

Table des matières

Table des matières

Introduction

Résumé du cahier des charges

1. But de l'application
2. Fonctionnalités à réaliser
3. Matériel et logiciels nécessaires
4. Livrable

Méthodologie

Analyse fonctionnelle

Liste des fonctionnalités disponibles

Présentation de l'interface

L'icone

La fiche frmMain

La fiche frmAjoutSprite

La Fiche frmSceneJeu

Mesures de sécurité mise en place

Analyse des Méthodes

Classe Sprite

AjoutControlPanel - Sprite

SuprControlPanel - Sprite

UpdateValue - Sprite

pbx_MouseUp - Sprite

pbx_MouseDown - Sprite

pbx_MouseMove - Sprite

pbx_Paint - Sprite

pbx_Click - Sprite

Classe Jeu

AddSprite - Jeu

Annexes

le truc du div permet de faire une nouvelle page

Introduction

Cette documentation est un rapport destiné au collègue d'expert en charge d'évaluer le projet. Il permet de présenter les différents aspects de la conception du projet FlappySharp. Le projet a été réalisé dans le cadre du Travail Pratique Individuel (TPI) pour valider mon CFC d'informaticien.

FlappySharp est un éditeur de jeux 2Ds qui permet aux utilisateurs de faire des jeux 2Ds qu'il pourra ensuite essayer. Pour ce faire, l'utilisateur peut créer des sprites à partir d'images qu'ils peuvent placer où ils veulent.

Résumé du cahier des charges

1. But de l'application

Offrir à l'utilisateur de faire un jeu 2D facilement sans avoir besoin de coder. Il permet de faire un jeu en 2D du type flappy bird en posant les Sprites où on veut.

2. Fonctionnalités à réaliser

L'application permet de faire des jeux 2Ds du type flappy bird. Dès le lancement l'utilisateur peut créer un jeu 2D soit en utilisant les Sprites mis à disposition soit en créant des Sprites avec des images importées. À la fin de la réalisation de son jeu 2D l'utilisateur pourra jouer au jeu qu'il vient de créer. Il pourra aussi le sauvegarder pour y rejouer ultérieurement ou y faire des modifications.

3. Matériel et logiciels nécessaires

- Ordinateur (Pc)
- Visual studio 2019
- Typora

4. Livrable

- Planning
- Rapport de projet
- Journal de travail

Méthodologie

Pour planifier mon projet, je me suis basé sur la méthode en 6 étapes. Cette méthodologie consiste à diviser le projet en plusieurs étapes afin de faciliter la planification et le développement.

1. S'informer
2. Planifier
3. Décider
4. Réaliser
5. Contrôler
6. Évaluer

Analyse fonctionnelle

L'analyse fonctionnelle traite de la partie visible de l'application vue par l'utilisateur. Dans cette partie je parlerai en premier des fonctionnalités qui seront disponibles dans l'application. Par la suite les interfaces qui seront disponibles aux utilisateurs. Enfin, les mesures de sécurité mises en place pour qu'il n'y ait pas d'exception.

Liste des fonctionnalités disponibles

Voici la liste des actions possibles pour l'utilisateur du FlappySharp :

- Ajout de Sprite

L'utilisateur a deux possibilités d'ajouter un Sprite dans la scène :

- utiliser un des Sprites déjà existant : l'application propose déjà des Sprites qui sont par défaut. L'utilisateur peut donc les ajouter et les modifier.
- créer un Sprite : l'utilisateur peut aussi créer son propre Sprite. Lors de l'ajout une fenêtre s'ouvre et l'utilisateur doit indiquer les valeurs pour le Sprite tel que la position, la taille et les images.

- Modification des paramètres des Sprites

- modifier le nom du Sprite
- modifier la taille du Sprite
- modifier la liste d'images du Sprite
- modifier l'intervalle de temps pour l'animation du Sprite
- modifier le calque et le ZOrder du Sprite
- modifier la position du Sprite
- modifier le tag du Sprite
- modifier la rotation du Sprite

- Lancement du jeu :

Une fenêtre s'ouvre et le jeu se lance donnant ainsi un aperçu du projet qu'ils ont réalisé.

- Sauvegarder/Ouvrir le projet :

L'utilisateur peut sauvegarder son projet au format XML qu'il pourra par la suite ouvrir avec l'application.

