Flappy Bird Processing

version 2.0

DIVERSITY by EPITECH

I. Introduction

Flappy Bird est un jeu d'arcade dans lequel vous devez appuyer sur l'écran pour faire sauter un oiseau. La règle est simple, parcourir la plus grande distance possible sans toucher les tuyaux. Processing lui est un logiciel permettant de coder rapidement des petits projets si cela demande un résultat visuel.

Fin janvier 2014, Flappy Bird est parvenu à être l'application la plus téléchargée de l'Apple Store. Fun Fact, des téléphones possédant le jeu original de flappy bird se sont vendu à plusieurs milliers d'euros !

Logo de Flappy Bird

Attendez un instant ! Un oiseau jaune vous a contacté. Il a besoin d'entraînement pour passer les jeux olympiques de la tuyauterie. Mais avant ça, il va falloir retrouver sa trace.

Logo de Processing

II. Consignes

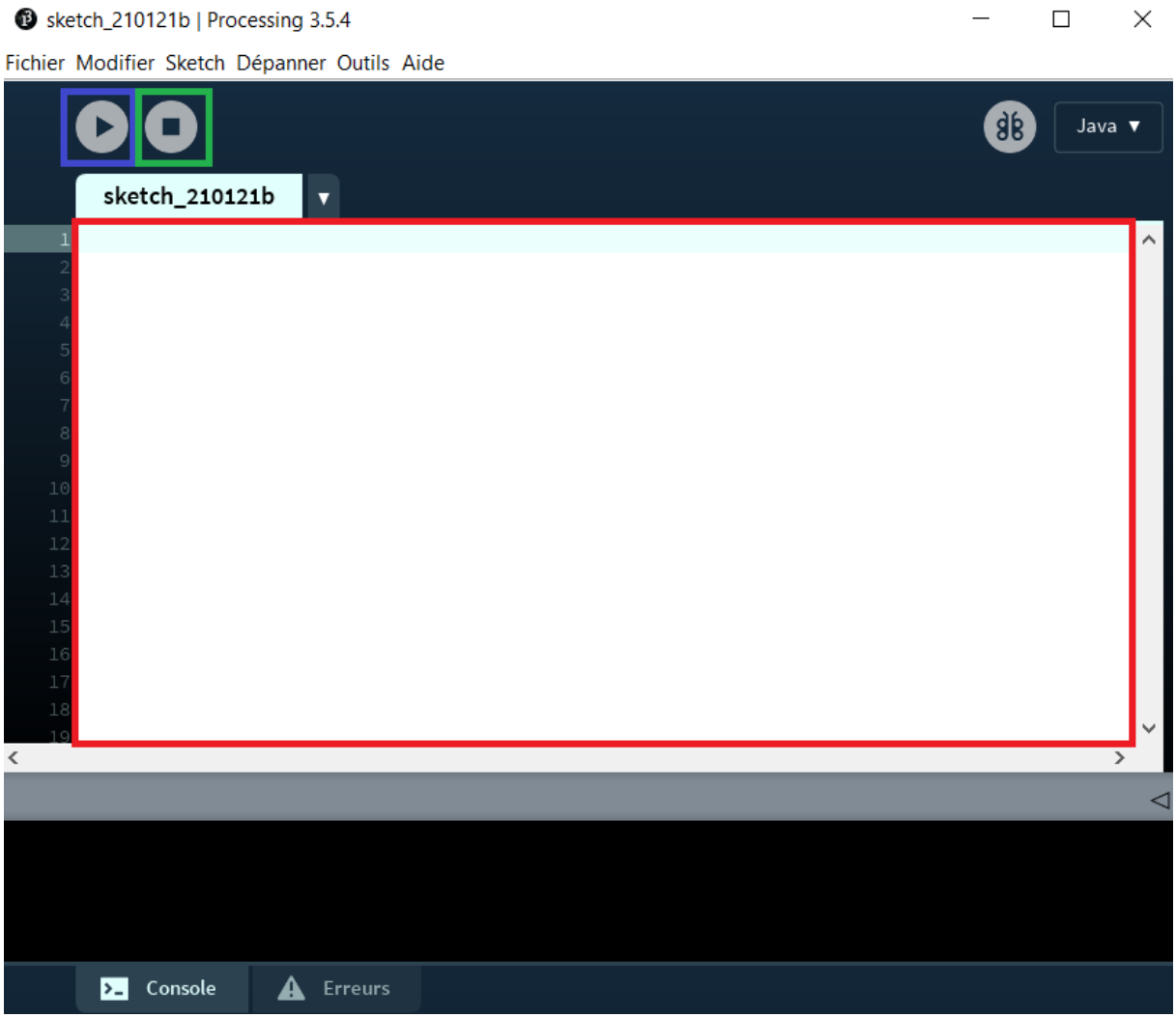
- Ce sujet nécessite l'environnement de développement Processing. Vous pouvez l'installer avec ce lien : Processing - Download.
- Le nom du repository est : cc_FlappyBirdProcessing. Si cela ne vous dit rien, lisez « le coffre à jouet du petit git ».
- Demandez de l'aide aux Cobras en cas de problème d'installation. Si plus rien ne va, recommencez depuis le début en faisant bien attention à toutes les étapes !
- Lors de ce projet, ce lien pourrait vous être grandement utile.
- Si vous bloquez, rappelez-vous que vous êtes accompagné(e) ! Demandez de l'aide à vos camarades ou à un Cobra.
- Internet est un outil formidable pour découvrir le fonctionnement des choses, servez-vous-en régulièrement !

