TACT factory

mobile agency

Fournisseur **souple** et **réactif**
d'applications mobiles innovantes
avec une **méthodologie** projet
rigoureuse.

Présentation

Mickael Gaillard (Architecte logiciel)
mickael.gaillard@tactfactory.com

Yoan Pintas (Lead Developper)
yoan.pintas@tactfactory.com

Erwan LeHuitouze (Lead Developper)
erwan.lehuitouze@tactfactory.com

Antoine Crônier (Developper)
antoine.cronier@tactfactory.com

Les variables - TP1

Algorithme :

- On souhaite déverrouiller un cadenas à code numérique disposant d'une molette numéroté
- On représentera notre doigts comme étant un curseur permettant de saisir les chiffres nécessaires à l'ouverture du cadenas
- La séquence à réaliser pour ouvrir le cadenas est « 12 : 9 : 14 : 8 »

Les variables – TP1

Une solution est :

```
ALGORITHME TP1
VARIABLE
    ENTIER : curseur
DEBUT
    curseur ← 12
    curseur ← curseur – 3
    curseur ← curseur + 5
    curseur ← curseur - 6
FIN
```

Les variables – TP2

Algorithme :

- Jasmine et Aladin possèdent tout les deux un panier de pomme
- Jasmine a 7 pommes dans son panier et Aladin 5
- Il regroupe leurs pommes dans un nouveau panier

Les variables – TP2

Une solution est :

```
ALGORITHME TP2
VARIABLE
    ENTIER : panierJasmine ← 7
    ENTIER : panierAladin ← 5
    ENTIER : panierGroupe
DEBUT
    panierGroupe ← panierJasmine + panierAladin
FIN
```

Les variables – TP3

Algorithme :

- Sur la base du TP2 modifier votre algorithme pour permettre à Jasmine et Aladin d'avoir le nombre de pomme qu'ils veulent dans leurs panier
- Afficher à l'utilisateur le nombre de pomme contenu dans le panier ou ils ont mis en commun leurs pommes

Les variables – TP3

Une solution est :

```
ALGORITHME TP3
VARIABLE
    ENTIER : panierJasmine
    ENTIER : panierAladin
    ENTIER : panierGroupe
DEBUT
    Lire(panierJasmine)
    Lire(panierAladin)
    panierGroupe ← panierJasmine + panierAladin
    Ecrire(panierGroupe )
FIN
```

Les variables – TP4

Algorithme :

-Vous devez créer un programme de calcul qui demande à un utilisateur de saisir 2 chiffres entiers

-Si les 2 chiffres sont identique le programme retourne la somme des 2 chiffres

-Si le premier nombre saisi est négatif et le deuxième est positif on retourne le produit des 2 chiffres

-Si les 2 chiffres sont positif ou négatif et qu'un des chiffres est supérieur à 10 ou inférieur à -10 on retourne la division des 2 chiffres

Les variables – TP4

Une solution est :

```
ALGORITHME TP4
VARIABLE
        ENTIER : chiffre1, chiffre2
        REEL : resultat
DEBUT
        Ecrire(" Veuillez saisir 2 chiffres ")
        Lire(chiffre1)
        Lire(chiffre2)
        SI chiffre1 = chiffre2 ALORS
                resultat ← chiffre1 + chiffre2
        FINSI
        SI chiffre1 < 0 ET chiffre2 >= 0 ALORS
                 resultat ← chiffre1 * chiffre2
        FINSI
        SI ((chiffre1 > 0 ET chiffre2 > 0) OU (chiffre1 < 0 ET chiffre2 < 0)) ET
                ((chiffre1 > 10 OU chiffre2 > 10) OU (chiffre1 < -10 OU chiffre2 < -10)) ALORS
                 resultat ← chiffre1 / chiffre2
        FINSI
        SI resultat != null ALORS
                Ecrire(resultat)
        FINSI
FIN
```

Les variables – TP5

Algorithme :

- Sur la base du TP4 modifier l'algorithme en utilisant SI SINON SI SINON
- Faire en sorte que l'algorithme retourne toujours une valeur
- L'ordre d'importance des opérations à réaliser est l'ordre des points du TP4

Les variables – TP6

Algorithme :

-Un étudiant veut faire cuire des côtes d'agneaux

-En fonction de la cuisson bleu indice 1, saignant indice 2, cuit indice 3, à point indice 4 et brûlé indice 5 l'étudiant effectuera différentes actions

-Si le repas n'est pas à point il laissera le repas cuire

-Si le repas est brûlé alors il sera déçu

-Si le repas est à point alors il sera content

-L'étudiant lancera l'algo à chaque fois qui voudra connaitre l'état de son repas

