

TP1: Conseils de développement.

420-V31-SF – Programmation de Jeux Vidéo III

Un objet réel = une classe

- ❑ À ne pas faire: mettre deux objets réels ou plus dans une classe
- ❑ Un objet peut avoir des méthodes avec son propre type d'objet en paramètre
 - ❑ Exemple: `bool TesterCollision(Sphere& autreSphere);`

Une étape à la fois; tout doit toujours compiler

- ☐ Faites une petite modification à la fois
- ☐ Assurez-vous que tout compile toujours.
- ☐ Pire erreur: écrire un tas de code un peu partout et espérer que tout va compiler du premier coup.
- ☐ Au contraire on peut enchaîner les bugs sans fin.
- ☐ Votre UML de départ est un plan, un guide, mais il peut changer en cours de développement.

Gardez les choses simples

- ☐ Évitez les pointeurs.
- ☐ Utilisez des tableaux fixes. Évitez les collections dynamiques.
- ☐ Pour savoir si vous les objets de votre collection de votre tableau sont actifs, utilisez
 - ☐ la console (attention, utiliser la console dans la loop de jeu le fera lagger brutalement).
 - ☐ du texte à l'écran (vous en avez déjà au départ; gardez-le et utilisez-le)
- ☐ Sur les objets à activer
 - ☐ Normalement dès que vous avez trouvé un objet à activer, vous le faites et sortez de la recherche.
 - ☐ Si votre recherche se fait au complet sans avoir un objet à activer alors vous utilisez tous vos assets.

Diagramme de classes: Conseils

Commencez tôt!

Gardez-le ouvert!

- ☐ Rapidement, faites l'agencement des classes
- ☐ Mettez toutes les classes sans méthodes ni attributs et faites les relations
- ☐ Puis réfléchissez aux attributs
- ☐ Les méthodes devraient venir d'elles-mêmes.
- ☐ Travailler avec un plan: beaucoup mieux que sans.

Ne perdez pas de temps

- ☐ Vous avez 15 jours
- ☐ C'est peu
- ☐ Mais ça se fait si on ne perd pas de temps.