Explication du code :

- Un // est un commentaire pour vous aider à comprendre ce qui est écrit sur la ligne. Le commentaire est ignoré par le programme. N'hésitez donc pas à en mettre pour vous aider !

III. Mettre l'aile à la pâte

Pour pouvoir aider votre oiseau jaune préféré, il vous faut lancer Processing. Une fois fait, vous devriez avoir quelque chose comme l'image ci-dessous.



Processing comment ça marche ?

Voici les différentes parties qui vous intéressent :

- En rouge, la zone utilisée pour écrire votre code
- En bleu, le bouton pour lancer le programme
- En vert, le bouton pour arrêter le programme

Ouvrez « flappy_bird.pde » pour pouvoir commencer à assister votre oiseau.

IV. Le starter pack

a. Setup

L'oiseau jaune vous a demandé de quitter la ville pour le retrouver dans des champs de tuyaux verts. Cela tombe bien, vous pourrez les utiliser pour l'entraîner. Pour partir à sa recherche, suivez ces étapes :

* Lancez le programme une première fois pour voir ce qu'il se trame autour de vous
* Regardez la première fonction, elle se nomme « setup ». Pour plus d'informations sur les fonctions allez lire le « manuel d'aide »
* Lisez les commentaires pour comprendre chaque ligne de cette fonction. Vous pouvez par exemple essayer de modifier les valeurs de cette ligne pour voir ce que ça donne

```
size(288, 512);
```

Si vous ne comprenez pas tout, vous pourrez y revenir plus tard lorsque vous utiliserez certaines variables.

b. Le champ de tuyaux

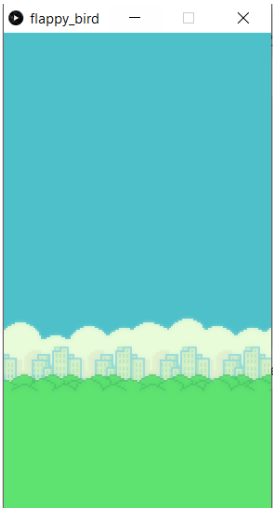
Vous êtes à mi-chemin entre la ville et les champs. Plus vous avancez, plus la ville s'éloigne et plus les champs s'épaississent.