Les variables – TP6

Une solution est :

```
ALGORITHME TP6
VARIABLE
    ENTIER : repas
DEBUT
    Lire(repas)
    SELON repas FAIRE
        1 : Ecrire("laisser cuire")
        2 : Ecrire("laisser cuire")
        3 : Ecrire("laisser cuire")
        4 : Ecrire("je suis content")
        5 : Ecrire("je suis déçu")
    FINSELON
FIN
```

Les variables – TP7

Algorithme :

-Une machine a café propose une liste de sélection avec des indices de 1 à 8

-Vous devez réaliser un algo qui selon la sélection de l'utilisateur effectuera les tâches suivantes

-Sélection du bac de grain de café à utiliser (4 bacs)

pour sélection indice 1 à 4 bac 1, indice 5 à 6 bac 2, indice 7 bac 3, indice 8 bac 4

-Moudre le grain(2 concasseur)

pour les bacs 1 et 2 concasseur 1, bac 3 concasseur 2

-Verser 10 unités de grain moulu pour les indices 1, 3,4, 6

-Verser 25 unités de grain moulu pour les indices 2, 5,7

-Verser 12 unités du bac 4 pour l'indice 8

Les variables – TP9

Algorithme :

- Ecrire un algorithme qui gère l'attaque dans un jeu
- Un personnage dans le jeu possède les attributs suivant :
  - des points de vie, une armure, une valeur d'attaque, des points d'action
  - il existe 5 armes avec des valeurs d'attaque différentes :
    - concasseur : 3
    - pelle : 2
    - gatling : 10
    - batte de cricket : 1
    - blaster : 6

Les variables – TP9

- Il existe 3 types d'armure :
  - Gilet bleu : 1
  - Armure de cuir : 2
  - Armure de plaque : 4
- Lors d'une attaque entre deux personnages on soustrait les dégâts d'attaque à la valeur d'armure du défenseur
- Les points d'action permettent à un personnage de lancer des attaques
- Suivant les différentes armes il faut plus ou moins de point d'action pour réaliser une attaque :
  - concasseur : 4
  - pelle : 1
  - gatling : 6
  - batte de cricket : 1
  - blaster : 3

Les variables – TP9

- S'équiper d'une arme coute 1 point d'action
- A chaque début de tour aucune arme n'est équiper pour chacun des personnages
- Un personnage fini son tour lorsqu'il ne peut plus effectuer d'action
- Un personnage mort n'attaque pas
- Les personnages attaquent dans l'ordre du tableau des personnages et se battent
- Contre le personnage suivant

Afin de tester votre algo :

-Ecrit le résultat de l'ensemble de vos test et des valeurs que prennent vos variables pour:

- 3 personnages :
- Personnage 1 : arme pelle, gilet bleu, point de vie 9, point d'action 7
- Personnage 2 : arme blaster, armure de cuir, point de vie 4, point d'action 12
- Personnage 3 : arme concasseur , armure de plaque, point de vie 15, point d'action 8

Les variables – TP10

Algorithme :

-Sur la base du TP9 utiliser un seul tableau multidimensionnel

Les variables – TP11

Algorithme :

- On cherche à créer un algorithme qui génère une carte de touché coulé pour 2 joueurs
- On utilisera un tableau à 2 dimensions afin de gérer la carte du jeu et son contenu
- La première dimension référencera l'axe des abscisses sur 24 cases
- La deuxième dimension référencera l'axe des ordonnées sur 18 cases
- La valeur de la case x, y sera le type de vaisseau présent dans la case, par joueur on a :
    - corvette : identifiant 1, taille 1, nombre 1
    - destroyer : identifiant 2, taille 3, nombre 2
    - croiseur : identifiant 3, taille 4, nombre 2
    - porte-avion : identifiant 4, taille 6, nombre 1
- Attention lors de l'insertion des vaisseaux ils doivent toujours être contenu dans la carte et ne doivent pas se superposer

Les variables – TP12

Algorithme :

- Ecrire une fonction qui prend un verbe du premier groupe et qui le conjugue au présent de l'indicatif
- La fonction devra retourner un tableau de chaine contenant le verbe conjugué pour chacun des pronoms
- Ecrire un algorithme qui utilise cette fonction et qui affiche le tableau de résultat à l'utilisateur

Les variables – TP13

Algorithme :

- Ecrire une fonction de recherche qui remplace les caractères « et » dans une chaine et retourne la chaine modifié
- Ecrire un algorithme qui utilise cette fonction et qui affiche la chaine avant et après modification

Les variables – TP14

Algorithme :