Présentation de l'interface

L'interface utilisateur comprend un total de 3 formulaires WindowsForms. La première est la forme principale celle où l'utilisateur fait son jeu 2D avec tous les paramètres des Sprites et il peut créer des Sprites soit en utilisant des Sprites déjà existant soit en en créant de nouveau à la suite de ce choix une fenêtre s'ouvre et l'utilisateur indique dans les champs les valeurs pour le Sprite. Une fois le jeu fini par l'utilisateur il peut y jouer il lui suffit donc d'appuyer sur le Button "Run" pour ouvrir une fenêtre où il pourra jouer à son jeu. Toutes les images utilisées au sein de l'application sont des images totalement libres de droits et utilisables dans le cadre du TPI.

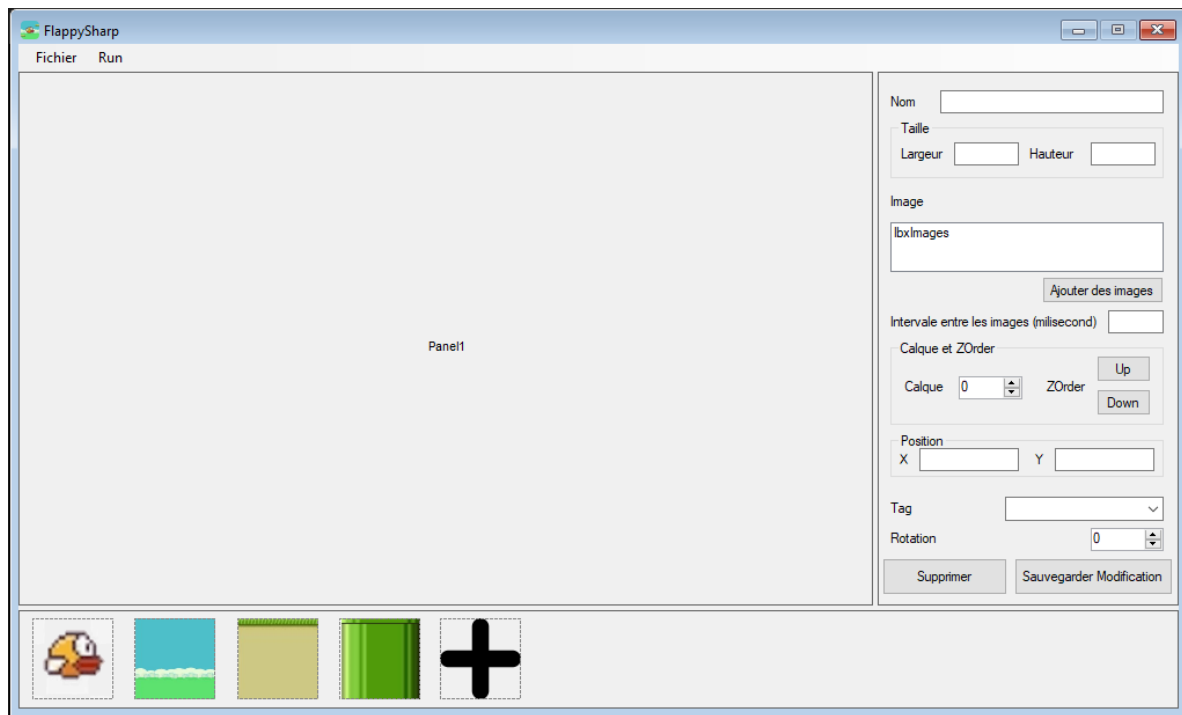
L'icone



L'icone de l'application a été récupéré sur Google Image à l'adresse : <http://d1png.com/png/6781141>

Je l'ai par la suite ajouter à l'application par le biais e propriétés du projet. L'icone est un flappy bird qui passe entre deux tuyau.