Il ne manque plus grand-chose pour arriver au point du rendez-vous. Pour changer l'ambiance grisâtre de votre fenêtre, vous allez devoir :

* Regardez la fonction « draw ». Supprimer les « // » ligne 3 pour obtenir :

```
    display();
```

* Allez à la fonction « display » qui se trouve à la ligne 92 et supprimez « /* » et « */ » qui sont utilisés pour écrire des commentaires sur plusieurs lignes
* Appelez la fonction « background » comme écrit sur la ligne 94 avec en paramètre, la variable contenant l'image « background-day.png ». Pour information, la variable « background » remplit le fond de la fenêtre avec une image ou une couleur
* Lancez le jeu pour voir le résultat !



Loin de la ville

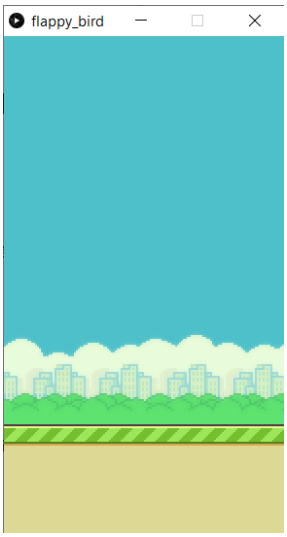
c. Le sol et la parallaxe

Plus vous avancez, plus le sol devient gazonneux et confortable. Vous comprenez pourquoi il veut s'entrainer ici, une chute sur le sol et vous n'avez aucun bobo. Pour simuler le sol, il va vous falloir :

* Ajoutez une ligne dans la fonction « display », à la fin de celle-ci. Appelez la fonction « image » pour pouvoir afficher le sol

```
image(imageApercu, positionX, positionY);
```

* « imageApercu » correspond à l'image à afficher
* « positionX » correspond à la position en x de la fenêtre
* « positionY » correspond à la position en y de la fenêtre
* Affichez le sol en « x » = 0 et « y » = 400. Relancez le jeu et le sol doit normalement être apparu



Le gazon des gazouilleurs

Vous allez maintenant animer le sol pour donner un effet de mouvement.

* Appelez la fonction « changement » au tout début de la fonction « draw »
* Retirez les « // » de la première ligne dans la fonction « changement »
* Enlevez 2 pixels à « positionSol » et réinitialisez-le à 0 lorsqu'elle passe en dessous de -23 pixels
* Trouvez où mettre « positionSol » pour faire bouger l'image, maintenant qu'elle change de valeur. Si tous se passe bien votre sol doit bouger de droite à gauche indéfiniment
* Appelez la fonction « changement » dans la fonction « draw »

d. Les tubes de l'été

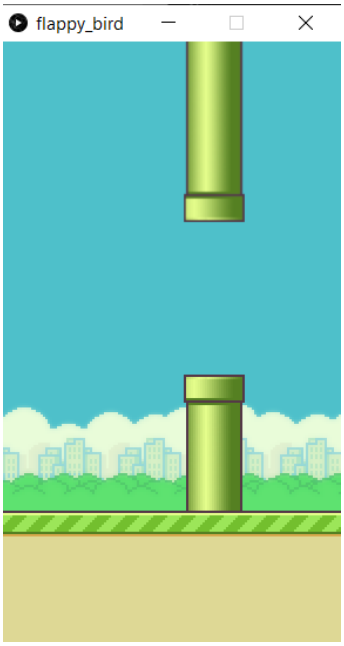
Vous commencez à vous enfoncer dans les champs de tuyaux alors que vous entendez un oiseau gazouiller. Vous approchez au but, il ne manque plus que quelques mètres. Pour simuler les tuyaux, vous allez vous y prendre en plusieurs étapes. Il y a dans un premier temps l'affichage des tuyaux et dans un second temps les collisions que nous verrons plus tard.

Vous l'avez peut-être remarqué plus tôt, mais en réalité vos tubes sont déjà affichés. En effet, ceux-ci sont en dehors de la fenêtre de jeu. Vous allez les faire bouger de la même manière que le sol.

