi qu'un nouveau calcul aléatoire, ou appuyer sur quitter pour l'éteindre

Nous pouvons créer un nouveau joueur, qui sera enregistré dans la base de données



Si avec ce joueur nous faisons un score plus élevé, alors c'est ce joueur qui sera désigné comme celui ayant le plus haut score

MEILLEURE JOUEUR	
NOM	SCORE
Jacky	6
Nom:	<input type="text"/>

**IMPORTANT : Personne ne peut avoir le même nom,
et les noms doivent tous commencer par une
MAJUSCULE**

```
mysql> select * from users
      -> ;
+-----+-----+
| Nom   | Score |
+-----+-----+
| Jacky |     6 |
| Anthony |    4 |
+-----+-----+
2 rows in set (0.00 sec)
```

Le joueur « Jacky » s'est bien enregistré dans la base de données

Bilan

❖ Synthèse

Le développement de la version 3 de la calculatrice EditCALC était intéressante car elle ajoute les bases de données, un domaine que ne pouvons approfondir en plus des cours dessus.

❖ Les objectifs ont-ils été atteints ?

Oui, nous avons bien respecté le cahier des charges, mais avec un retard de 4 jours.

❖ En quoi le développement du projet nous a apporté quelque chose

Nous avons pu en découvrir plus sur l'interface Homme-Machine avec Swing et nous avons pu renforcer nos connaissances en JAVA et en base de données. Le jeu était complet en soit, mais, afin de rendre ce développement plus enrichissant, une base de données était incontournable. En effet, aujourd'hui, la grande majorité des applications possèdent une base de données.

❖ Quelles ont été les difficultés rencontrées

La plupart des nouvelles fonctionnalités étaient facile à faire et à intégrer. Mais l'intégration des requêtes SQL dans le code et l'affichage des valeurs spécifiques aux joueurs dans les différentes frames nous ont demandé plus de temps et de ressources. Ayant seulement une semaine par rapport à la version 2, nous devions nous organiser afin de mieux gérer ces risques. Nous nous sommes donc concentrés sur l'interface graphique avant d'y intégrer la base de données.

Tableau des évolutions

Version 1	Version 2	Version 3
<ul style="list-style-type: none"> - Création de la calculatrice - Affichage de la publicité au démarrage pendant 10 secondes - Calculs aléatoire avec résultat inférieur ou égal à 10 - Affichage de la réponse choisi sous forme d'image - Affichage de la réponse, bonne ou mauvaise pendant 10 secondes - Fermeture manuelle condamnée - Fermeture automatique 	<ul style="list-style-type: none"> - Système de vie - Animation pour féliciter l'élève et réciproque - Fermeture manuelle autorisée après un calcul - Possibilité de continuer - Affichage de la publicité pendant 5 secondes en sortie - Affichage de la réponse, bonne ou mauvaise pendant 10 secondes - Fermeture manuelle condamnée - Fermeture automatique 	<ul style="list-style-type: none"> -Le score et le nom du meilleur élève s'affiche - Mise à jour du meilleur élève lorsqu'un autre élève à un score supérieur au sien



Intégration



Elimination

Conclusion

Le développement de cette version 3 rend ce projet plus complet, en y intégrant une base de données et en travaillant sur les requêtes SQL. Cette version 3 était un vrai challenge à relever car elle devait être développée une semaine après la présentation de la V2. Nous devions y appliquer toutes les expériences acquises jusqu'à maintenant afin de délivrer l'application dans les délais.