



**TP2 – Zoo Tycoon**  
**- Version 2 joueurs -**

## **Objectifs du travail**

- Créer un contrôle personnalisé
- Utiliser un Thread
- Utiliser des graphiques
- Créer une application ergonomique et conviviale
- Créer une application utilisant le composant créé

## **Dates de remise**

**Le jeudi 9 mai 2019 : remise finale**

- Une auto-évaluation de votre programme (ce qui fonctionne très bien, moins bien, pas du tout et pourquoi) ;
- Les jeux d'essais utilisés pour tester : liste des jeux d'essais et explication de leurs choix ;
- Le projet C# dans sur le compte Lea **d'un seul coéquipier.**

Fatigués du tourbillon incessant de la ville?

La pollution, le bruit, l'odeur d'égouts, de vieilles chaussettes humides et de vidanges qui est toujours présente, le jour comme la nuit?

Sans parler des stationnements par-dessus les bancs de neiges et des trottoirs sur lesquels on peut jouer au curling. Et bien sûr, c'est pas mieux l'été : la chaleur accumulée dans l'humidité relative au-dessus des 100%, où aucune brise de fraîcheur n'a mis les pieds depuis longtemps.

Non, vraiment, à part votre cours de programmation à BdeB qui est comme un rayon de soleil plombant un coffre au trésor rempli de cartes cadeaux Amazon de 1\$ chacune, le tout accompagné d'un orchestre symphonique qui joue le dernier tube de Justin Bieber, tout le reste à Montréal vous répugne.

Oui, même les petites patates de la cafétéria ne semblent plus aussi délicieuses qu'avant.

Vous rêvez d'une vie plus simple, plus posée, là où le chant majestueux d'un petit colibri accompagne le café bien chaud du matin, là où les enfants jouent sur leur iPhone 4s plutôt que sur leur iPhone 7, oui vraiment, un monde plus simple, plus pittoresque, plus zen.

L'appel de la nature quoi!

Déprimé et affaibli à la fenêtre de l'autobus de la STL (en retard il va sans dire), un coup de génie surgit :

Vous allez acheter un zoo en campagne!

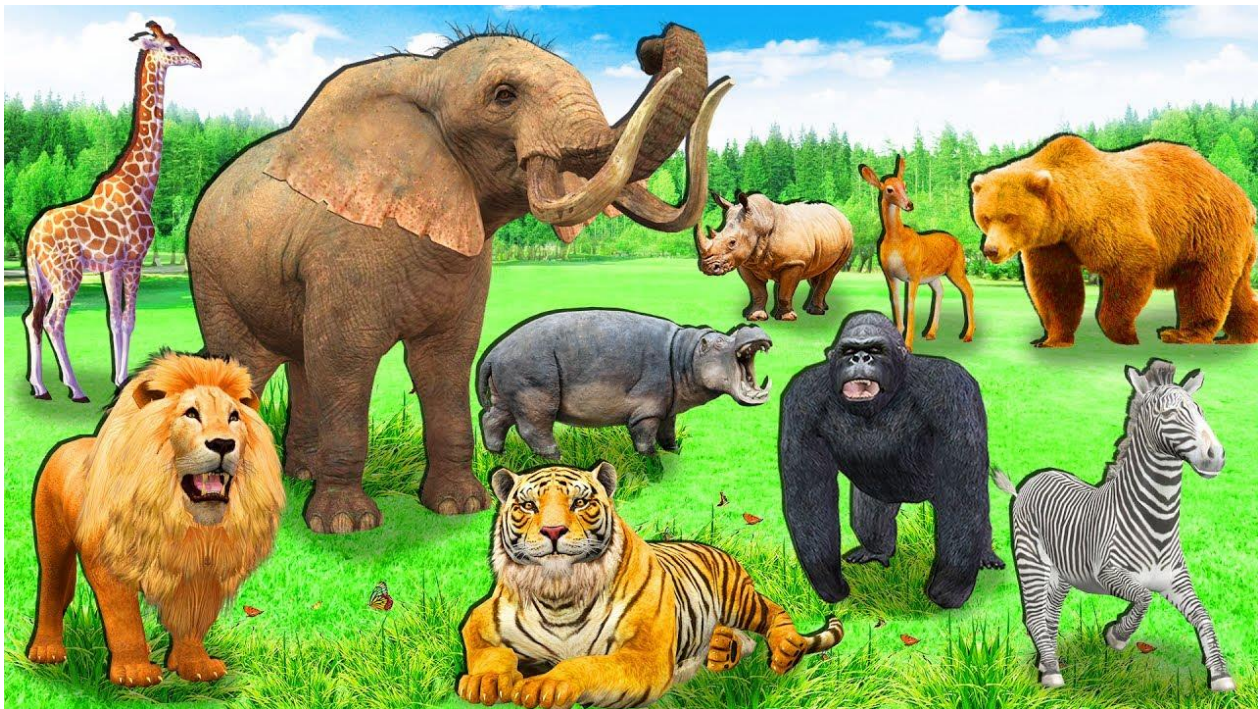
Le déclic fût immédiat : le torse bombé et le regard perçant, vous tirez sur la corde de l'autobus d'une main ferme, sortez au premier arrêt et vous partez courir d'un pas décidé dans le nord.



