

Examen – 2^{ème} Session
Documents de CM, TD et TP autorisés
Jeudi 19 juin 2025 - Durée 1h30

Un étudiant souhaite développer une application proposant une plateforme de description de jeux vidéo afin, dans un 1^{er} temps de décrire des informations sur des jeux, et dans un 2^{ème} temps de permettre des échanges entre des joueurs sur ces jeux. Un premier prototype avec quelques fonctionnalités a été implémenté.

L'application principale nommée « PlateformeJeux » comporte trois classes : la classe « Jeu » pour modéliser un jeu, la classe « LesJeux » pour modéliser l'ensemble des jeux et la classe « PlateformeJeux » qui est l'application principale (qui dérive de la classe « JFrame »).

Un jeu est décrit par son titre, son éditeur, sa catégorie qui est un mot-clé (Bataille, Aventure, Réflexion, Logique, etc.) et une photo gérée à l'aide d'une « ImageIcon ». L'ensemble des jeux est géré par une liste de type « ArrayList<Jeu> » dans la classe « LesJeux ».

L'interface du prototype de l'application est la suivante :



Elle comporte une liste nommée « ListeTitres » de type « JList » des titres des jeux en haut à gauche. Une zone d'édition nommée « Edition » de type « JTextArea » en bas à gauche. Une liste nommée « ListeCategories » de type « JComboBox » en haut à droite, avec en dessous un bouton nommé « BPhoto » de type « JButton », et en bas à droite, un bouton nommé « BTous » de type « JButton » contenant le texte « Tous les jeux ». Au centre, toutes les photos des jeux sont affichées sous forme d'une galerie de photos qui sont affichées sur des boutons cliquables.

L'annexe 1 décrit partiellement la classe « Jeu ». L'annexe 2 décrit partiellement la classe « Les Jeux ». L'annexe 3 décrit partiellement la classe principale et l'annexe 5 décrit la méthode « initComponents() » de la classe principale.

Classes « Jeu » et « LesJeux »

1. (1 pt) Donner le code de la méthode « `public String toString()` » de la classe « `Jeu` » en vous aidant de l'interface décrite ci-dessus.
2. (1 pt) Donner le code de la méthode « `public String toString()` » de la classe « `LesJeux` » en respectant la présentation de l'interface ci-dessus.
3. (1 pt) Expliquer en une phrase ce que fait la méthode « `public ArrayList<String> getCategories()` ».
4. (1 pt) Expliquer en une phrase ce que fait la méthode « `public LesJeux getJeuCategorie(String cat)` ».
5. (1,5 pts) Pour tester le programme, des jeux « tests » ont été insérés grâce à la méthode « `public void creationJeuTest()` » de la classe « `LesJeux` ». Donner la partie de code de cette méthode, permettant de créer un jeu dont le titre est « `League of Legends` », l'éditeur est « `Riot Games` », la catégorie est « `Bataille` », et dont le fichier de la photo est stocké dans le répertoire « `Img` » du répertoire « `src` » du projet en ressource. Le jeu créé devra être ajouté à la liste des jeux.

Classe principale « `PlateformeJeu` »

6. (3 pts) En vous aidant de la méthode « `public void initComponents()` » de la classe « `PlateformeJeu` » décrite en Annexe 5, donner sous forme d'une arborescence la description de l'interface avec les noms et les types des composants et si besoin le texte du composant, et les stratégies d'organisation (layout). Vous devez respecter les noms des composants utilisés dans la méthode.
7. (3 pts) Le constructeur de cette classe « `PlateformeJeu` », appelle 2 méthodes, pour remplir les listes.
 - a. « `initListeTitres()` » qui remplit la liste « `ListeTitres` » de type `JList` avec les titres des jeux. Donner le code de cette méthode.
 - b. « `initListeCategories()` » qui remplit la liste « `ListeCategories` » de type « `JComboBox` » avec les noms des catégories des jeux (sans doublons).
8. (3 pts) Le constructeur appelle également la méthode « `creationPanneauJeu()` » qui permet la création dynamique de la galerie de photos, sur des boutons cliquables. La méthode appelée lors d'un clic souris est la méthode « `public void actionPerformed(java.awt.event.ActionEvent evt)` ». Compléter le code de cette méthode « `private void creationPanneauJeu()` » pour créer les boutons de la galerie et afficher les photos des jeux. Les boutons seront numérotés en attribuant leur numéro à leur propriété « `Name` ».
9. (2,5 pts) Donner le code de la méthode « `public void actionPerformed(java.awt.event.ActionEvent evt)` » appelée lors d'un clic sur un bouton. Cette méthode doit récupérer le bouton cliqué, récupérer son numéro, récupérer le jeu correspondant, afficher les informations textuelles de ce jeu dans la zone d'édition nommée « `Edition` », et afficher sa photo sur le bouton nommé « `BPhoto` ».

On souhaite améliorer l'affichage de la galerie des photos des jeux en utilisant la classe « `BoutonImage` » donnée en Annexe 4, comme le montre l'interface ci-dessous.



10. (3 pts) Expliquer les éléments qui sont importants dans cette classe « `BoutonImage` », et les modifications à apporter à l'application principale, pour mettre en place cette amélioration.

