



Jeu de l'équipage

IUT de Bayonne-Département informatique



DUMAI ETIENNE | TD₂/TP₃
MARQUES DA SILVA THOMAS | TD₂/TP₃
Implémentation d'un besoin client
Année universitaire 2023-2024

Table des matières

1. Préambule	2
Principe du jeu :	2
Déroulement d'une manche :	2
1. Algorithme :	4
2. Dictionnaire des variables :	5
3. Initialiser la partie :	7
4. Modifier les paramètres :	9
5. Réponse modifier les paramètres :	10
6. Jouer la partie :	11
7. Jouer la Manche se décompose en 5 étapes :	13
8. Déterminer le joueur :	16
9. Jouer les lancers :	17
10. Tirer 5 nombres aléatoires de 1 à 6 :	20
11. Switch du dé 1 à 6 :	22
Switch 1	22
Switch 2	23
Switch 3	24
Switch 4	25
Switch 5	26
Switch 6	27
12. Ajout des points du lancer courant au score du joueur :	28
13. Fin de chaque lancer :	29
14. Verif arrêt :	30
15. Finaliser la partie :	31
16. Code source :	32

1. PREAMBULE

Principe du jeu :

- Il s'agit d'un jeu de lancer de dés, à 5 dés. La version du jeu à programmer se joue à 2 joueurs, un humain contre la machine.
- Une partie compte 5 manches, chaque manche compte 3 lancers de dés. A chaque lancer, on lance les 5 dés en même temps.
- Le but du jeu est d'obtenir, à la fin de la partie, le maximum de points. Les points sont obtenus lors des lancers des dés. Le total des points d'un joueur est la somme des points cumulés au cours des 5 manches.

Déroulement d'une manche :

- Un des joueurs commence (par exemple, le joueur humain), il dispose de 3 lancers de dés.
- Lors d'une manche, le joueur ne peut commencer à cumuler des points qu'après avoir réuni, et dans cet ordre : un bateau, un capitaine, un équipage.
- Une fois le bateau, le capitaine et l'équipage réunis, les points fournis par les dés peuvent alimenter le score du joueur.

Règle de la partie :

- On ne peut pas avoir de capitaine sans avoir un bateau
- On ne peut pas avoir d'équipage sans capitaine
- On ne peut pas cumuler de points sans avoir d'équipage
- Le bateau est représenté par la valeur 6 du dé, le capitaine par la valeur 5, l'équipage par la valeur 4.
- La fin de la partie est atteinte :
 - Soit par abandon du joueur humain
 - Soit à la fin du nombre de manches fixé Si le joueur humain abandonne, c'est l'autre joueur (ici, la machine) qui gagne. Sinon, le gagnant est le joueur qui a cumulé le plus de points au cours des 5 manches.

Interface du jeu attendu :

Entrez un pseudo: user

Bilan partiel AVANT la nouvelle manche :

Score user = 0

Score Machine = 0

manche #1, joueur user

Lancer #1

Contenu du Lancer : 4 3 2 4 3

Bilan du Lancer :

- Équipement = NON_ÉQUIPE
- Score du Lancer = 0
- Nouveau score pour user = 0

Souhaitez-vous continuer ? (O/N)