- Sur la base du TP13 utiliser la mécanique des pointeurs et modifier le type de retour de la fonction pour renvoyer le nombre de modification qui ont été appliqués

Les variables – TP15

Algorithme :

-Sur la base du TP14 utiliser la mécanique des fichiers pour permettre d'utiliser directement un fichier plutôt qu'une chaine

Les variables – TP16

Algorithme :

- Ecrire une fonction qui prend un mot de passe en paramètre, un fichier et un entier
- La fonction accède au fichier afin d'utiliser les mots qu'il contient pour essayer de trouver le mot de passe
- L'entier passé en paramètre indique le nombre de combinaison de concaténation à réaliser avec les mots du fichier
- Lors d'une concaténation on testera les différentes disposition possible des mots du fichier
- La fonction renverra un tableau indiquant la position des mots utilisés pour trouver la solution de façon ordonnée
- Si aucun résultat n'est trouvé la fonction renverra un tableau d'une case avec la valeur -1

Les variables – TP17

Algorithme :

-Sur la base du TP9 utilisez les notions d'objet pour réadapter l'algorithme

Les variables – TP18

Algorithme :

-On souhaite réaliser un algorithme qui permet de faire rouler des trains

-On définira une classe « Train » qui aura comme attribut une machine de tête, des wagons et un type de rails sur lequel il roulera

-On définira une autre classe « CheminDeFer » qui aura comme attribut des types de rails, des gares et des trains

-Les règles de gestion de l'algorithme à réaliser sont les suivantes :

- Un train peut rouler que sur un type de rails qui lui correspond
- Un train ne peut rouler que dans un sens jusqu'à une gares
- Un seul train peut rouler dans un seul sens sur des rails
- Plusieurs train peuvent rouler dans le même sens jusqu'à une gare
- Une gare peut être fermé ou ouverte

Les variables – TP18

- Des rails peuvent être en réparation
- Les wagons peuvent contenir des marchandises et des animaux ou bien des passagers
- Le wagon de tête est capable de tracter un poids correspondant au poids des wagon et de leur marchandise à une vitesse maximum
- Si le poids que le wagon peut tracté est dépassé on contera 1km/h de moins pour 1/Tonne
-Les gares contiennent les heures d'arriver des différents train et l'ordre dont les trai circule sur le chemin de fer est lié à leur vitesse d'une gare à l'autre

Les variables – TP19

Algorithme :

-On souhaite réaliser un algorithme qui demande à l'utilisateur de choisir un nombre entre 1 et 100.

-L'algorithme devra alors trouver le nombre choisi en demandant à l'utilisateur si le nombre recherché est plus grand, plus petit ou égale au nombre proposé par l'algorithme

-On se basera sur le principe de la dichotomie en indiquant le nombre maximum d'essaie que l'algorithme doit réaliser

Les variables – TP20

Algorithme :

-On souhaite réaliser un algorithme qui recherche un mot de passe associé à un compte utilisateur

-On dispose de 3 fichiers « Users1 », « Administrateur » et « sam »

-Ces 3 fichiers sont triés par ordre alphabétique et associe un utilisateur à son mot de passe

-Dans le fichier « Users1 » on retrouve la nomenclature suivante : « unUser : sonPwd »

-Dans le fichier « Administrateur » on retrouve la nomenclature suivante : « unUser*sonPwd »

-Dans le fichier « sam » on retrouve la nomenclature suivante : « unUser[sonPwd] »

-Attention pour le fichier sam les nom d'utilisateur et les mots de passe sont chiffré, il faut utiliser l'objet « Md5 » avec la fonction « Md5.Decrypt(CHAINE:param) » pour récupérer les Information en claire

Les variables – TP22

Algorithme :

- On souhaite réaliser une fonction de tri à bulles pour ranger un tableau de chien
- Un chien possède 3 attributs : une taille, un poids et une race
- Il y a 5 races de chien : Labrador, Bulldog, Yorkshire, Caniche et Boxer
- La fonction servira à ordonnées des listes de chien lors des concours canin afin qu'ils puissent concourir par race, taille et poids

Les variables – TP23

Algorithme :

-On souhaite réaliser une fonction de tri quicksort pour des tableaux d'entier

Les variables – TP24

Algorithme :

-On souhaite réaliser un algorithme qui permet de jouer à la bataille de carte

-On possède un jeu de 32 cartes et le jeu se déroule avec 2 joueurs

-Les objets à utiliser sont des « piles »

-Il faudra distribuer les cartes et former une « pile » par joueur, le paquet étant une pile de 32 cartes

-Le jeu consiste à ce que les deux joueurs retourne la première carte de leur pile et que l'on compare la valeur des deux cartes

