

# Structures

## Exercise 1

```
package main

import "fmt"

type item struct {
    id    int
    name  string
    price int
}

type game struct {
    item
    genre string
}

func main() {
    games := []game{
        {
            item: item{id: 1, name: "god of war", price: 50},
            genre: "action adventure",
        },
        {
            item: item{id: 2, name: "x-com 2", price: 40},
            genre: "strategy",
        },
        {
            item: item{id: 3, name: "minecraft", price: 20},
            genre: "sandbox",
        },
    }

    fmt.Printf("Le magasin propose %d jeux.\n\n", len(games))

    for _, g := range games {
        fmt.Printf("#%-4d: %-15q %-20s %d€\n",
            g.id,
            g.name,
            ("+"g.genre+"),
            g.price,
        )
    }
}
```

## Exercise 2

```
package main

import (
    "bufio"
```

```

    "fmt"
    "os"
)

type item struct {
    id    int
    name  string
    price int
}

type game struct {
    item
    genre string
}

func main() {
    games := []game{
        {
            item: item{id: 1, name: "god of war", price: 50},
            genre: "action adventure",
        },
        {
            item: item{id: 2, name: "x-com 2", price: 40},
            genre: "strategy",
        },
        {
            item: item{id: 3, name: "minecraft", price: 20},
            genre: "sandbox",
        },
    }

    fmt.Printf("Le magasin propose %d jeux.\n", len(games))

    in := bufio.NewScanner(os.Stdin)
    for {
        fmt.Print(`
Commandes :
> list    : liste tous les jeux
> quit    : quitte

`)
        fmt.Print("Votre choix : ")
        in.Scan()
        fmt.Printf("\n")

        switch in.Text() {
        case "quit":
            fmt.Println("Au revoir !")
            return
        case "list":
            for _, g := range games {
                fmt.Printf("#%-4d: %s %s %d€\n",
                    g.id,
                    g.name,
                    (" "+g.genre+""),
                    g.price,
                )
            }
        }
    }
}

```

```

        )
    }
    default:
        fmt.Println("Commande inconnue.")
    }
}
}
}

```

### Exercise 3

```

package main

import (
    "bufio"
    "fmt"
    "os"
    "strconv"
    "strings"
)

type item struct {
    id    int
    name  string
    price int
}

type game struct {
    item
    genre string
}

func main() {
    games := []game{
        {
            item: item{id: 1, name: "god of war", price: 50},
            genre: "action adventure",
        },
        {
            item: item{id: 2, name: "x-com 2", price: 40},
            genre: "strategy",
        },
        {
            item: item{id: 3, name: "minecraft", price: 20},
            genre: "sandbox",
        },
    }

    // Index les jeux par id
    gamesByID := make(map[int]game)
    for _, g := range games {
        gamesByID[g.id] = g
    }

    fmt.Printf("Le magasin propose %d jeux.\n", len(games))
}

```

```

    in := bufio.NewScanner(os.Stdin)
    for {
        fmt.Print(`
Commandes :
> list    : liste tous les jeux
> id N    : affiche le jeu d'identifiant N
> quit    : quitte

`)

        fmt.Print("Votre choix : ")
        in.Scan()
        fmt.Printf("\n")

        cmd := strings.Fields(in.Text())

        if len(cmd) == 0 {
            // Pas de commande, on continue
            continue
        }

        switch cmd[0] {
        case "quit":
            fmt.Println("Au revoir !")
            return
        case "list":
            for _, g := range games {
                fmt.Printf("#%-4d: %-15q %-20s %d€\n",
                    g.id,
                    g.name,
                    ("+"g.genre+""),
                    g.price,
                )
            }
        case "id":
            if len(cmd) != 2 {
                fmt.Println("ID invalide.")
                continue
            }
            id, err := strconv.Atoi(cmd[1])
            if err != nil {
                fmt.Println("ID invalide.")
                continue
            }

            g, ok := gamesByID[id]
            if !ok {
                fmt.Println("Jeu introuvable.")
                continue
            }
            fmt.Printf("#%-4d: %-15q %-20s %d€\n",
                g.id,
                g.name,
                ("+"g.genre+""),
                g.price,
            )
        }
    }

```

```

        default:
            fmt.Println("Commande inconnue.")
        }
    }
}

```

## Exercice 4

```

package main

import (
    "bufio"
    "fmt"
    "os"
    "strconv"
    "strings"
)

type item struct {
    id    int
    name  string
    price int
}

type game struct {
    item
    genre string
}

func main() {
    games := []game{
        {
            item: item{id: 1, name: "god of war", price: 50},
            genre: "action adventure",
        },
        {
            item: item{id: 2, name: "x-com 2", price: 40},
            genre: "strategy",
        },
        {
            item: item{id: 3, name: "minecraft", price: 20},
            genre: "sandbox",
        },
    }

    // Index les jeux par id
    gamesByID := make(map[int]game)
    for _, g := range games {
        gamesByID[g.id] = g
    }

    fmt.Printf("Le magasin propose %d jeux.\n", len(games))

    in := bufio.NewScanner(os.Stdin)

```

```

    for {
        fmt.Print(`
Commandes :
> list : liste tous les jeux
> id N : affiche le jeu d'identifiant N
> add  : ajoute un jeu
> quit : quitte le programme

`)

        fmt.Print("Votre choix : ")
        in.Scan()
        fmt.Printf("\n")

        cmd := strings.Fields(in.Text())

        if len(cmd) == 0 {
            // Pas de commande, on continue
            continue
        }

        switch cmd[0] {
        case "quit":
            fmt.Println("Au revoir !")
            return
        case "list":
            for _, g := range games {
                fmt.Printf("#%-4d: %-15q %-20s %d€\n",
                    g.id,
                    g.name,
                    "("+g.genre+")",
                    g.price,
                )
            }
        case "id":
            if len(cmd) != 2 {
                fmt.Println("ID invalide.")
                continue
            }
            id, err := strconv.Atoi(cmd[1])
            if err != nil {
                fmt.Println("ID invalide.")
                continue
            }

            g, ok := gamesByID[id]
            if !ok {
                fmt.Println("Jeu introuvable.")
                continue
            }
            fmt.Printf("#%-4d: %-15q %-20s %d€\n",
                g.id,
                g.name,
                "("+g.genre+")",
                g.price,
            )
        case "add":

```

```

correctValue := false
var name, genre string
var price int
var err error

for !correctValue {
    fmt.Print("Nom du jeu : ")
    in.Scan()
    name = strings.TrimSpace(in.Text())
    if name == "" {
        fmt.Println("Nom invalide.")
        continue
    }
    correctValue = true
}

correctValue = false
for !correctValue {
    fmt.Print("Genre du jeu : ")
    in.Scan()
    genre = strings.TrimSpace(in.Text())
    if genre == "" {
        fmt.Println("Genre invalide.")
        continue
    }
    correctValue = true
}

correctValue = false
for !correctValue {
    fmt.Print("Prix du jeu : ")
    in.Scan()
    price, err = strconv.Atoi(in.Text())
    if err != nil {
        fmt.Println("Prix invalide.")
        continue
    }
    correctValue = true
}

newGame := game{
    item: item{
        id:    len(games) + 1,
        name:  name,
        price: price,
    },
    genre: genre,
}

games = append(games, newGame)
gamesByID[newGame.id] = newGame
fmt.Printf("\nNouveau jeu ajouté :\n#%-4d: %-15q %-20s %d€\n",
    newGame.id,
    newGame.name,
    "("+newGame.genre+")",
    newGame.price,

```

```
    )  
    default:  
        fmt.Println("Commande inconnue.")  
    }  
}  
}
```