

# **Technologie Objet**

## **Idées de Projets**

### **Equipe 4**

Mathieu Malak  
Matthieu Trichard  
Ethan Boswell  
Nathan Albarede  
Othmane Elhour  
Benjamin Schlögel

### **Sommaire**

<b>Paint</b>	<b>2</b>
<b>Agenda électronique</b>	<b>3</b>
<b>Space Opera</b>	<b>4</b>
<b>Algorithme des Fourmis</b>	<b>5</b>

# Paint

## I\_ Présentation du projet

Création d'une application de dessin au pointeur (souris ou stylet) avec une interface graphique qui permettra de créer plusieurs dessins, parcourir nos dessins ou dessiner sur des images enregistrées. On implémenterait plusieurs fonctionnalités comme des formes géométriques (cercle, rectangle, ...) mais aussi des claques, pouvoir déplacer des objets etc.

Cette application pourra être utilisé pour du dessin, de la signature de documents, comme tableau blanc ou encore pour convertir le type de fichier d'une image connu.

## II\_ Intérêt pour la programmation

L'intérêt pour la partie programmation est que l'on apprendrait à créer une interface graphique avec plusieurs niveaux (accueil, affichage d'image, exploration de fichier,...). Mais aussi à apprendre la gestion d'image en Java, leurs créations, modifications, etc. Il sera nécessaire de gérer plusieurs tailles d'image mais aussi plusieurs formats (pdf, jpeg, ...). Il sera aussi important d'avoir une application fluide pour qu'elle soit utilisable.

# Agenda électronique

## I - Description du projet

L'idée est de créer un agenda électronique similaire à ce que l'on peut retrouver sur la plupart des ordinateurs / smartphones. Ce logiciel aura naturellement une interface graphique permettant à l'utilisateur de parcourir l'agenda à quatre échelles différentes (par jour, par semaine, par mois, par an) que ce soit dans le passé ou dans le futur. Différentes fonctionnalités que pourrait inclure cet agenda seraient :

- La possibilité de définir un événement, c'est-à-dire :
  - Un titre
  - Une description
  - Une date et heure de début et de fin
  - Une répétition (tous les jours, toutes les semaines...)
  - Un code couleur
  - Un tag
- Une fonction recherche à base de tags à ajouter à un événement
- Un bouton "retour" qui pourrait renvoyer l'utilisateur à la date actuelle s'il ne veut pas parcourir
- le calendrier dans l'autre sens
- Une prise en charge automatique de la date actuelle par rapport à la date définie sur l'ordinateur
- Sur l'interface graphique, un curseur indiquant la date d'aujourd'hui

## II - Intérêt pour la programmation

Le principal intérêt de ce projet sera la gestion des dates et leur implémentation graphique. Pour gérer correctement le calendrier, il faudra prendre en compte les années bissextiles, les jours de la semaine, et le fait que l'utilisateur puisse idéalement être capable de parcourir le calendrier jusqu'à une date arbitrairement éloignée (la question de la mémoire et de l'efficacité se pose donc).

L'interface graphique doit également être réactive : il faut pouvoir passer d'une vue à l'autre de manière rapide et ergonomique. Cependant, elle doit aussi afficher une grande quantité d'informations :

- Le jour / le mois / l'année
- Tous les événements visibles selon la vue actuelle : leur titre, description, couleur
- Le bouton retour et recherche
- Le curseur de la date d'aujourd'hui

Pour toutes ces raisons, on peut considérer que l'implémentation de cet agenda présente un intérêt notable sur le plan de la programmation.

# Shoot'em up

## I - Description du projet

Création d'un jeu type "shoot them up" : on contrôle un véhicule qui doit survivre à une grande quantité d'ennemis : pour cela, le joueur peut tirer des projectiles afin de détruire ces ennemis, et se déplacer pour éviter les tirs adverses.

Il y a plusieurs type de shoot'em up possible (nous réfléchissons encore lequel choisir) :

- Fixe (type Space invaders) : le joueur ne peut se déplacer que sur une seule dimension (par exemple une ligne horizontale), et les ennemis arrivent d'un seul bord de l'écran et se déplacent dans la direction du joueur.
- A défilement : le niveau défile au fur et à mesure, dévoilant de nouveaux ennemis, le joueur peut lui se déplacer en deux dimensions, mais ne peut pas effectuer de rotation.
- Multidirectionnel : le joueur se situe au centre de l'écran, et peut se déplacer dans toutes les directions (il peut s'orienter comme il le souhaite). Les ennemis peuvent arriver de tous les côtés.

Les sprites (ou lutin) pourront être de formes simples (un triangle pour le vaisseau du joueur, un carré pour les vaisseaux ennemis et une ligne pour les tirs) ou bien plus élaborés (image type pixel art par exemple).

Les possibles fonctionnalités de ce jeu sont :

- Système de vie
- Système de points, et de high-score
- Plusieurs niveaux ou bien un niveau "infini"

## II - Intérêt pour la programmation

La partie graphique de ce jeu peut être intéressante car assez complexe : si nous choisissons un jeu à défilement ou multidirectionnel, il faudra gérer le défilement d'un fond d'écran. De plus, il faudra apprendre à gérer des images (sprites).

Dans tous les cas, il faudra aussi gérer les contrôles (déplacement et tirs avec des touches du clavier, voir orientation avec le curseur de la souris si niveau multidirectionnel).

L'intelligence artificielle des adversaires sera simpliste : des déplacements vers le joueurs et des tirs à une certaine fréquence peuvent suffire. Peut-être pourrions-nous implémenter des déplacements aléatoires.

# Algorithme des Fourmis

## I - Description du projet

Cette proposition de sujet s'inspire du comportement des fourmis recherchant de la nourriture pour leur colonie à travers plusieurs chemins.

Dans un premier temps, une fourmi parcourt seule un territoire de manière aléatoire pour trouver une source de nourriture, puis une fois cette source trouvée elle retourne à la colonie (suivant un chemin plus ou moins long) pour chercher d'autres fourmis, pour exploiter cette source. Sur son chemin, cette fourmi y a déposé des phéromones volatiles pour guider les autres fourmis vers la source depuis le nid. Différents chemins de phéromone sont créés pour rejoindre la source.

L'objectif est d'emprunter le chemin le plus court. Cela est résolu grâce au dépôt des phéromones par chaque fourmi qui en déposent lors de chaque parcours.

## II - Intérêt pour la programmation

- Ce projet nous permettra de mettre en application une **Machine Learning** : Apprentissage supervisé (le modèle va au fur et à mesure réduire l'écart entre les résultats obtenus et attendus, c'est-à-dire trouver le chemin le plus court entre le Nid et la Nourriture).
- De plus, l'**interface graphique** utilisée sera conséquente :
  - tous les chemins à afficher
  - la quantité de phéromone présente sur les différents chemins
  - le déplacement des fourmis
  - le Nid et la Nourriture
  - une évolution de l'interface dans le temps
- Une bonne partie du projet utilise des probabilités : une fourmi est plus susceptible d'emprunter un chemin où il y a plus de phéromones qu'une autre.
- La création des chemins se fait en utilisant de l'aléatoire : pour trouver une source de nourriture et pour revenir au nid (pour appeler d'autres fourmis)