Meta-Simulateur de Jeu

Livraison Janvier 2015

Soutenance le 2015

- Ce qui n'est pas spécifié est à votre imagination…
- Surprenez vous, surprenez moi !
- Vous pouvez me contacter : metavlc@gmail.com
- Ce document complète ce qui a été présenté en cours

Objectif : Développer un logiciel permettant de créer plusieurs types de Simulation : (exemples…), au moins 3 types de scénarios.

- Une simulation de déplacement (environnement sans contrainte, tableau)
- Une simulation de « labyrinthe » (environnement contraint (mur, porte), tableau)
- Une simulation de trafic
- Un jeu type « tower defence »
- Une fourmilière
- Une rencontre sportive
- Autre simulation (« à vous d'inventer »)

Pour chaque Simulation, on définit :

- **Un Gestionnaire de Jeu**

  Fonctionnalités attendues :
  - Créer un nouveau Jeu
  - Charger les fichiers de données (Format XML)
  - Sauvegarder
- **Un Moteur de simulation**
  - Tour de jeu :
    1. Propagation des informations générales
    2. Propagation des ordres (ne plus bouger, sans ordre….) – propagation hiérarchique
    3. Pour chaque **personnage** : **AnalyseSituation()**
    4. Pour chaque **personnage** : **Execution()**
    5. Pour chaque **Conflit : Médiation des conflits()**
    6. Pour chaque objet : **MiseAJour()**
    7. **RécupérerInformations()** + **CalculStatistique()**
    8. Affichage
  - Options paramétrables

- **Un environnement de Zones** (Zones + Accès entre zone)
    - Une zone peut contenir plusieurs personnages
    - Une zone peut contenir plusieurs objets
- **Une gestion des personnages** et **des objets**
    - Des personnages
        - Point de vie (décrémenté à chaque tour)
        - Position : Zone
        - Des caractéristiques : nom, état,…
        - Des comportements selon type de personnage, permet de réagir aux contextes
        - Objectif définit :
            - Trouver 1 Objet
            - Atteindre une Zone définie
            - Survivre
    - D'un Etat-major
    - Des objets du Jeu
        - Position : Zone
        - Des caractéristiques : nom
        - Type d'objet : véhicules, visibilités, nourritures, pièges et trésors
- **Module de statistique pour mesurer les simulations**
    - Pour chaque tour
        - Les personnages ayant atteint leurs objectifs complets
        - Les personnages ayant récupéré leurs objets
        - Les personnages morts
        - Le classement des personnages par Point de Vie
        - La position et les caractéristiques de chaque personnage
        - Autres…
- **Une IHM pour montrer/visualiser la simulation à chaque tour**
    - Permet d'avancer au tour suivant
    - Permet de visualiser la progression de la simulation et des personnages
    - Permet de sauvegarder une **simulation en cours** (que l'on pourra recharger)

Démonstration

- Méthode ChargerSimulation(), à partir d'un fichier de données en XML.
- **Au début de la simulation** :
    - Point de vie des personnages
        - Au hasard
        - Tous les personnages ont le même nombre de points de vie
        - Nombre de points de vie selon le type du personnage
    - Positionnement des personnages
        - Au hasard

        - A une position commune à tous les personnages
        - Chacun sa position (prédéfinit)
    - Définition des objectifs pour chaque personnage (1 objet et 1 zone cible)
        - Au hasard
        - Tous les mêmes objectifs
        - Chacun son objectif
    - Stratégie de déplacement et de ramasser/déposer objet
        - Au hasard
        - La même pour tous les personnages
        - Chacun sa stratégie de déplacement selon le type du personnage
- **En cours d'exécution du personnage**
    - Chaque personnage ne perçoit que les zones autour de lui (les zones ayant un accès avec la zone où il est positionné)
    - S'il possède un objet de visibilité ou son type de personnage possède des capacités, un personnage peut percevoir plus de zones.
    - Chaque personnage perçoit le contenu (autres personnages + objets +caractéristiques de la zone) de toutes zones qui lui sont visibles
    - **AnalyseSituation(),** lui permet de préparer son exécution (se qu'il va poser ou ramasser et où il va se déplacer), si il doit réaliser un ordre ou définir son objectif…
    - **Exécution()**, Le personnage exécute le comportement adapté.
    - **RésoudreLesConflits()**

Processus de développement

- Equipe de 3 ou 4 étudiants
- Langage Java, C#, Delphi ou C++

Livraison attendue

- Un document fiche technique (Nom du produit, Les développeurs, le langage de développement, les documents livrés,…etc.)
- Un document expliquant le produit (1 page recto-verso)
- Un document de conception complet
- Un document expliquant pour chaque Pattern Utilisé
    - Les classes utilisées
    - Diagrammes de classes limités au Pattern
    - Intérêts du Pattern
- Un exécutable (ou une installation) sur un environnement Windows
- L'ensemble du code produit + librairies supplémentaires + indications pour tout réinstaller

Soutenance (15 min)

- Pitch de 3 minutes (chrononométré)
- Démonstration de 5 minutes (montrer un scénario préparé)
- Questions sur la livraison
- Navigation dans le code (une fonctionnalité) présentée