

Exercice 1

Vous devez concevoir un jeu vidéo dans lequel le joueur doit s'occuper de personnages. A sa création, un personnage est un enfant. Il devient un adulte au bout de $t1$ unités de temps. Il meurt au bout de $t2$ unités de temps.

Le joueur peut faire des mariages ou des divorces. Seuls les adultes célibataires, divorcés ou veufs peuvent se marier. Seuls les personnages mariés peuvent divorcer. Quand un personnage marié meurt, son conjoint devient veuf.

Votre travail : Dessinez le diagramme d'états-transitions modélisant les successions d'états possibles pour un personnage. Dessinez le diagramme de classes correspondant.

Exercice 2

Vous devez maintenant compléter la conception afin que le joueur puisse nourrir ses personnages. Un personnage est bien-nourri, sous-nourri ou sur-nourri, en fonction de la fréquence à laquelle le joueur le nourrit, et sa durée de vie diminue lorsqu'il est sur- ou sous-nourri :

- un personnage bien-nourri devient sous-nourri s'il n'a pas été nourri depuis $t3$ unités de temps, et il devient sur-nourri s'il est nourri moins de $t4$ unités de temps après le dernier repas ;
- un personnage sous-nourri redevient bien-nourri s'il est nourri avant $t5$ unités de temps, sinon sa durée de vie $t2$ est diminuée de δt unités de temps toutes les $t5$ unités de temps ;
- un personnage sur-nourri redevient bien-nourri s'il n'est pas nourri pendant $t6$ unités de temps, sinon sa durée de vie $t2$ est diminuée de δt unités de temps à chaque fois qu'il est nourri.

A sa création, un personnage est bien-nourri.

Votre travail : Dessinez le diagramme d'états-transitions modélisant les successions d'états possibles par rapport à la nourriture puis intégrez ce diagramme avec celui de l'exercice 1, et complétez le diagramme de classes.

Exercice 3

La représentation graphique d'un personnage (sa hauteur, sa largeur et sa couleur) dépend de son état. Par ailleurs, chaque personnage émet des sons dont la nature, la fréquence et la tonalité dépendent de son état.

Votre travail : Complétez le diagramme de classes de la partie 2 afin d'intégrer les aspects liés aux représentations graphique et sonore. Vous distinguerez le modèle, qui définit l'état interne d'un personnage, des vues graphique et sonore, qui définissent ses représentations graphique et sonore en fonction de son état. Vous utiliserez le design pattern *Observer* afin de répercuter les modifications d'états dans le modèle sur les vues.

Rappel du design pattern *Observer* vu en cours :

