

## LIF3 - Introduction au projet

### Printemps 2015-2016 BlackJack

1

## Un jeu de Black Jack

- Un jeu de 52 cartes
- Un joueur joue contre la banque
- L'objectif est de s'approcher de 21 points sans les dépasser, en tirant des cartes
- Les cartes de 2 à 10 valent leur valeur, les figures valent 10, l'as vaut 1

N. Guin

Licence Lyon1 - UE LIF3

2

## Déroulement d'une partie

- La banque tire une carte
- Le joueur tire deux cartes
- Le joueur tire ensuite autant de cartes qu'il le veut
- S'il dépasse 21 il a perdu
- Sinon la banque joue, jusqu'à dépasser 17
- On compare les résultats du joueur et de la banque

N. Guin

Licence Lyon1 - UE LIF3

3

## Ce que vous allez réaliser

```
----- Jeu de la banque -----
Cartes : ((As Carreau))
Score : 1
----- Jeu du joueur -----
Cartes : ((Valet Carreau) (4 Carreau))
Score : 14
-----
Voulez-vous une autre carte ? (o/n)
o
----- Jeu de la banque -----
Cartes : ((As Carreau))
Score : 1
----- Jeu du joueur -----
Cartes : ((Valet Carreau) (4 Carreau) (5 Pique))
Score : 19
-----
Voulez-vous une autre carte ? (o/n)
n
La banque joue...
----- Jeu de la banque -----
Cartes : ((As Carreau) (2 Pique) (10 Coeur) (As Coeur) (Roi Pique))
Score : 24
----- Jeu du joueur -----
Cartes : ((Valet Carreau) (4 Carreau) (5 Pique))
Score : 19
-----
La banque a dépassé 21: vous gagnez! Voulez-vous faire un autre tour? (o/n)
n
o
```

N. Guin

Licence Lyon1 - UE LIF3

4

## Ce que vous allez réaliser

```
----- Jeu de la banque -----
Cartes : ((6 Pique))
Score : 6
----- Jeu du joueur -----
Cartes : ((5 Carreau) (Dame Pique))
Score : 15
-----
Voulez-vous une autre carte ? (o/n)
o
----- Jeu de la banque -----
Cartes : ((6 Pique))
Score : 6
----- Jeu du joueur -----
Cartes : ((5 Carreau) (Dame Pique) (As Pique))
Score : 16
-----
Voulez-vous une autre carte ? (o/n)
o
----- Jeu de la banque -----
Cartes : ((6 Pique))
Score : 6
----- Jeu du joueur -----
Cartes : ((5 Carreau) (Dame Pique) (As Pique) (7 Pique))
Score : 23
-----
Vous avez dépassé 21! vous perdez...
Voulez-vous faire un autre tour? (o/n)
o
```

N. Guin

Licence Lyon1 - UE LIF3

5

## Ce que vous allez réaliser

```
----- Jeu de la banque -----
Cartes : ((3 Pique))
Score : 3
----- Jeu du joueur -----
Cartes : ((7 Trefle) (As Trefle))
Score : 8
-----
Voulez-vous une autre carte ? (o/n)
o
----- Jeu de la banque -----
Cartes : ((3 Pique))
Score : 3
----- Jeu du joueur -----
Cartes : ((7 Trefle) (As Trefle) (9 Coeur))
Score : 17
-----
Voulez-vous une autre carte ? (o/n)
n
La banque joue...
----- Jeu de la banque -----
Cartes : ((3 Pique) (4 Pique) (3 Carreau) (9 Trefle))
Score : 19
----- Jeu du joueur -----
Cartes : ((7 Trefle) (As Trefle) (9 Coeur))
Score : 17
-----
La banque fait mieux que vous: vous perdez... Voulez-vous faire un autre tour? (o/n)
n
Au revoir...
```

N. Guin

Licence Lyon1 - UE LIF3

6

## Décomposition du projet

- Que faut-il que le programme sache faire ?
  - Manipuler des cartes
- Comment représenter les cartes ?
  - Une carte est caractérisée par
    - sa hauteur (as, valeurs de 2 à 10, valet, dame, roi)
      - Valet = 11, dame = 12, roi = 13
    - sa couleur (carreau, cœur, pique, trèfle)
  - Une carte sera représentée par un couple (hauteur, couleur)
  - Donc '(11 3) représente le valet de pique