La fiche frmMain



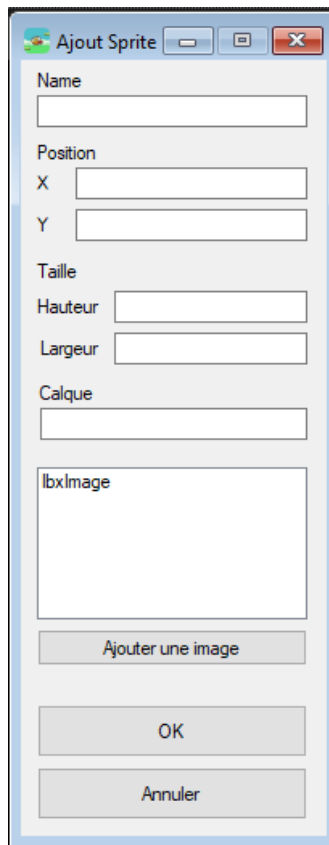
Le formulaire principale est la forme de création de jeu. Le formulaire contient plusieurs parties :

- La première partie qui est la scène de jeu ou on place les Sprites du jeu qu'on peut déplacer.
- Après, il y a le menuStrip qui permet de lancer le jeu en appuyant sur "Run" ce qui ouvrira une nouvelle fenêtre avec le jeu que l'utilisateur à créer. Il permet aussi de sauvegarder son jeu au format XML en utilisant la sérialisation (le format XML a été choisie ici pour que l'utilisateur puisse directement changer les valeurs de ces Sprites sans avoir besoin de lancer l'application) et d'ouvrir un projet au format XML en utilisant la désérialisation.
- Ensuite, il y a la partie de paramétrage des Sprites qui se met à jour quand on clique sur un Sprite. Cette partie contient six TextBox qui permettent de modifier des paramètres tel que le nom du Sprite, la taille du Sprite, que ce soit en largeur ou en hauteur, le délai entre les images que l'utilisateur a choisies pour le Sprite qui est désactivé si le Sprite ne contient qu'une seule image et enfin la position du Sprite. Cette partie contient aussi des numeriqueUpDown qui sont utilisés pour la rotation du Sprite ainsi que pour définir le calque du Sprite ainsi que d'une listBox pour afficher les images que contiennent les Sprites et d'un Button qui permet d'ajouter des images aux Sprites, de deux Button pour changer le ZOrder de chaque Sprite qui permet de

changer dans un même calque la profondeur des Sprites et enfin de deux Button qui permettent de sauvegarder les modifications faite sur le Sprite et d'un Button de suppression du Sprite qui supprime totalement le Sprite.

- Enfin, il y a la partie ajout de Sprite qui contient des Sprites déjà prédéfinie qui se mettent automatiquement sur la scène de jeu avec des paramètres par défaut que l'utilisateur peut ensuite modifier. Il y a aussi le PictureBox qui contient un plus qui ouvre une nouvelle fenêtre qui permet de créer un Sprite celons les volontés de l'utilisateur en spécifiant des paramètres.

La fiche frmAjoutSprite



The image shows a Windows-style dialog box titled "Ajout Sprite". It has a standard title bar with minimize, maximize, and close buttons. The dialog contains several input fields: "Name", "Position" (with sub-fields for "X" and "Y"), "Taille" (with sub-fields for "Hauteur" and "Largeur"), and "Calque". Below these fields is a PictureBox control labeled "lbImage". Under the PictureBox is a button labeled "Ajouter une image". At the bottom of the dialog are two buttons: "OK" and "Annuler".

La forme frmAjoutSprite s'ouvre après que le Button "Run" menuStrip de la forme frmMain soit pressé et elle permet de créer un Sprite en indiquant dans les champs les valeurs qu'aura le Sprite. Les valeurs que l'utilisateur peut définir pour le Sprite sont le nom ne peut pas être le même si le nom choisi est le même qu'un déjà existant un chiffre s'ajoute à la fin du nom le rendre unique, la position (X et Y), la taille (Hauteur et Largeur), le calque où il sera et le ZOrder se met automatiquement pour

qu'il n'y ait pas deux Sprite sur le même ZOrder et enfin les images qui permettent de faire une animation du Sprite. Il y a ensuite deux Button qui permettent à l'utilisateur d'ajouter le Sprite ou de stopper l'ajout du Sprite.

La Fiche frmSceneJeu

Mesures de sécurité mise en place

Voici une liste des contrôles mis en place pour éviter tout éventuelles mauvaises manipulations des utilisateurs et d'empêcher les erreurs dans le code.

