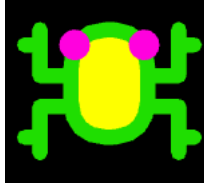


INFO-F-202
Langages de programmation 2
Université libre de Bruxelles

Projet



Justin Dallant John Iacono Alexis Reynouard

1 Sommaire

Le but du projet est de coder un jeu inspiré de Frogger, créé en 1981.



Dans ce jeu, vous contrôlez une grenouille qui commence en bas de l'écran et que vous devez amener sur un nénuphar libre en haut de l'écran. Vous recommencez ensuite en bas avec une nouvelle grenouille, et une fois que les cinq nénuphars sont occupés, le jeu passe au niveau suivant. Notez que dans chaque rangée, les objets (bûches, voitures, camions, etc.) apparaissent selon un motif répétitif. Ils n'apparaissent pas au hasard.

La grenouille a trois vies. Il existe de nombreuses façons de mourir :

- La grenouille meurt si elle est écrasée par une voiture ou un camion.
- La grenouille meurt si elle quitte l'écran.

- La grenouille meurt si elle tombe dans l'eau.
- La grenouille meurt si elle est sur une tortue qui plonge sous l'eau.
- La grenouille meurt si elle manque le nénuphar.
- La grenouille meurt si le nénuphar est déjà occupé par une grenouille.

Il existe d'autres façons pour la grenouille de mourir dans le jeu, impliquant des serpents, des crocodiles et des loutres. Mais pour la simplicité, nous ne discuterons pas de ces animaux. Le vrai jeu comporte des bonus telles que des grenouilles violettes et des mouches, que nous ignorerons également.

Pour référence, voyez :

- [Page Wikipedia de Frogger](#),
- [Description des éléments du jeu](#),
- [Jouer à Frogger en ligne](#) et
- [Jouer à Frogger en ligne](#).

Vous pouvez réaliser le projet individuellement ou par groupe de deux. Si vous le complétez en groupe, les deux étudiants recevront des notes identiques (sauf si un étudiant ne se présente pas à la présentation orale).

Vous devez remettre :

- un zip avec votre code, d'autres fichiers nécessaires (vous pouvez utiliser des images ou des fichiers texte) et un Makefile,
- un document au format pdf expliquant votre démarche (voir ci-dessous).

Ces documents sont dûs pour le 14 janvier à 23h59 pour la première session. Les dates de la session d'été seront annoncées ultérieurement.

La semaine du 15 janvier, il y aura entretien oral où vous ferez une démonstration de votre programme et répondrez aux questions.

[Vous devez cliquer ici pour choisir une heure pour présenter votre projet.](#)

Pour la deuxième session, le projet sera identique et sera rendu en août. La date exacte sera annoncée une fois que le calendrier des examens de la deuxième session sera fixé. Vous pouvez passer l'examen écrit en première session et faire le projet en deuxième session (ou inversement) si vous le souhaitez.

2 Notation

La notation prendra notamment en compte les éléments suivants :

- tâches terminées,
- rapport écrit,
- qualité du code,
- présentation orale et questions,
- retard.

Votre note maximale sera de 10 si vous ne terminez que les tâches de bases et passera à 20 si vous terminez également 10 tâches additionnelles. À l'exception des tâches de bases qui doivent être réalisées en premier, vous avez le droit d'effectuer une tâche sans avoir effectué les tâches précédentes. La qualité de votre code, votre rapport écrit, votre présentation orale et vos réponses aux questions à l'oral détermineront votre note par rapport au maximum déterminé par le nombre de tâches que vous effectuez.

3 Tâches

Les tâches de base consistent en la réalisation d'un niveau fonctionnel de Frogger. Plus précisément, il faut implémenter les points ci-dessous.

Une grenouille que vous pouvez déplacer dans quatre directions avec le clavier. Pour cette tâche, la grenouille peut simplement être un carré vert. Si vous choisissez d'utiliser les touches ZQSD pour les déplacements, veuillez ajouter W (pour haut) et A (pour droite) (et inversement si vous utilisez WASD).