## Représentation interne/ représentation externe

- Une carte est représentée par un couple de nombres pour la machine
- Il faut une représentation plus naturelle pour dialoguer avec l'utilisateur

→ fonctions pour faire l'interface

```
(carte_affichable '(12 3)) → (Dame Pique)
(liste_cartes_affichables '((12 3) (5 1)))
→ ((Dame Pique) (5 Carreau))
```

## Décomposition du projet

- Quelles opérations le programme doit-il savoir faire sur les cartes pour jouer au blackjack ?
- On pourrait essayer de distinguer
  - des opérations de base sur les cartes qu'on pourra réutiliser pour un autre jeu de cartes
  - des opérations spécifiques au blackjack

## Opérations de base

- Créer un jeu de 52 cartes

```
(creer_52_cartes) → ((1 1) (2 1) ... (12 4) (13 4))
On va utiliser une fonction intermédiaire :
```

```
(cartes_suivantes '(8 4))
→ ((9 4) (10 4) (11 4) (12 4) (13 4))
```

- Piocher des cartes (sur le dessus de la pioche)

```
(piocher_n_cartes '((13 4) (5 3) (1 1) (6 3)) 2)
→ (((13 4) (5 3)) ((1 1) (6 3)))
```

→ liste des cartes piochées, liste des cartes restantes

- Se défausser (remettre des cartes en dessous de la pioche)

```
(defausser_cartes '((1 1) (1 2)) '((3 4) (4 3)))
→ ((1 1) (1 2) (3 4) (4 3))
```

## Opérations de base

- Mélanger un jeu de cartes

```
(melanger_jeu_cartes '((10 1) (4 4) (5 2)))
→ ((10 1) (5 2) (4 4))
```

- On va procéder en plusieurs étapes :

- Il faut tirer une carte au hasard pour en faire la première carte du nouveau jeu mélangé.
- Pour cela, on choisit au hasard l'indice  $i$  de la carte à tirer, et l'on extrait du jeu la carte d'indice  $i$ .

## Opérations de base

- (extraire\_ieme\_carte '((1 1)(12 3)(14 4)) 2)

```
→ ((12 3) ((1 1)(14 4)))
```

- (tirer\_une\_carte '((1 2)(13 4)(5 3)(1 1)(6 3)))

```
→ ((6 3) ((1 2)(13 4)(5 3)(1 1)))
```

- Fonction prédéfinie pour le hasard `random(5)`

```
→ 0, 1, 2, 3 ou 4
```

## Opérations spécifiques au blackjack

- Représentation de l'état du jeu : quelle structure de données ?
- Liste de 3 listes :  
(cartes\_de\_la\_banque cartes\_du\_joueur cartes\_de\_la\_pioche)
- Primitive de construction :  
initialisation du jeu : creer\_blackjack
- Primitives d'accès :  
cartes\_joueur / cartes\_banque / cartes\_pioche
- Primitives de mise à jour :  
banque\_pioche\_n / joueur\_pioche\_n

N. Guin

Licence Lyon1 - UE LIF3

13

## Opérations spécifiques au blackjack

- Connaître le nombre de points d'une main de cartes  
(total\_cartes '((1 3) (13 4) (5 4))) → 16
- faire\_jouer\_banque : tirer des cartes en mettant à jour le blackjack jusqu'à atteindre 17 points
- tout\_defausser : défausser toutes les cartes afin de recommencer à jouer

N. Guin

Licence Lyon1 - UE LIF3

14

## Interactions avec l'utilisateur

- Afficher à l'écran : fonction prédéfinie display

Ex :

```
> (display "Bonjour")
Bonjour
> (display 'Bonjour)
Bonjour
> (define a 'bonjour)
> (display a)
Bonjour
```

- Passer à la ligne : fonction prédéfinie newline  
Ex : (newline)