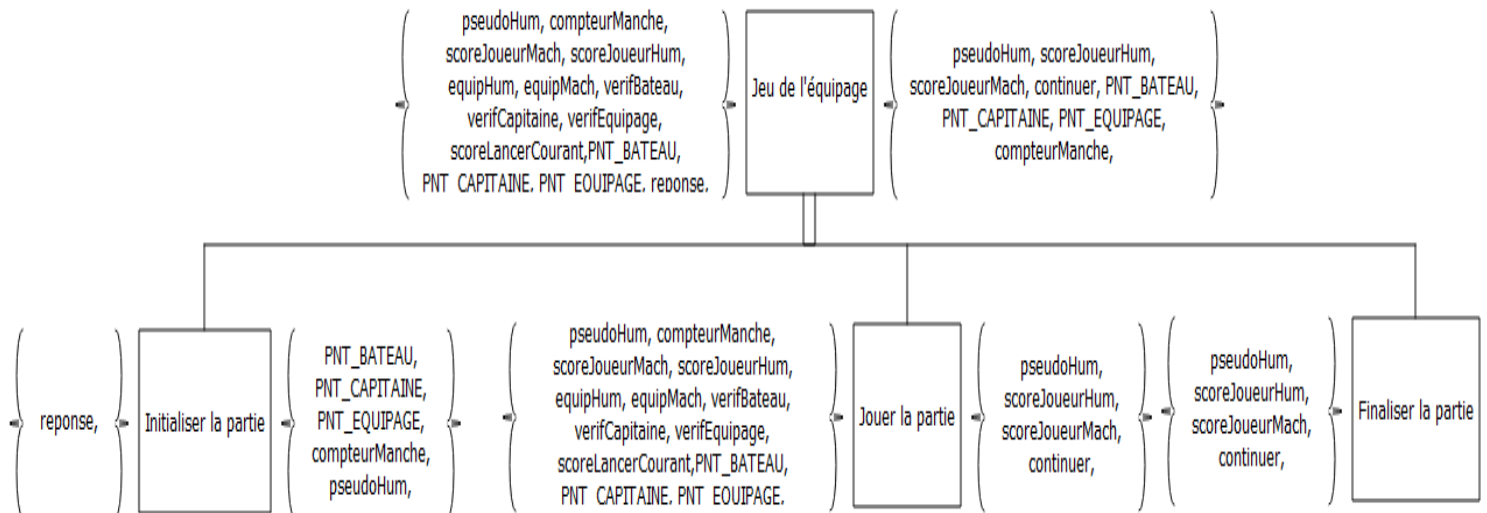
Précision :

Une fois le joueur équipé (bateau + capitaine + équipage), chaque face du dé apporte au score du joueur le nombre de points indiqués par le dé :

- La face '1' du dé apporte 1 point au score du joueur
- La face '2' du dé apporte 2 points au score du joueur
- La face '3' du dé apporte 3 points au score du joueur
- La face '4' du dé apporte 4 points au score du joueur
- La face '5' du dé apporte 5 points au score du joueur
- La face '6' du dé apporte 6 points au score du joueur

1. ALGORITHME :

L'algorithme « du jeu de l'équipage » s'appuie sur un dictionnaire (un fichier texte) contenant la liste des mots du dictionnaire français. Ce dictionnaire est utilisé pour vérifier que les mots proposés par le joueur existent.



L'algorithme principal du jeu se décompose en 3 sous-problèmes principaux :

- Initialiser la partie :
 - Dans cette étape, il s'agit d'initialiser les éléments permettant de commencer la partie.
 - Dans notre cas, il s'agit de définir les points du Capitaine, Bateau, Equipage par défaut. Il s'agit d'initialiser le compteurManche à 0, de modifier les paramètres des points du Capitaine, Bateau, Equipage si on le souhaite et d'afficher le pseudo que l'on souhaite à l'aide d'une saisie.
- Jouer la partie :
 - Dans cette étape, il s'agit de définir le joueur qui va commencer la manche et de jouer la manche.
- Finaliser la partie :
 - Dans cette dernière étape, il s'agit de demander à l'utilisateur s'il veut continuer la partie et d'afficher le score en fonction de sa réponse.