Treize rangées. Dans le jeu réel, les rangées 1 et 7 sont des trottoirs, les rangées 2 à 6 sont des rangées de voitures ou de camions (routes), les rangées 8 à 12 sont des rangées d'eau avec des tortues ou des bûches et la rangée 13 est la rangée de nénuphars. Pour la tâche de base, vous pouvez remplacer la rangée de nénuphars par un trottoir et les rangées d'eau par des routes.

Un message d'échec qui apparaît en même temps que la mort de la grenouille si celle-ci entre en contact avec une voiture ou quitte l'écran.

Un message de victoire qui apparaît quand la grenouille arrive dans la rangée 13.

Les tâches additionnelles sont à réaliser une fois les tâches de base terminées.

1. **Rangées d'eau.** Créez des rangées d'eau avec des troncs de différentes tailles et des tortues. (Les tortues ne plongent pas pour cette tâche). La grenouille meurt si elle saute dans l'eau (ou sort de l'écran) et se déplace sur les bûches et les tortues, comme dans le jeu réel.
2. **Nénuphars.** Créez les nénuphars de la rangée 13 comme dans le jeu réel. Si vous déplacez la grenouille vers un nénuphar vacant, elle y reste et une nouvelle grenouille que vous contrôlez apparaît en bas. Lorsque tous les nénuphars sont occupés par des grenouilles, affichez le message « Victoire! ». Si la grenouille entre dans la rangée 13 ailleurs, elle meurt.
3. **Vies.** Comme dans le jeu, faites en sorte que vous ayez trois vies, qui sont affichées en bas. Lorsque la grenouille meurt, une vie est perdue et le jeu continue avec une nouvelle grenouille qui apparaît en bas. Lorsque toutes les vies sont perdues, le jeu se termine. Le message d'échec n'apparaît qu'à ce moment. Notez que placer une grenouille sur un nénuphar n'ajoute pas de vie.
4. **Tortues plongeantes.** Faites plonger et refaire surface des tortues, comme dans le jeu. Si la grenouille est sur une tortue qui plonge, elle meurt.
5. **Directions de la grenouille.** Dans le jeu, la grenouille est orientée dans la direction du mouvement précédent. Mettez en œuvre ceci. Notez que vous devrez dessiner la grenouille comme autre chose qu'un carré afin que vous puissiez voir son orientation.
6. **Score.** Afficher un score. Vous pouvez utiliser n'importe quelle méthode raisonnable pour calculer le score, cela ne doit pas nécessairement être comme dans le jeu réel.
7. **Meilleur score.** Affichez quelque part le score le plus haut obtenu par le joueur pour chaque niveau. Ce score doit être enregistré et persister à la fermeture du programme. Il faut aussi pouvoir le réinitialiser.
8. **Écran d'accueil.** Lorsque vous démarrez le jeu, un écran d'accueil apparaît pendant une seconde avant que l'écran principal du jeu ne s'affiche. L'écran d'accueil doit inclure votre nom.
9. **Niveaux et sélection de niveau.** Implémentez différents niveaux que vous enregistrez dans un ou plusieurs fichiers. Vous devez y retenir le type de chacune des rangées 2 à 12, la séquence de voitures et camions / troncs et tortues qui s'y déplace de façon répétée,

et la vitesse de déplacement sur la rangée. Il vous faut ensuite un écran de sélection de niveau et un moyen d'y accéder depuis le menu principal. L'écran de sélection de niveau peut être simple.

10. **Éditeur de niveau.** Ajoutez la possibilité de pouvoir modifier un niveau de manière interactive au lieu d'y jouer. Vous devriez pouvoir enregistrer les modifications et ajouter de nouveaux niveaux.

4 Qualité du code

Votre score de qualité de code sera basé, entre autres, sur la façon dont vous codez en utilisant les principes de la programmation orientée objet. Voici des points que l'on attend dans votre projet :

- Décompositions en classes et en fichiers.
- Aucune variable globale.
- Pas de logique répétée.
- Commentaires et documentation du code.
- Code des méthodes assez court.
- Utilisation correcte des constructeurs et des destructeurs.
- Pas d'utilisation des classes compliquées de fltk. (Utilisez simplement les fonctions de dessin, placées dans des classes comme nous l'avons fait dans les laboratoires.)
- Utilisation du modèle de conception MVC (couvert dans l'avant-dernière leçon).