N. Guin

Licence Lyon1 - UE LIF3

15

## Interactions avec l'utilisateur

- Lire au clavier : fonction prédéfinie read

Ex :

```
> (read)
toto
```

- Utilisation du résultat de la fonction :  
(let ((reponse (read)))  
 (if (eq? reponse 'o)  
 'toto  
 'tata))

N. Guin

Licence Lyon1 - UE LIF3

16

## Sortir du fonctionnel

- Utilisation de la séquence : fonction prédéfinie begin

```
(define exemple  
  (lambda ()  
    (begin  
      (display "Bonjour, souhaitez-vous continuer (o/n)")  
      (newline)  
      (let ((reponse (read)))  
        (if (eq? reponse 'o)  
            (exemple)  
            (display "Au revoir !"))))))
```

N. Guin

Licence Lyon1 - UE LIF3

17

## Utilisation des primitives d'E/S et de la séquence pour le blackjack

- Fonction qui affiche l'état du blackjack :  
afficher\_jeu
- Fonction faire\_jouer\_joueur qui propose au joueur de piocher une ou plusieurs cartes (tant qu'il ne dépasse pas 21) et qui retourne le blackjack mis à jour

N. Guin

Licence Lyon1 - UE LIF3

18

## Fonction pour réaliser un tour de jeu

- la banque tire une carte
- le joueur tire 2 cartes
- le joueur continue à jouer s'il le souhaite
- quand le joueur s'arrête, ses points sont comptabilisés
- si le joueur a dépassé les 21 points, la fonction affiche le fait qu'il a perdu
- sinon, la banque continue à jouer
- à la fin du jeu de la banque, son nombre de points est comptabilisé
- puis la fonction affiche le jeu et un message précisant qui est le gagnant

## Dernière partie du projet

- Réaliser l'analyse de jeu de la bataille
  - Décomposer le problème
  - Quelles fonctions de base peut-on réutiliser ?
  - Peut-on adapter certaines fonctions du blackjack ?
  - Quelles nouvelles fonction faut-il créer ?
- On ne vous demande pas de programmer, mais de faire ce que nous venons de faire pour le blackjack
  - Vous devez donner la spécification des nouvelles fonctions

## Comment rendre le travail (valable aussi pour le TP noté)

- Vous déposer sur TOMUSS le fichier avec votre nom
  - Ex : Projet-ZARA.scm
  - Les première lignes de votre fichier doivent être vos nom, prénom et numéro d'étudiant (en commentaire)
- Dépôt à faire à chaque séance de TP, et lors de l'évaluation (3 mai 2016) – colonnes TOMUSS :
  - TP8\_Rendu – séance du 12 avril
  - TP9\_Rendu – séance du 26 avril
  - TP\_Projet\_Rendu – évaluation du 3 mai

## Comment rendre le travail (valable aussi pour le TP noté)

- Toutes vos fonctions doivent être commentées et testées
  - Type des arguments et du résultat
  - Des noms d'arguments significatifs
  - Ce que fait la fonction (en particulier si c'est une fonction que vous introduisez)
  - Des commentaires sur certaines parties pour comprendre ce qu'elle fait
  - Elle est suivie (en commentaire) des tests effectués et des résultats obtenus
- Ceci compte pour une part importante de l'évaluation

## Exemple

```
; Nathalie Guin
; numéro d'étudiant : 00000007

; fonction qui calcule le nombre d'occurences
; d'un element x dans une liste l
(define mystere ; -> un entier
  (lambda (x l) ; x un element, l une liste
    (cond ((null? l) 0)
          ((eq? x (car l)) (+ 1 (mystere x (cdr l))))
          ; on compte 1 pour l'element
          (else (mystere x (cdr l)))))

; (mystere 'a '(e a z z t)) -> 1
; (mystere 'a '(e a z a t)) -> 2
; (mystere 'a '(e z z t)) -> 0
```

## Pour le projet

- Il vaut mieux ne pas tout faire mais le faire bien
- Vous devez travailler chez vous en dehors des deux séances
- Vous aurez à présenter oralement votre travail à votre enseignant : on note plus votre démarche et ce que vous avez compris que ce qui marche