Nids de poules, ponts croulants, sloche, grêle, grésil, pluie verglaçante, rats enragés, scouts qui vendent des biscuits, non, rien n'arrêtera votre course, pas même un café gratuit chez McDo.

Une seule pensée en tête, un retour aux sources, la vie, la vraie de vraie vie, là où les vaches dansent, les coqs dansent, les cochons dansent, tout le monde danse (bon, peut-être que vous commencez à halluciner de déshydratation avancée et de faim). Votre but ultime :

- 1- S'acheter un zoo loin loin de la ville.
- 2- Le remplir d'animaux.
- 3- ???
- 4- Profit !



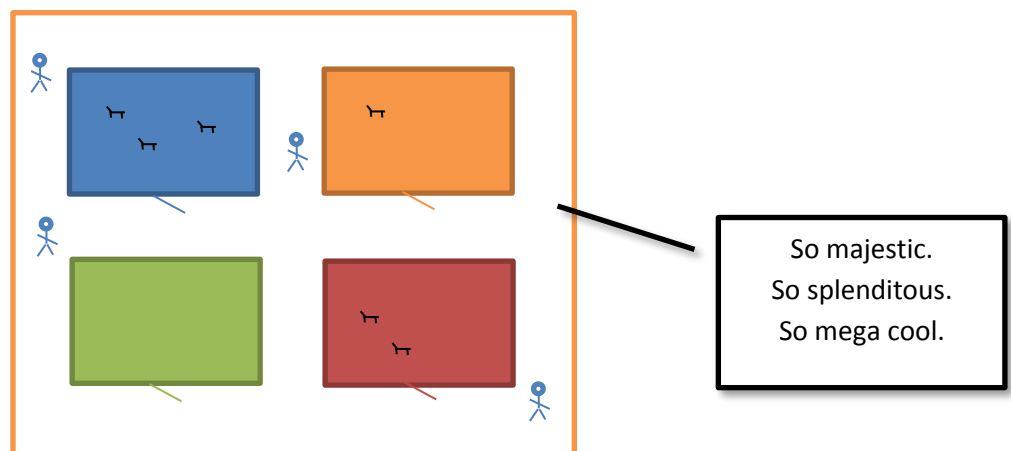
## Travail à accomplir

**Notez que lorsqu'il y a ambigüité dans l'énoncé, on vous demande de vous baser sur les conventions de ce type de jeux.**

### Idée générale du jeu

Lors d'une nouvelle partie, le héros est devant la porte de sa maison, le cœur léger et l'âme joyeusement souriante. Il n'a qu'une idée en tête : construire le zoo le plus rentable qui soit, tout en respectant les besoins des nombreux animaux de son zoo.

Vous devrez permettre à votre joueur de remplir son zoo de différents animaux, en plus de lui permettre d'enjoliver celui-ci. Lorsque vous êtes prêts, vous ouvrez votre zoo et regardez les nombreux visiteurs affluer!



## Spécifications

### Interface utilisateur :

- Le héros se déplace à l'aide du clavier sur la mappemonde : wasd et/ou flèches (un seul choix requis).
- L'interaction avec les tuiles se fait à l'aide d'un cliquer de souris, en fonction de l'action choisie.
- Voir en tout temps :
  - o Jour actuel (ex : 24 mai 2019)
  - o \$ du joueur
  - o Nombre d'animaux
  - o Nombre de déchets

### Héros :

- Sur la mappemonde, le héros est toujours animé (il marche).
- Il se déplace case par case, sans transition.
- Il regarde toujours dans la direction qu'il se déplace.
- Il se déplace à l'aide du clavier sur les tuiles permises, non occupées (donc si un animal ou un visiteur se trouve sur une tuile, il ne peut pas s'y déplacer tout de suite).
- Débute avec 100\$ en poche.

