README

Nous avons fait les étapes de 1 à 8 du projet, sans l'étape 6 (Qt).
Nous estimons la durée moyenne de travail par semaine de 5h par personne.

Correspondance : énoncés / noms choisis
agit_sur() / agir_sur()

Guide de compilation :
Ci-dessous nos différents programmes de tests des différentes semaines :

P4) test de la classe Vecteur. Le but est de tester nos opérateurs algébriques de base, nos fonctions mathématiques élémentaires utiles dans la suite de notre projet et nos constructeurs.

[Dans le **main** de testVecteur.cc, on construit notamment :
a) vect0 : un vecteur de taille n rempli uniquement de 0 (invoquant le 1e constructeur de la classe Vecteur)
b) vect1: un vecteur à 3 dimension (invoquant le 2e constructeur de la classe Vecteur)
c) vect5 : un vecteur construit via une liste d'initialisation (invoquant le 3e constructeur de la classe Vecteur)
d) vect6, un copie de vect5 (grâce au constructeur de copie)]

Pour lancer ce test, taper la commande : **./testVecteur**

P5) test de la classe Balle. Le but est de tester l'affichage et l'effet de la gravité de notre premier objet, ainsi que ses méthodes.

[Dans le **main** de testBalle.cc, on invoque le constructeur de balle dont les arguments sont dans l'ordre : la masse, la masse volumique, le rayon, la force, les paramètres (ici, position) et les paramètres dérivés (ici, vitesse). (la masse est ajustée grâce à la masse volumique et au rayon : se référer au fichier CONCEPTION, **ObjetMobile**, pour en savoir plus sur ce constructeur)]

Pour lancer ce test, taper la commande : **./testBalle**

P6) test de la classe Pendule (affichage, effet de la gravité, et méthodes de Pendule)

[Dans le **main** de testPendule.cc, on invoque le constructeur de Pendule dont les arguments sont dans l'ordre : la masse, la masse volumique, le rayon, la longueur, les paramètres (ici, l'angle), le frottement, les paramètres dérivés (ici, vitesse angulaire), la force, l'origine (point d'attache) et la direction. (la masse volumique est ajustée grâce à la masse et au rayon : se référer au fichier CONCEPTION, **ObjetMobile**, pour en savoir plus sur ce constructeur)

Pour changer le temps de l'intégration, il faut changer la variable **T_i**
Pour changer le nombre d'itérations, il faut changer la variable **nbr_iter**]

Pour lancer ce test, taper la commande : **./testPendule**

P6) test des sous-classes de Obstacles

[Dans le **main** de testObstacle.cc, on invoque le constructeur de Plan dont les arguments sont dans l'ordre : l'origine, vecteur normal.
Le constructeur de Brique dont les arguments sont dans l'ordre : l'origine, la longueur, la largeur, la hauteur.
Et le constructeur de PortionPlan dont les arguments sont dans l'ordre : l'origine, vecteur normal, la longueur, la largeur, vecteur parallèle à la longueur.]

Pour lancer ce test, taper la commande : **./testObstacles**

P7) test de la classe IntegrateurEuleurCromer

[Dans testIntegrateur1.cc, on teste les intégrations sur une "masse" (rayon=0), puis sur une "vraie" Balle dans testIntegrateur2.cc.
Pour changer le temps de l'intégration, il faut changer la variable **T_i**
Pour changer le nombre d'itérations, il faut changer la variable **nbr_iter**]

Pour lancer ces tests, taper la commande : **./testIntegrateur1** puis **./testIntegrateur2**

P7) test de la classe Ressort (affichage, effet de la gravité, et méthodes de Pendule)

[Dans le **main** de testRessort.cc, on invoque le constructeur de Ressort dont les arguments sont dans l'ordre : la masse, la masse volumique, le rayon , la constante d'élasticité, les paramètres, le frottement, les paramètres dérivées, la force, l'origine et la direction. (la masse volumique est ajustée grâce à la masse et au rayon: se référer au fichier CONCEPTION, **ObjetMobile**, pour en savoir plus sur ce constructeur).
Pour changer le temps de l'intégration, il faut changer la variable **T_i**
Pour changer le nombre d'itérations, il faut changer la variable **nbr_iter**]

Pour lancer ce test, taper la commande : **./testRessort**

P9) test de la classe Système : ici on s'intéresse à faire interagir une collection d'objet mobiles sous l'effet de différentes contraintes (comme par exemple un champ de force, un obstacle, un autre objet mobile…).

[Dans le **main**, on construit un système avec un premièrement un intégrateur et deuxièmement un temps (pour changer le temps d'intégration, il faut donc modifier ce 2^{e} paramètre à la valeur voulue). Puis, les ObjetMobile, ChampForce, Obstacle sont ajouté au système grâce la méthode polymorphique ajoute_a()]

2 tests :
a) [Dans le **main**, on fait le premier test d'évolution (évolution de la Balle + choc avec le sol)]. Pour lancer ce test, taper la commande : **./Systeme**
b) [Dans le **main**, on fait le deuxième test d'évolution (évolution de Balle et Pendule + choc entre Balle et Pendule + choc entre Balle et sol). Pour lancer ce test, taper la commande : **./TestChocs_systeme**

P9) test des chocs entre ObjetMobile
[Dans le **main**, on effectue ici 3 chocs différents :
- CHOC 1 : nous testons le choc entre une Balle et un Plan
- CHOC 2 : nous testons le choc entre 2 Balles
- CHOC 3 : nous testons le choc entre une Balle est un Pendule]

Pour lancer ce test, taper la commande : **./testChocs**

P11) test des objets composés
[Dans le **main**, on teste ici le bon fonctionnement des objets composé (précisément du ventilateur) en testant ses méthodes ajoute_a(), son bon ajout au système et son affichage]
Pour lancer ces tests, taper la commande : **./ExerciceP11**

[Dans le **main**, on teste ici le bon fonctionnement du Ventilateur, principalement de sa méthode agir_sur (), et son constructeur.
Pour changer le temps de l'intégration, il faut changer la variable **T_i**
Pour changer le nombre d'itérations, il faut changer la variable **nbr_iter**]

Pour lancer ces tests, taper la commande : **./testVentilateur**

[Dans le **main**, on teste ici le bon fonctionnement du Vent, principalement de ses méthodes agir_sur (), get_vitesse_vent et get_direction_vent(), et son constructeur.
Pour changer le temps de l'intégration, il faut changer la variable **T_i**
Pour changer le nombre d'itérations, il faut changer la variable **nbr_iter**]

Pour lancer ces tests, taper la commande : **./testVent**

P12) test de : IntegrateurNewark et IntegrateurRungeKutta. Ces 2 intégrateurs sont en effet plus précis que les 2 autres.

[Dans le main de ExerciceP12.cc, on a directement fait les intégrations (pour un système et balle) pour les 2 intégrateurs]

Pour lancer ce test, taper la commande **: ./ExerciceP12**