Vous allez devoir faire une condition (cf. « Manuel d'aide ») qui vérifie si le tube a dépassé le côté gauche de la fenêtre. Si c'est le cas, réinitialisez la position en « x » à la valeur initiale et en « y » avec une valeur aléatoire compris entre 200 et 300. Pour générer un nombre aléatoire entre « A » et « B » par exemple, vous pouvez écrire :

```
int(random(A, B));
```

Lancez le jeu, des tuyaux doivent maintenant apparaître à l'infini



Les tuyaux prêts pour l'entraînement

Félicitations !

Vous avez finalement réussi à retrouver l'oiseau qui vous avez demandé de l'aide. Désormais, il va falloir l'entrainer pour les JO !

1ème récompense

V. Flappy Bird

a. Qu'est ce qui est jaune et qui attend

Après avoir discuté avec lui, il vous apprend qu'il se prénomme Jonathan et qu'il voudrait représenter la ligue des oiseaux jaunes lors des JO. Vous le préparez donc pour le début de l'entrainement, prêt pour le décollage ! Pour ce faire, suivez ces étapes :

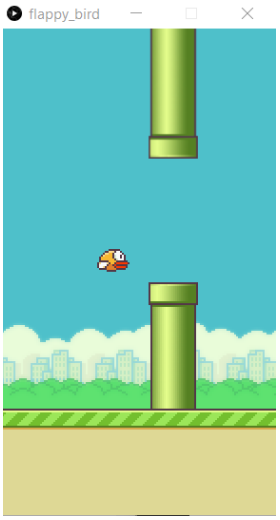
- Rendez-vous dans la fonction « display » et appelez celle-ci pour afficher une image représentant l'oiseau en « x » 100 et « y » 100
- Pour l'animer, il va falloir modifier la fonction « oiseau » et enlevez les commentaires

La variable « hauteur » est la hauteur à laquelle est Jonathan. De la gravité est simulée avec la variable « vitesse », plus elle devient élevée, plus l'oiseau perd de la hauteur.

En dessous, il y a des conditions qui pourraient s'apparenter à : « si … sinon si … sinon ». Pour que votre oiseau s'anime, il faut changer la valeur de la variable « oiseau » à l'aide des 3 images contenues dans le tableau « perso ». Vous devez donc :

- Affectez à « oiseau » la 2ème image de « perso » si l'oiseau à une vitesse élevée
- Attribuez à « oiseau » la 3ème image de « perso » si l'oiseau à une vitesse très basse
- Sinon, donnez « oiseau » la 1ère image de « perso » si l'oiseau à une vitesse élevée
- Remplacez « hauteur » quelque part dans le code. Par la même occasion, appelez la fonction « oiseau » dans la fonction « draw »

Si vous lancez le jeu, l'oiseau tombe hors de la fenêtre



No Pain No Gain

b. Saut’qui peut

Maintenant que Jonathan sait s’élancer dans les airs, il va falloir le faire voler.

* Retirez les commentaires dans la fonction « jump ». La condition déjà écrite vérifie si le joueur appuie sur la touche espace. Si c’est le cas, attribuer la vitesse de Jonathan à -4,5 pour annuler la chute
* Appelez la fonction « jump » dans la fonction « draw »
* Lancez le jeu, l’oiseau peut enfin sauter lorsque vous appuyez sur la touche « espace »

Voici un tarin triste

c. Collision et non illusion

Les tuyaux ne sont pas assez résistants pour faire en sorte que Jonathan apprenne de ses erreurs ! Après tout, se cogner n'est pas la fin du monde, déjà que le sol est mou alors bon.