**Enclos :**

- 4 enclos disponibles dans le zoo.
- Les enclos sont fixes, ils ne peuvent pas être agrandis ou déplacés.
- Ils n'acceptent qu'un type d'animal : tant qu'il y a un animal présent dans l'enclos il n'est pas possible de rajouter un animal différent.

**Animaux :**

- Sur la mappemonde, les animaux sont toujours animés.
- Ils se déplacent case par case, sans transition. Ils regardent toujours dans une même direction, de votre choix.
- Ils ne peuvent être ajoutés qu'à l'intérieur d'enclos existants.
- Un seul type d'animal par enclos est permis.
- Ils se déplacent aléatoirement sur les cases permises, et demeurent à l'intérieur de l'enclos (ils ne peuvent également pas embarquer sur la même tuile que le héros ou autres animaux)
- Lorsque le joueur clique sur un animal et qu'il est adjacent, cet animal est automatiquement nourri et fait un bruit adéquat (ex : un lion rugit, un mouton bêle, une fourmi cröonde<sup>1</sup>).
- Ce sont des animaux magiques, ils ne peuvent pas mourir ni être malade.

**Visiteurs (avoir 4 dessins différents, 2 hommes et 2 femmes)**

- En fonction du nombre d'animaux, le nombre de visiteurs doit augmenter -> 1 visiteur pour chaque animal.
- Les visiteurs arrivent par la porte d'entrée du Zoo.
- Un visiteur qui est dans votre Zoo depuis moins d'une minute ne peut en aucun cas quitter votre zoo. Un visiteur qui « marche » sur la porte de sortie et qui a passé plus d'une minute dans votre Zoo quittera automatiquement. Il sera remplacé quelques secondes plus tard par un nouveau visiteur.
- Un visiteur peut aléatoirement laisser tomber un déchet par terre.
- Un visiteur se déplace de façon totalement aléatoire, et ne peut pas entrer à l'intérieur d'un enclos. Il ne peut pas se déplacer par-dessus un autre objet.

**Rémunération**

- Chaque visiteur qui rentre paie un prix fixe d'entrée (2\$ par animal présent au moment de son arrivée)
- Après chaque minute (temps réel) passée dans votre parc, chaque visiteur repaie 1\$ pour chaque animal présent à ce moment précis, moins 0.10\$ pas déchets présents actuellement dans le parc.

**Concierge**

- Il est possible d'engager un (ou plusieurs) concierge. Ils se déplacent aléatoirement aux endroits permis aux visiteurs, et ramassent automatiquement les déchets qui se trouveraient sur une case.

---

<sup>1</sup> Cette minute humoristique est une gracieuseté de Mathieu, dont l'humour exceptionnelle n'est plus à prouver, et ce depuis vraiment très très longtemps



- Frais de 2\$ par minute (temps réel) pour chaque concierge.

### Carte de jeu - Mappemonde (User control):

- La partie « jouable » de la mappemonde reprend un peu le travail accompli du Labo #5, en utilisant le même fichier source (.cs) adapté à la nouvelle feuille de *sprites* fournie.
- La carte complète du jeu est visible simultanément depuis l'écran.
- Le monde devra permettre les mêmes spécifications que le Labo 5 (ex : largeur/hauteur).
  - o La « scène de jeu » doit être un contrôle utilisateur qui affiche chacune des tuiles du jeu selon la position du héros.
  - o ATTENTION : Il faut dessiner chacune des tuiles à même le `Paint()` du contrôle. **On ne doit pas avoir 200+ contrôles personnalisés sur la scène de jeu.**
- Certaines tuiles ne peuvent être traversées par le héros (buissons, roches, clôtures etc.). En cas de doute, vérifiez avec moi.