ANNEXE 1
Classe « Jeu »

```
public class Jeu {
    private String titre;
    private String editeur;
    private String categorie;
    private Imagem photo;

    public Jeu (String titre, String editeur) {
        this.titre = titre;
        this.editeur=editeur;
        this.categorie="Autre";
        this.photo=new Imagem(getClass().getResource("/img/ImageDefault.jpg"));
    }

    public String getTitre() {return titre;}
    public void setTitre(String titre) {this.titre = titre;}
    public String getEditeur() { return this.editeur; }
    public String getCategorie() { return this.categorie; }
    public void setEditeur(String editeur) { this.editeur = editeur; }
    public void setCategorie(String cat) { this.categorie = cat; }
    public Imagem getPhoto() {return this.photo;}
    public void setPhoto(Imagem photo) {this.photo = photo;}

    public String toString() { // A COMPLETER }
}
```

ANNEXE 2
Classe « LesJeux »

```
public class LesJeux {
    private ArrayList<Jeu> lstJ;
    public LesJeux() { this.lstJ= new ArrayList<>();}

    public void ajoutJeu(Jeu j) { this.lstJ.add(j);}
    public Jeu getJeu(int i) { if (i>=0 && i< this.lstJ.size()) return this.lstJ.get(i); else return null;}
    public int getNbJeux() {return this.lstJ.size();}
    public String toString() { // A COMPLETER }
    public ArrayList<String> getCategories() // A EXPLIQUER
    { ArrayList<String> lst = new ArrayList<>();
      for (int i=0; i<this.lstJ.size(); i++)
      { Jeu je = this.lstJ.get(i);
        boolean trouve = false;
        for (int j=0; j< lst.size(); j++)
          if (lst.get(j).equals(je.getCategorie())) trouve = true;
        if (!trouve) lst.add(je.getCategorie());
      }
      return lst;
    }
    public LesJeux getJeuxCategorie(String cat) //A EXPLIQUER
    { LesJeux lst = new LesJeux();
      for (int i=0; i<this.lstJ.size(); i++)
      { Jeu je = this.lstJ.get(i);
        if (cat.equals(je.getCategorie()))
          lst.ajoutJeu(je); }
      return lst;
    }
}
```

```

public void creationJeuxTest(){ //A COMPLETER }
}

```

ANNEXE 3

Classe « PlateformeJeux » (description partielle)

```

public class PlateformeJeux extends javax.swing.JFrame {
    private LesJeux lesJ;
    public PlateformeJeux() {
        initComponents();
        this.lesJ = new LesJeux();
        this.lesJ.creationJeuxTest();
        initListeTitres();
        initListesCategories();
        creationPanneauJeux();
        Edition.setText(this.lesJ.toString());
    }

    private void initListeTitres(){ //A COMPLETER }
    private void initListesCategories() { // A COMPLETER }

    private void creationPanneauJeux()
    { int nbJ=this.lesJ.getNbJeux();
      int n=(int) (Math.sqrt(nbJ));
      PJeux.setLayout(new GridLayout(n,n));
      // A COMPLETER
    }

    public void btActionPerformed(java.awt.event.ActionEvent evt)
    { // A COMPLETER }
    ...
}

```

ANNEXE 4

Classe « BoutonImage »

```

public class BoutonImage extends JButton {
    private Image img;

    public BoutonImage() { super(); this.img=null;}
    public BoutonImage(Image im) {super(); this.img=im;}

    public Image getImage () { return this.img;}
    public void setImage (Image im) { this.img=im;this.repaint();}

    public void paint(Graphics g) {
        super.paint(g);
        if (this.img != null)
        { Image imgB = this.img.getScaledInstance(this.getWidth(),this.getHeight(), Image.SCALE_DEFAULT);
          this.setIcon(new ImageIcon(imgB));
        }
    }
}

```

ANNEXE 5

Classe « PlateformeJeu » - méthode « initComponents() »

```
private void initComponents() {
    PTitre = new javax.swing.JPanel();
    LTitre = new javax.swing.JLabel();
    PJeux = new javax.swing.JPanel();
    PActions = new javax.swing.JPanel();
    ListeCategories = new javax.swing.JComboBox<>();
    BPhoto = new javax.swing.JButton();
    BTous = new javax.swing.JButton();
    PActions2 = new javax.swing.JPanel();
    jScrollPane2 = new javax.swing.JScrollPane();
    ListeTitres = new javax.swing.JList<>();
    jScrollPane3 = new javax.swing.JScrollPane();
    Edition = new javax.swing.JTextArea();

    LTitre.setText("Plateforme de jeux vidéo");
    PTitre.add(LTitre);
    getContentPane().add(PTitre, java.awt.BorderLayout.NORTH);

    PJeux.setLayout(new java.awt.GridLayout(4, 4));
    getContentPane().add(PJeux, java.awt.BorderLayout.CENTER);
    PActions.setLayout(new java.awt.GridLayout(3, 1));
    PActions.add(ListeCategories);
    PActions.add(BPhoto);
    BTous.setText("Tous les jeux");
    PActions.add(BTous);
    getContentPane().add(PActions, java.awt.BorderLayout.EAST);

    PActions2.setLayout(new java.awt.GridLayout(2, 1));
    jScrollPane2.setViewportView(ListeTitres);
    PActions2.add(jScrollPane2);
    jScrollPane3.setViewportView(Edition);
    PActions2.add(jScrollPane3);
    getContentPane().add(PActions2, java.awt.BorderLayout.WEST);
    ...
    pack();
}
```