Les classes de dessin de base qui appellent les fonctions fltk doivent être séparées de votre code spécifique à Frogger.

Votre code doit être aussi modulaire que possible. Par exemple, si l'on vous demande d'intervir la fonctionnalité de deux lignes, vous devriez pouvoir le faire très facilement.

5 Questions orales

Vous aurez une présentation orale pendant laquelle il vous sera demandé de faire une démonstration de votre programme et de votre code. Durant cette présentation nous vous poserons quelques questions à ce sujet. Nous vous demanderons peut-être d'apporter une modification mineure au code pour voir si vous comprenez votre propre code. Si vous avez réalisé le projet en équipe de deux, les deux membres de l'équipe doivent se présenter ensemble.

6 Rapport écrit

Vous devez inclure un rapport écrit au format PDF, d'au maximum 10 pages. Il n'y a pas de minimum de longueur, mais le rapport doit au minimum comprendre les points suivants

- Tâches. Indiquez les tâches que vous avez accomplies.
- Classes. Pour chaque classe, fournissez l'interface (pas le corps des méthodes) et décrivez brièvement le rôle de chaque classe et comment elle se rapporte aux autres classes.
- Logique du jeu. Supposons que je démarre le jeu, sélectionne le premier niveau (si la tâche 9 est terminée), attend 10 secondes, puis fais un coup. Décrivez en détail ce qui se passe dans votre code.

- Modèle-Vue-Contrôleur. Avez-vous utilisé ce modèle de conception ? Si c'est le cas, expliquez comment vos classes correspondent à ce modèle de conception. Sinon, expliquez vos choix.

Prenez garde au fait vous que ce rapport fera partie des critères qui influencerons votre note (bien que ce ne soit pas le critère principal), et pas un document optionnel à bâcler. En particulier, essayez de rendre celui-ci propre et clair pour le lecteur.

7 Retard

Vos points seront réduit de 1% pour chaque heure de retard.

8 Ce sur quoi vous n'êtes PAS noté

Vous êtes noté sur votre capacité à créer un programme volumineux, complexe et orienté objet. Vous n'êtes pas noté sur votre capacité à avoir un beau jeu. Normalement, un graphiste créerait de jolis éléments visuels, et vous êtes des informaticiens, pas des graphistes. L'aspect visuel devrait être assez bon pour que les différents éléments soient reconnaissables et que les animations soient raisonnables, mais aller au-delà n'aidera pas votre note. Il est tout à fait acceptable que les voitures et les camions soient des rectangles, les tortues soient des cercles, etc.

Pour certains d'entre vous, l'écran des applications créées avec ftk clignote lors de l'exécution de programme. Cela se produit le plus souvent si vous l'exécutez dans une machine virtuelle. Cela n'affectera pas votre note.

9 Plagiat

Nous devons aborder le problème du plagiat dans le code. Ce n'est pas facile, car il est tout à fait normal de couper et coller de petits morceaux de code. Cependant, il n'est pas acceptable que vous copiez une un gros morceau de code spécifique à ce jeu d'autres étudiants ou du Web.

La règle suivante s'applique : si vous avez plus d'une seule ligne de code que vous avez obtenue ailleurs, vous devez inclure un commentaire avec l'endroit où vous l'avez obtenue (comme l'URL). Notons que vous êtes libre de partager toutes les classes que vous créez qui encapsulent un simple dessin ftk avec d'autres étudiants, tant que vous les publiez publiquement sur le canal de projet des équipes.

Toute violation de cette politique, même minime, peut entraîner un 0 sur le projet.

Nous ne ferons pas de distinction entre ceux qui ont fourni du code et ceux qui ont copié du code ; si deux groupes partagent le code sans attribution, les deux obtiendront un 0. Vous pouvez éviter cela en insérant simplement un commentaire !