### Interactions du joueur :

- Sur la mappemonde :
  - o Cliquez de souris sur une case adjacente permet d'effectuer l'action qui était sélectionnée par le joueur. L'adjacence fonctionne sur les 8 cases entourant le joueur ainsi :

	X	X	X				
	X	☺	X				
	X	X	X				

- o Cliques disponibles (clique **gauche**) si adjacent au héros :
  - Placer un nouvel animal (au choix du joueur) directement dans un enclos (\$\$\$)
    - Toujours un animal adulte
    - Genre aléatoire
  - Nourrir un animal (+ cri de l'animal) (1\$ par clique, peu importe s'il avait faim ou non).
  - Ramasser un déchet sur le sol
  - Engager un concierge
- o Cliques disponibles (clique **droit**) peu importe l'emplacement du joueur :
  - Clique droit sur un animal (**User control**)
    - Afficher les informations de tous les animaux présents dans le parc, avec celui cliqué clairement identifié et affiché :
      - o Image
      - o Type
      - o Genre (M/F)
      - o Bébé/Adulte
      - o Nourri/a faim. Depuis combien de temps il a faim si c'est le cas.

- Si l'animal attend un bébé ou non (doit être une femelle adulte, en présence d'un mâle adulte dans le même enclos)
- Cliquez droit sur un visiteur
  - Afficher les informations du visiteur cliqué :
    - Image
    - Nom généré aléatoirement depuis une liste de prénoms et de noms de famille de votre choix<sup>2</sup>
    - Sexe (M/F)
    - Depuis combien de jours il est arrivé dans le zoo

### Animaux (Permettre minimalement 3 types d'animaux différents dans votre zoo, au choix)

Type d'animal	Prix d'achat d'un adulte (Genre aléatoire)	Période de gestation	Période avant de devenir adulte	Doit être nourri avant ce temps pour éviter une contravention
Lion	35 \$	110 jours	110 jours	1 mois
Mouton	20 \$	150 jours	150 jours	1 mois
Grizzly	30 \$	220 jours	220 jours	1 mois
Rhinocéros	40 \$	480 jours	480 jours	2 mois
Licorne	50 \$	360 jours	360 jours	2 mois
Buffle	40 \$	340 jours	340 jours	2 mois

### Vitesse de jeu

- Une année complète de jeu (365 jours) doit durer environ 5 minutes.
- Un jour → 5 minutes / 365 jour
- Afficher en tout temps le jour actuel (1 janvier 2019, 2 janvier 2019, 3 janvier 2019...)

### Fin de la partie

- Aucune fin de partie à coder
- Indiquer le profit total du joueur en quittant le jeu

## Spécifications techniques

**Vous n'avez pas à permettre la sauvegarde du jeu.**

### Dossiers

Vous devez regrouper en package les classes qui partagent un même domaine du jeu (positionner les classes dans des dossiers dans le projet).

<sup>2</sup> Pas de blague de mauvais goût ici. Vous êtes avertis.

## Code

- Utilisez le fichier source fourni au Labo #5, mais adapté à la nouvelle feuille de *sprites* fournie avec le TP.
- Codez des classes (avec héritage) pour vos joueurs, visiteurs, animaux... et utilisez-les.
- Utilisez des listes (et le polymorphisme!) pour gérer vos objets intelligemment dans le jeu.
- Évitez la redondance de code.

## Utilisation du serveur SVN

Un serveur de gestion de source (Subversion) sera disponible à l'adresse [https://svn.minilab.bdeb.gc.ca/svn/nom\\_du\\_depot](https://svn.minilab.bdeb.gc.ca/svn/nom_du_depot), et vous devrez y mettre vos sources pour travailler sur votre projet en équipe. Les détails sur l'utilisation de cet outil vous seront communiqués en classe.

## Critères d'évaluation

Votre programme doit respecter les normes de programmation.  
Le travail est à faire seul ou à deux, à moins d'avis contraire.  
Vous devez utiliser Visual Studio 2017

Critère	Nombre de points accordés
<b>Analyse</b>	4
<b>Jeux d'essais</b>	4
<b>Auto-évaluation</b>	
<b>Interface</b> <ul style="list-style-type: none"> <li>esthétique</li> <li>adaptation à la clientèle</li> <li>facilité d'utilisation</li> </ul>	5
<b>Composants utilisateurs créés</b> <ul style="list-style-type: none"> <li>fonctionnement (propriétés, méthodes, événements)</li> <li>respect des consignes</li> </ul>	5
<b>Fonctionnement du jeu</b> <ul style="list-style-type: none"> <li>respect des consignes</li> </ul>	24
<b>Code et programmation</b> <ul style="list-style-type: none"> <li>Utilisation et adaptation du code existant</li> <li>intégration dans le code existant</li> <li>Programmation selon l'approche orientée-objet</li> <li>Respect des normes</li> <li>Commentaires</li> </ul>	8
<b>Total</b>	<b>45 *</b>