Pour gérer les collisions entre l'oiseau, les tuyaux ainsi que le sol, rendez-vous dans la fonction « chocOiseau ». Enlevez les commentaires et regardez, il y a écrit « boolean » dans la déclaration de la fonction. Cela signifie qu'il va falloir renvoyer soit « true » soit « false ». La première condition déjà écrite gère la collision entre l'oiseau et les tuyaux, pour finir les collisions, vous allez devoir :

* Complétez la 2ème condition qui est si la hauteur est supérieure ou égale à 372 qui correspond à la position du sol alors il faut retourner « true », sinon il faut retourner « false »
* Appelez « chocOiseau » dans la fonction « draw »

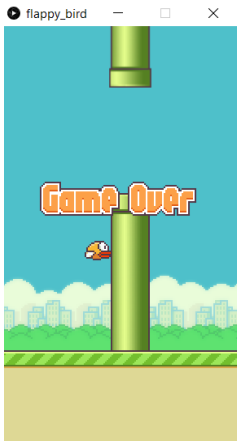
Pour qu'une action se déclenche si la fonction « chocOiseau » retourne « true », il va falloir faire une condition telle que : si… sinon …

Pour ce faire :

* Si la variable « finDuJeu » est égale à « false » alors on fait tout ce qui était là auparavant dans la fonction. Sinon, appelez uniquement la fonction « GameOver ». Pour changer la valeur de « finDuJeu », vous pouvez faire :

```
finDuJeu = chocOiseau();
```

* Allez dans la fonction « GameOver » et appelez la fonction « display ». Cela va permettre d'afficher tous les éléments et de ne pas avoir un fond gris.
* Affichez l'image « gameOver » en « x » = 45 et « y » = 192
* Rajoutez une condition pour savoir quand le joueur appuie sur la touche « R ». Pour ça, reprenez la condition dans la fonction « jump » et adaptez-la. Si la touche « R » est enfoncée, alors appelez la fonction « setup » pour réinitialiser le jeu

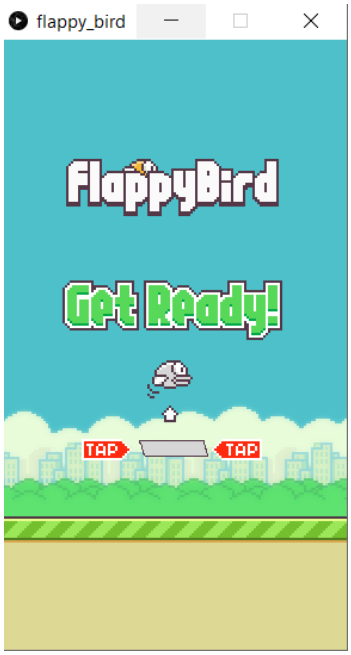


La bataille est perdue mais pas les JO

d. Menu birdger

Jonathan peut enfin s'entrainer à voler jusqu'aux JO. Son terrain d'entrainement est idéal. Vous allez terminer le jeu en faisant un petit menu de démarrage. Suivez ces étapes :

* Ajoutez un « sinon si … » dans la fonction « draw ». Elle vérifiera si c'est la première partie lancée
* Ajoutez par la même occasion une condition dans le premier « if » pour confirmer que « firstTime » est égale à « false » (cf. « Manuel d'aide »)
* Vérifiez si la variable « firstTime » est vraie. Si c'est le cas appelez la fonction « display » puis « startGame »
* Rendez-vous dans cette dernière pour coder le menu. Affichez l'image « startMsg » en « x » = 50 et « y » = 100
* Vérifiez si le joueur appuie sur « espace ». Si c'est le cas, on donne la valeur « false » à la variable « firstTime »



Aperçu du menu

Félicitations !

Vous avez réussi, Jonathan vole. Vous avez mené à bien votre mission. La communauté des oiseaux jaunes vous remercie et ils n'hésiteront pas à vous aider pour votre prochaine aventure !

2ème récompense

VI. Conclusion

Maintenant que Jonathan est prêt pour les JO, vous allez devoir les organiser. Pour ce faire vous pouvez rajouter ces éléments :

* Ajoutez des Flappy coin
* Ajoutez de la musique et des sons
* Ajoutez un saut vers l'avant

Si vous êtes curieux, pensez à poser vos questions aux Cobras. Ils seront ravis de partager leurs connaissances avec vous.

Tous les petits oiseaux jaunes n'ont pas eu cette chance…