LOG8715

TP2: Optimisation

Version 3.0

Responsable: Olivier Gendreau

Chargé de laboratoire: Louis-Philippe Lafontaine-Bédard

Auteurs: Samuel Bellomo, Martin Paradis et Louis-Philippe Lafontaine-Bédard

Introduction

L’objectif de ce laboratoire est de faire la conception et d’implémenter une structure de donnée optimisée pour une simulation déjà fournie.
Pour ce faire, vous devrez utiliser les techniques vues en cours afin d’être capable d’augmenter le nombre d’entités simulées à son maximum.

Le travail sera en équipe de deux.
L’implémentation sera faite avec le moteur Unity avec le projet de base fourni sur Moodle.

Travail à accomplir

Préparation

Installation et ouverture du projet

Si ce n'est pas déjà fait, installez **Unity 2019.4.16f1** et ouvrez le projet pour la plateforme Windows.
Veuillez utiliser la version d'archive disponible au site:

https://unity3d.com/get-unity/download/archive

Assurez-vous d'utiliser la version spécifiée d'Unity, étant donné qu'une version différente demanderait une migration du projet. Des points seront enlevés si la version est différente.

Ouvrez la scène `Scenes/Main.unity`

Fonctionnalités déjà existantes

La simulation contient déjà une implémentation de base.
Description de la simulation déjà existante:
La simulation contient des cercles qui évoluent à l'écran.
Il y a deux types de cercles:

- Des cercles dynamiques qui ont une vitesse et se déplacent à l'écran.
- Des cercles statiques qui ne bougent pas.
    - Le ⅕ des cercles est statique.

- À chaque détection de collision entre deux cercles, la taille des entités dynamiques est réduite de moitié. Les cercles statiques ne changent pas.
    - Il n'y a pas de restitution au contact d'entités. Lors d'un contact, la vitesse des entités restera inchangée.
    - Une fois en dessous de la taille minimum indiquée dans la config, ces entités n'ont plus de collisions et se passent à travers l'une l'autre.
    - Lorsque l'entité entre en collision avec un bord de l'écran, en plus de rebondir son état se remet à celui d'origine. Ceci inclut sa taille et sa capacité à entrer en collision avec les autres cercles.

Couleurs:
Les entités ont une couleur qui change selon leur type.

- Les entités statiques sont rouges.
- Les entités dynamiques avec collisions sont bleues.
- Les entités dynamiques sans collisions sont vertes.

Travail à faire: Optimisations

Ce sera à vous de modifier le ComponentManager. Vous devrez déterminer et implémenter la ou les structures de données les plus appropriées pour ce TP.
Les systèmes déjà fournis devraient continuer à exécuter de la même façon.

Nombre de cercles à simuler: 950
La simulation doit rouler à 30 FPS avec ce nombre de cercles.

La performance sera évaluée sur les postes du L4818. Vous pouvez vous y connecter à distance pour effectuer vos tests finaux.

Restrictions:
Seul le code de ComponentsManager devrait être modifié. Vous êtes libre d'ajouter les fichiers et classes que vous voulez. Tous les autres fichiers seront remplacés à la correction, y compris les fichiers de settings d'Unity. Vous devrez vous-même faire votre implémentation, vous ne pourrez donc pas utiliser les implémentations déjà existantes comme DOTS.
Le but du TP est de vous faire expérimenter avec les structures vues en cours, votre implémentation devrait donc rester dans l'esprit du TP.

Le profileur d'Unity peut être accédé à partir du menu *Window/Analysis/Profiler*/. Le profileur en mode *deep profiling* peut être utile pour trouver les méthodes qui prennent le plus de ressources, mais il ne sera pas assez profond pour vous donner la solution. C'est à vous de comprendre qu'est-ce qui se passe bas niveau et d'optimiser en conséquence.

Il vous est conseillé de prendre du temps au début de votre implémentation afin de planifier vos optimisations et de ne pas perdre du temps à implémenter une solution qui ne vous rapportera pas beaucoup de gains de performance.

Mode *headless*

Afin de ne pas être limité par la carte graphique du poste de test, un mode sans rendu graphique est disponible. À ce moment, seule la simulation sera prise en compte et le *bottleneck* devrait être la mémoire et le CPU.
Un *build* du jeu sera utilisé pour l'évaluation. L'évaluation sera faite en utilisant la ligne de commande:
`LOG8715-TP2.exe -batchmode -nographics -logFile TP2.log`
Qui exécutera votre jeu en background.
Pour quitter le jeu, exécuter `taskkill /IM LOG8715-TP2.exe` sur la ligne de commande
Pour voir vos statistiques, vous pouvez ouvrir le fichier `TP2.log` créé par l'exécution de votre exécutable.

Code déjà fourni

Vous pouvez trouver la statistique de *FPS* utilisé pour évaluer le TP dans la console d'Unity et dans le fichier de log TP2.log (lorsque vous exécutez le *build*).

Quelques options s'ajoutent dans ECSManager pour vous aider à debugger. "Debug print" fera un log à chaque frame sur le contenu du component manager et "should display entity IDs" affichera à l'écran les IDs des cercles qui évoluent à l'écran.

Rapport

Format PDF, interligne 1.5, 12 pts

Max 300 mots

Vous devrez rendre une courte discussion sur les optimisations et la maintenabilité de votre solution pour ce TP. À noter qu'une discussion implique de justifier vos points. Il vous est donc demandé de décrire vos optimisations, de les justifier et de discuter de la maintenabilité de votre solution.
Une capture d'écran de vos résultats de profilage doit être incluse dans le rapport.
Un diagramme de votre solution doit être inclus dans le rapport.

Évaluation (/15)

Un *build* en mode headless sur la ligne de commande sera utilisé pour l'évaluation.
Votre simulation devra être capable de gérer le nombre requis d'entités tout en étant capable de maintenir 30+ FPS.
L'évaluation de la performance finale de votre solution sera évaluée en l'exécutant sur les postes du laboratoire en mode *headless*, c'est-à-dire sans affichage graphique. Les postes devraient avoir la bonne version de Unity, vous pouvez donc déjà tester votre performance sur ceux-ci.

Implémentation	
Rapport	3
Implémentation	4
Performance	8
Manque de clarté/maintenabilité	-3
Projet ne compile pas	-10
Mauvais format de remise et langue	-3
Retards	Perte de points exponentielle. jour 1: -2 jour 2: -4 jour 3: -8 **0 après 3 jours de retard**

Remise

Votre dossier de remise devrait contenir votre projet Unity. Le nom du fichier zip devrait contenir les matricules des coéquipiers.

LOG8715_TP2_matricule1_matricule2.zip

|___ rapport.pdf

|___ ProjetUnity

|____ Assets/…

|____ ProjectSettings/…

|____ ….

Ne pas inclure le dossier Library et autre fichiers générés. Pour une liste complète de fichiers à éviter, voir le .gitignore à

https://github.com/github/gitignore/blob/master/Unity.gitignore

Si votre projet utilise git, vous pouvez utiliser la commande git

```
git archive -o TP2.zip HEAD
```

Pour créer votre archive zip.

Le projet ne devrait pas être très lourd, veuillez remettre le zip sur Moodle avant la **date indiquée sur Moodle.**