

# Soft(ware)Ball

## But du jeu

Jouer en équipe pour marquer un maximum de points au cours d'une partie en 6 manches

## Déroulement

Chaque manche consiste à résoudre un problème de déplacement de ballon en utilisant des composants programmables : à chaque composant est associé un programme unique.

Pour vérifier la validité d'une solution, les composants sont interprétés par les membres de l'équipe.

Quand l'équipe est satisfaite de la solution proposée, elle valide la manche et procède au décompte des points. Chaque problème résolu rapporte 10 points auquel on retranche le coût des programmes utilisés.

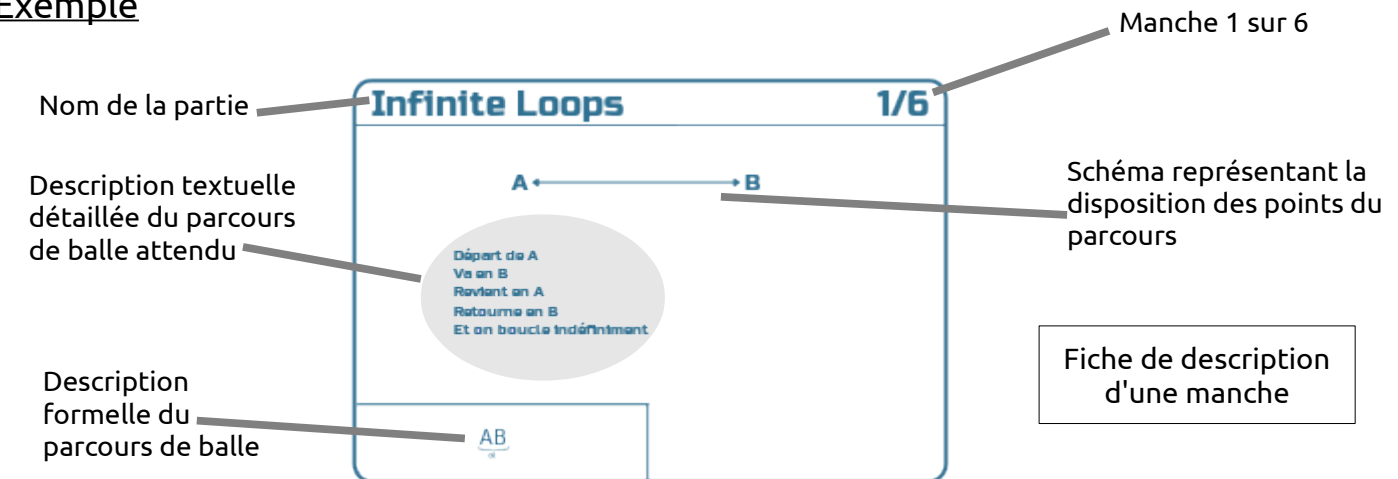
Coût d'un programme = nombre d'événements + nombre d'actions + nombre d'états

Ce coût peut être rapidement évalué en comptant le nombre de verbes.

Les programmes payés lors d'une manche peuvent être réutilisés dans les manches suivantes sans coût additionnel. Il est toutefois impossible d'y apporter la moindre modification.

Tout programme dont l'écriture est terminée peut être dupliqué à coût nul.

## Exemple



### Solution possible

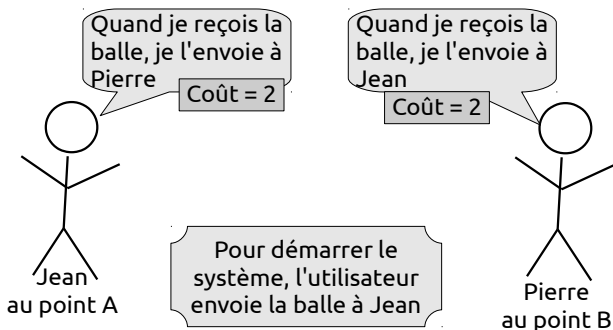


Tableau de score après la 1ère manche

Etape	Points	Verbes	Score	Total
1	10	4	6	6
2	10			
3	10			
4	10			
5	10			
6	10			

### Autre solution possible

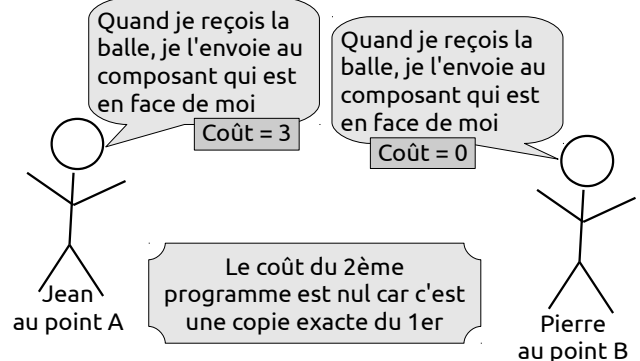


Tableau de score après la 1ère manche

Etape	Points	Verbes	Score	Total
1	10	3	7	7
2	10			
3	10			
4	10			
5	10			
6	10			

## Structure d'un programme

Un programme est un ensemble de règles

```
<programme> ::= <règle> [<programme>]
```

Chaque règle décrit une opération exécutée lorsqu'un événement survient pour un des acteurs

```
<règle> ::= "Quand" <acteur> <événement> ", " <opération>
```

Une opération est une action réalisée par le composant programmé. Plusieurs actions peuvent être enchaînées ou soumises à condition

```
<opération> ::=  
    "moi" <action>  
    | <opération> [ "puis" <opération> ]  
    | "si" <condition> "alors" <opération> [ "sinon" <opération> ]
```

Un acteur du jeu est un composant qui peut être identifié directement ou en fonction d'un état

```
<acteur> ::= "moi" | "lui" | <nom> | "le composant qui" <état> | "un composant"
```

Les noms des composants sont définis a priori et ne peuvent pas changer au cours du jeu

```
<nom> ::= "Pierre" | "Jean" | ...
```

Une condition est le fait d'observer un acteur dans un état donné. Les conditions peuvent être regroupées par des opérateurs logiques.

```
<condition> ::=  
    <acteur> <état>  
    | <condition> "et" <condition> | <condition> "ou" <condition>
```

Une action de jeu est un effort physique et non ambigu effectué par un composant. La liste des actions est présentée à titre d'exemples. Elle peut être agrandie après autorisation de l'arbitre. On notera toutefois que les composants ne peuvent ni lire, ni écrire, ni calculer, ni parler.

```
<action> ::=  
    envoyer la balle à <acteur> | aller au point <position>  
    | tourner <acteur> vers <acteur> | retourner <acteur>  
    | tourner <acteur> à gauche | tourner <acteur> à droite | ...
```

Une position est matérialisée par un point sur le sol

```
<position> ::= "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | ...
```

Un événement est l'observation de quelque chose qui se produit dans l'espace de jeu. Cela inclut l'ensemble des actions possibles.

```
<événement> ::=  
    envoyer la balle | entendre <mot> | <action>  
    | recevoir la balle | recevoir la balle de <acteur>
```

Un mot est un son prononcé par l'utilisateur (car les composants ne parlent pas).

Un état est une situation stable dans laquelle se trouve un acteur, La liste des états est présentée à titre d'exemples. Elle peut être agrandie après autorisation de l'arbitre.

```
<état> ::=  
    être au point <position> | être à côté de <acteur>  
    | être à gauche de <acteur> | être à droite de <acteur> | ...
```

Quand un programme est constitué, il peut être transformé en français correct (pronoms personnels sujets, adjectifs possessifs, verbes conjugués, contractions...) sans que cela n'impacte son coût.