

Nicolas Picard

Martin Pintiau

ESILV 3A TD N

Premier rendu de projet POOI

Nous avons réfléchi à beaucoup de possibilités qui permettraient de faciliter l'organisation du parc d'attraction Zombillénium. Le fait de pouvoir proposer un logiciel complet est bien plus pratique que le Excel déjà existant, et nous mettrons tout en œuvre afin de rendre le logiciel le plus complet possible à Zombillénium.

Vous trouverez ci-joint les cartes CRC et le diagramme montrant la modélisation du projet. Nous avons détaillé un maximum tout en cherchant à rester adaptable aux futures contraintes que nous allons rencontrer. Evidemment, les interactions entre chaque classe risque d'évoluer avec les fonctionnalités qui seront ajoutées au cahier des charges, nous avons également imaginé celles qui nous semblaient les plus probables.

Classe : ParcAttraction	
Classe mère : NULL	
Classe fille : NULL	
Attributs : string nom, string adresse, int capital, list<personnel> direction, list<personnel> personnel, list<attraction> attractions, list<maintenance> maintenances, list<salle> salles, int nbVisiteur, int nbActionnaires	Collaborateur : Personnel, Attraction, Salle
Propriétés :	
Méthodes : void ajouterAttraction(Attraction new_attraction), void retirerAttraction(Attraction old_attraction), void licencier(personnel renvoyé), recruter(personnel embauché), void modifierNbClient(int newNbClient)	

Classe : Attraction	
Classe mère : NULL	
Classe filles : Roller Coaster, Dark Ride, Spectacle, Boutique	
Attributs : int id, string nom, Boolean estFermee, Boolean estEnMaintenance, maintenance maintenance, int nbMonstresNecessaires, list<Monstre> typeRequis, list<monster> employes, int nbEntree	Collaborateur : Maintenance, Monstre, ParcAttraction
Propriétés :	
Méthodes : void mettreEnMaintenance(Maintenance new_maintenance), void finirMaintenance(),	

Classe : Roller Coaster	
Classe mère : Attraction	
Classe fille : NULL	
Attributs : string categorie, double tailleMin, int ageMin	Collaborateur :
Propriétés :	
Méthodes :	

Classe : Dark Ride	
Classe mère : Attraction	
Classe fille : NULL	
Attributs : int duree, string deplacement	Collaborateur :
Propriétés :	
Méthodes :	

Classe : Spectacle	
Classe mère : Attraction	
Classe fille : NULL	
Attributs : string categorie, Salle salle, int duree	Collaborateur : Salle
Propriétés :	
Méthodes : void annuler()	

Classe : Salle	
Classe mère : NULL	
Classe fille : NULL	
Attributs : string nom, int categorie, int nbPassages, boolean estLibre	Collaborateur : Spectacle, ParcAttraction
Propriétés :	
Méthodes : void libererSalle(), void occuperSalle()	

Classe : Boutique	
Classe mère : Attraction	
Classe fille : NULL	
Attributs : List<Article> catalogue	Collaborateur : Article
Propriétés :	
Méthodes : void vendre(Article articleVendu, int quantite), void commander(Article new_Article)	

Classe : Article	
Classe mère : NULL	
Classe fille : Souvenir, Nourriture	
Attributs : string nom, double prix, int stock	Collaborateur : Boutique
Propriétés :	
Méthodes :	

Classe : Souvenir	
Classe mère : Article	
Classe fille : NULL	
Attributs :	Collaborateur :
Propriétés :	
Méthodes :	

Classe : Nourriture	
Classe mère : Article	
Classe fille : NULL	
Attributs :	Collaborateur :
Propriétés :	
Méthodes :	

Classe : Maintenance	
Classe mère : NULL	
Classe fille : NULL	
Attributs : string type, int nbMonstresNecessaire, List<Monstre> typeRequis, int dureeMaintenance	Collaborateur : Monstre, Attraction
Propriétés :	
Méthodes :	

Classe : Personnel	
Classe mère : NULL	
Classe fille : Monstre, Sorcier	
Attributs : int matricule, string prenom, string nom, Boolean isMale, list<string> fonctions, Boolean peutEtreLicencie	Collaborateur : ParcAttraction
Propriétés :	
Méthodes : void ajouterFonctions(list<string> new_Fonctions), void supprimerFonctions(list<string> old_Fonctions)	

Classe : Monstre	
Classe mère : Personnel	
Classe filles : Fantôme, Minotaure, Momie, Démon, Loup Garou, Vampire, Zombie	
Attributs : Attraction affectation, int cagnotte	Collaborateur : Attraction, Maintenance
Propriétés :	
Méthodes : void incrementerCagnotte(int n), void decrementerCagnotte(int n), void changerAttraction(Attraction new_attraction)	

Classe : Fantôme	
Classe mère : Monstre	
Classe fille : NULL	
Attributs :	Collaborateur :
Propriétés :	
Méthodes :	

Classe : Minotaure	
Classe mère : Monstre	
Classe fille : NULL	
Attributs :	Collaborateur :
Propriétés :	
Méthodes :	

Classe : Momie	
Classe mère : Monstre	
Classe fille : NULL	
Attributs :	Collaborateur :
Propriétés :	
Méthodes :	

Classe : Démon	
Classe mère : Monstre	
Classe fille : NULL	
Attributs : int force	Collaborateur :
Propriétés :	
Méthodes : void modifierForce(int new_force)	

Classe : Loup Garou	
Classe mère : Monstre	
Classe fille : NULL	
Attributs : double indiceCruaute	Collaborateur :
Propriétés :	
Méthodes : void modifierCruaute (double new_Cruaute)	

Classe : Vampire	
Classe mère : Monstre	
Classe fille : NULL	
Attributs : double indiceLuminosite	Collaborateur :
Propriétés :	
Méthodes : void modifierLuminosite (double new_Luminosite)	

Classe : Zombie	
Classe mère : Monstre	
Classe fille : Squelette	
Attributs : int degreDeDecomposition, string couleur	Collaborateur : Squelette
Propriétés :	
Méthodes : void modifier degreDeDecomposition (int new_ Decomposition), void transformerEnSquelette(Squelette new)	

Classe : Squelette	
Classe mère : Zombie	
Classe fille : NULL	
Attributs :	Collaborateur :
Propriétés :	
Méthodes :	

Classe : Sorcier	
Classe mère : Personnel	
Classe fille : NULL	
Attributs : string tatouage, list<pouvoir> pouvoirs	Collaborateur : Pouvoir (Pour l'instant ne nous savons pas à quoi servent les sorciers mais nous supposons qu'ils collaboreront également avec Attraction et éventuellement avec Monstre)
Propriétés :	
Méthodes : void ajouterPouvoir(list <Pouvoir> newPowers), void supprimerPouvoir(list <Pouvoir> oldPowers), void modifierGrade(string newTatouage), void utiliserPouvoir(Pouvoir sortLancé)	

Classe : Pouvoir	
Classe mère : NULL	
Classe fille : NULL	
Attributs : string nom, int coutMagie	Collaborateur :
Propriétés :	
Méthodes :	