- La taille des fenêtres est fixe pour ne pas avoir de problème d'affichage
- Un traitement des inputs a été fait sur les TextBox qui ne doivent pas contenir de texte
- L'intervalle entre les images n'est pas accessible s'il n'a pas plus d'une image
- Les bouton "Supprimer" et "Sauvegarder les modifications" sont inaccessibles si aucun Sprite n'est sélectionné
- On ne peut pas ajouter de Sprite tant que tous les champs ne sont pas remplis

Analyse des Méthodes

Classe Sprite

Cette classe qui hérite de PictureBox est utilisé pour créer les Sprites. Pour la création d'un Sprite le constructeur reçoit des paramètres

- string Nom
- Size Taille

- Dictionary<string,Bitmap> Images
- int Calque
- int ZOrder
- Point Position
- Panel ZoneScene (ou les sprites sont ajouter)

Cette classe est utilisée dans la classe Jeu.

AjoutControlPanel - Sprite

Cette méthode déclarée en public a pour but de mettre le sprite (PictureBox) dans les controls du panel (ZoneScene). Elle est appelée dans le constructeur du Sprite et dans la méthode RefreshControl() de la classe Jeu.

SuprControlPanel - Sprite

Cette méthode déclarée en public a pour but de supprimer le Sprite. Elle est appelé par la méthode RefreshControl() et par btnSupr_Click(). Elle reçoit un bool comme paramètre. Si le bool est false la méthode supprime le Sprite seulement dans les controls du panel, mais si le bool est true la méthode supprime totalement le Sprite.

UpdateValue - Sprite

Cette méthode déclarée en public a pour but de mettre à jour les valeurs du Sprite. Elle est appelée par la méthode UpdateValueSpriteSelected() dans la classe Jeu. Elle reçoit en paramètre un nom (string), une taille (Size), des images (Dictionary<string,Bitmap>), un interval entre les images (int), un calque (int), un ZOrder (int), un tag (string), une rotation (int) qu'il va ensuite mettre dans les bonnes variables.

pbx_MouseUp - Sprite

Cette méthode se produit sur le relâchement du click de la souris sur Sprite. Le paramètre sender permet de savoir sur quel Sprite le relâchement du click a été fait par la suite on vérifie si le Sprite existe si non on arrête la méthode sinon on change la variable dragging a false et on crée un nouveau rectangle qu'on implémente à la variable collision.

pbx_MouseDown - Sprite

Cette méthode se produit sur l'appui constant de la souris sur le Sprite. Le paramètre sender permet de savoir sur quel Sprite le click est appuyer par la suite on vérifie si le Sprite existe si non on arrête la méthode après on met la variable dragging a true et on mets à jour les variables xPos et yPos avec la position X et Y du Sprite.

pbx_MouseMove - Sprite

Cette méthode se produit au moment du déplacement de la souris. La méthode permet de vérifier si le sender n'est pas null s'il ne l'est pas il fait bouger le Sprite en fonction de la position de la souris.

pbx_Paint - Sprite

Cette méthode se produit au moment où le rafraîchissement est demandé. Le PaintEventArgs dessine un rectangle au tour du Sprite qui permettra par la suite de faire la collision entre deux Sprites.

pbx_Click - Sprite

Cette méthode se produit au moment où on click sur le Sprite. On met la variable selected a true qui est utilisé pour savoir qu'elle Sprite est sélectionné.

Classe Jeu

[Description à faire pour la classe jeu]

AddSprite - Jeu

Cette méthode déclarée en public a pour but de créer un Sprite à partir des valeurs reçues en paramètre. La classe reçoit les paramètres elle calcule automatiquement le zorder du Sprite pour que ce soit le dernier Sprite du Calque pour qu'il n'y ait pas deux Sprite sur le même. Au moment de l'ajout du Sprite en appelant le constructeur de la classe Sprite on appelle la méthode CheckNomExist() pour vérifier si le nom existe déjà si c'est le cas la méthode rajoute un nombre à la fin du nom pour le rendre unique.

Annexes

Lien pour les images qui seront déjà dispo sur l'appli:

https://www.pngfind.com/mpng/iRmmbbJ_flappy-bird-atlas-png-atlas-png-flappy-bird/

Icon de l'appli : <https://dlpng.com/png/6781141>