-Le joueur possédant la carte la plus forte récupère la carte de son adversaire et la place dans une autre pile

-Si il y a égalité les joueurs pose une carte supplémentaire face caché puis une carte face ouverte et on compare la valeur de ces cartes, on répète ceci tant que personne gagne

-Une fois la pile d'un joueur vide il utilise alors la pile de ses mains gagnés

-Le jeu se terminera lorsque l'un des joueurs aura récupéré toutes les cartes du jeu

Les variables – TP25

Algorithme :

-On souhaite réaliser une classe Voitures qui gère une liste de Voiture

-Cette classe permettra de gérer une liste de Voiture en permettant l'ajout de voiture en fin de liste ou à un indice déterminé ou en début de liste

-On pourra supprimer une Voiture par son ID ou par son nom

-On pourra supprimer toutes les Voitures dont l'âge est supérieur à une valeur

-On pourra connaitre le nombre de Voiture total

-On pourra connaitre la liste de toutes les voitures qui ont fini première à une course

Les variables – TP26

Algorithme :

-Sur la base du TP25 on souhaite réutiliser la classe Voitures et lui ajouter une fonctionnalité de tri et de recherche ainsi que transformer la liste chainée en liste doublement chainée

-On pourra trier la liste de Voiture par âge ou par nom

-On pourra effectuer une recherche sur la liste de Voiture en fonction d'un des attributs

Les variables – TP27

Algorithme :

- On souhaite créer un zoo
- Le zoo est composé d'animaux et de cage pouvant les accueillir
- Il y a différent types d'animaux :
  - Mammifère (singe, lion, dauphin)
  - Reptile (serpent, lézard)
  - Oiseau (perroquet, autruche, pélican)
  - Poisson (raie, carpe)
- Il y a différent type de cage :
  - Terrestre (aride, tempéré, marre)
  - Aérienne
  - Aquarium (haut fond, côtier)

Les variables – TP27

Algorithme :

- Les animaux possèdent différentes caractéristiques :
  - Taille
  - Poids
  - Nom
  - Type
  - Race
  - Régime alimentaire
  - Male / Femelle
  - Une liste de leur enfant
  - Un père
  - Une mère
  - Une liste des cages ou il a séjourné
  - La fréquence de leur repas
  - La date de leur dernier repas
  - La date de leur prochain repas
  - Une liste des animaux avec lesquels il peuvent cohabiter
  - Espace minimum en m^3

Les variables – TP27

Algorithme :

- Une cage possède plusieurs attribut :
  - Dimension x,y,z
  - Liste des animaux présents
  - Espace habitable en m^3
- Plusieurs règles de gestion doivent être prises en compte :
  - Un animal doit être placé dans une cage correspondant à ces attributs
  - Un animal doit être mis en contact qu'avec des animaux avec qui la cohabitati
    est possible
- Dans l'algorithme principal de votre code créer votre propre zoo

Les variables – TP28

Algorithme :

- On souhaite réaliser un jeu médiéval fantastique
- Le jeu comprend 2 types de personnages, les joueurs et les monstres
- Un personnage et un monstre peuvent avoir une classe :
  - Le ranger
  - L'archer
  - Le paladin
  - Le sorcier
- Il y a plusieurs type de monstre :
  - Le gobelin
  - L'ogre
  - Le troll
  - L'elfe noir
  - Le kobold
  - Le hérisson nain avec une canne et une jambe cassé

Les variables – TP28

Algorithme :

- Un personnage possède plusieurs attributs :
  - Santé
  - Attaque à distance
  - Dégâts
  - Armure
  - Liste d'arme
  - Point de mouvement
  - Attaque au corps à corps
  - Point d'action
  - Esquive
  - Liste de sort
- Le jeu se décompose en niveau
- Un niveau est représenté par une carte ou on positionne des obstacles et des monstres
- Un monstre possède une agro (distance à laquelle il attaque un joueur)
- Vous devez prévoir des fonctions pour sauvegarder et charger les parties

Les variables – TP28

Algorithme :

- Vous êtes libre d'imaginer les attributs complémentaire nécessaire à la réalisation d
  cet exercice
- Dans votre algorithme principale vous initialiserez le jeu et permettrez à l'utilisateu
  de jouer

Les variables – TP29

Algorithme :

- Sur la base du TP 28 il faut modifier l'intégralité du système de sauvegarde pour utiliser des curseurs
- Vous devez créer le MCD de votre base de données
- Vous devez créer les différentes requêtes nécessaires à l'utilisation de votre base de données dans votre algorithme

Change log

- Antoine CRONIER 2015 : Initial document