

POO - grand devoir 2

- Pour ce projet vous allez travailler par équipes de 2-3 personnes
- Le projet est à rendre sur la plateforme Moodle. S'il y a des problèmes lors du téléchargement vers ou depuis la plateforme, contactez sur Teams votre assistant des travaux pratiques : Iulia Stanica ou Alexandru Bratosin.
- Un seul upload par équipe suffit!
- Date limite: 13eme semaine (Lundi, 13 janvier 8:00, affichée sur moodle) et présentée pendant le TP de cette semaine-la
- Vous devez être présent à la séance de TP respective pour recevoir les points. Un devoir mis sur Moodle et pas présenté, la note sera 0!
- Pour toute question concernant le sujet du projet ou les exigences, utiliser le groupe de Teams de POO ou envoyer un message sur le chat

Travail avec: classes, relations, interfaces graphiques

Obs!

- TPs avec Iulia Stanica - utilisation de JavaFX obligatoire
- TPs avec Alexandru Bratosin - utilisation de Java Swing obligatoire

Tâches:

Rappelez vous le jeu de survie du premier grand devoir! Vous devez implémenter l'interface graphique du jeu en 2D. Tous les détails, éléments et contraintes du premier grand devoir restent présents dans celui-ci aussi.

La première fenêtre va être un menu avec plusieurs boutons:

- Resume - fait un processus de Load (à partir d'un ou plusieurs fichiers) de la carte de jeu précédente et garde les ressources et les objets de l'inventaire du joueur
- New Game - génère une carte nouvelle (processus aléatoire, on ne doit pas avoir toujours la même carte)
- Options - permet la personnalisation de l'expérience en fonction de vos préférences (par ex. Changement de background, arrêt / démarrage musique etc.)
- Help - instructions à propos de contrôles dans le jeu, histoire, objectifs
- Exit - ferme le jeu

Le jeu principal va se passer de cette manière. La carte va être affichée comme une matrice sur laquelle le joueur peut se déplacer en utilisant les touches WASD. Chaque emplacement de la matrice peut être vide (défini sur 0) ou contenir un objet, vous pouvez décider de la représentation de la manière dont la matrice contient les objets ou les ennemis. Si le joueur rencontre un emplacement vide de la matrice, il peut fabriquer un objet sur cet emplacement vide. Si l'emplacement contient un objet, le joueur peut ramasser l'objet et vider l'emplacement. Si le joueur rencontre un ennemi, un combat commencera (le joueur frappe en premier, puis l'ennemi, cela continue jusqu'à ce que l'un d'eux meure). Le déplacement d'une cellule à l'autre / les actions effectuées doivent produire des changements en temps réel dans l'interface (ex si un objet est ramassé, la cellule respective va devenir vide)

- Tous les types d'objets doivent être affichés d'une manière graphique différente (images / icônes, couleurs etc) sur la carte

- Le combat doit se passer dans une fenêtre séparée, en incluant des animations suggestives
- La quantité de bois, pierre, nourriture du joueur doit être affichée dans l'interface
- Un inventaire avec les objets ramassés doit être affichée dans l'interface
- Vous devez concevoir un système de points / win-lose en fonction de vos critères (combat, ressources, objets rares etc.)
- Les classes du premier grand devoir doivent être intégrées avec leur logique pour aider la création du jeu
- En JavaFX, l'utilisation d'un fichier css / des instructions style css est obligatoire!

Points

- Scene Menu - 0.5p
- Chaque bouton avec fonctionnalités dans le menu (5 x 0.2) - 1p
- Scene combat - 1p
- Interface graphique plaisante pour le jeu – 2p
- Generation aleatoire des cartes - 0.5p
- Changements dans l'interface conformément aux mouvements et actions - 1.5p
- Systeme points / win-lose - 0.5p
- Inventaire - 0.5p
- GUI present dans la scene pour afficher les ressources - 0.5p
- Integration classes devoir 1 - 1p
- Toutes les verifications necessaires pour les cas extremes, pas de crash quand les commandes sont invalides - 1p