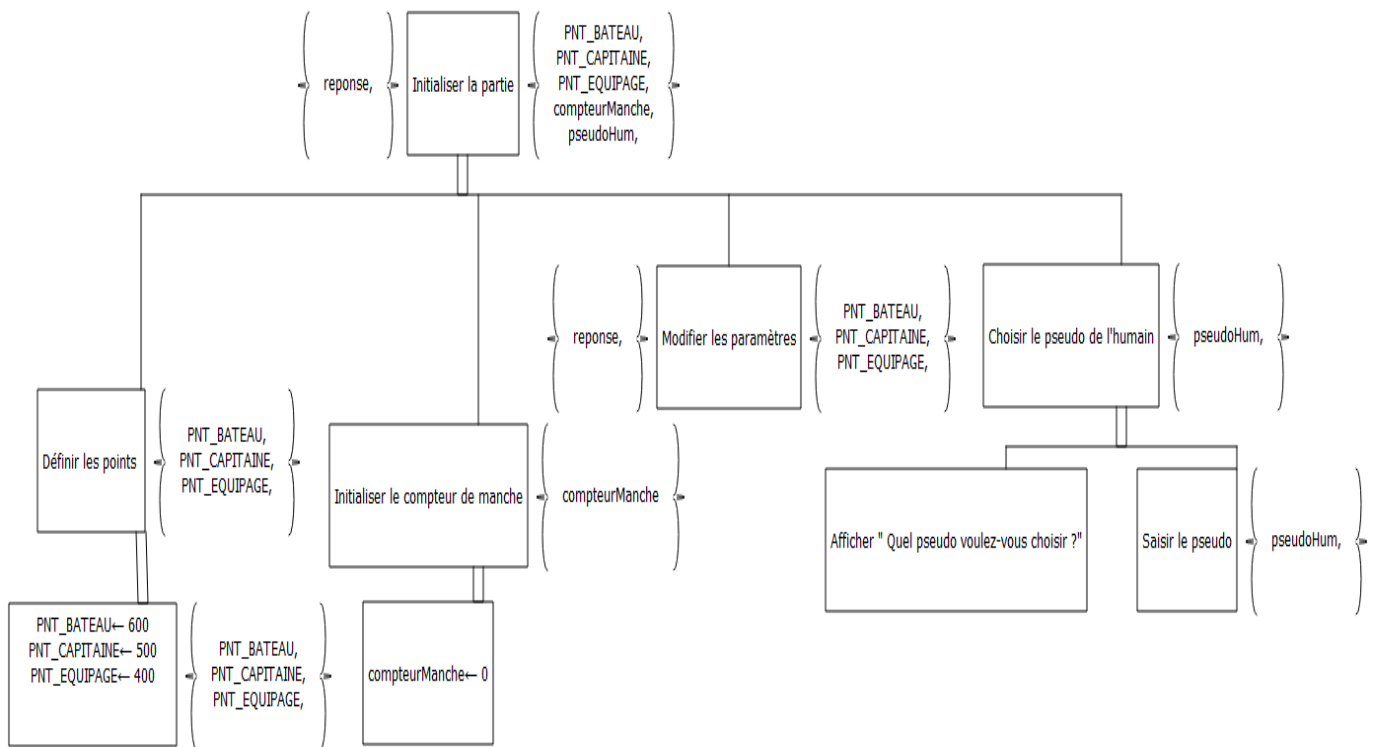
2. DICTIONNAIRE DES VARIABLES :

Nom	Type	Signification	Mode d'initialisation	Exportation
compteurManche	entier	Variable qui compte les manches au fur et à mesure de la partie	Prédéfinie à 0	Non
compteurLance	entier	Variable qui compte les lancers au fur et à mesure de la partie	Prédéfini a 0	Non
PTN_BATEAU	entier	Variable qui contient le nombre de point qu'un bateau vaut dans la partie, les points sont donnés lorsqu'un « 6 » est tiré	Prédéfini a 600 mais peut être modifiée	Non
PNT_CAPITAINE	entier	Variable qui contient le nombre de point qu'un capitaine vaut dans la partie, les points sont donnés lorsqu'un « 5 » est tiré	Prédéfini a 500 mais peut être modifiée	Non
PTN_EQUIPAGE	entier	Variable qui contient le nombre de point qu'un capitaine vaut dans la partie, les points sont donnés lorsqu'un « 4 » est tiré	Prédéfini a 400 mais peut être modifiée	Non
pseudoHum	chaîne de caractères	Variable qui contient le pseudo choisi par la personne qui joue contre la machine	Saisie	Non
equipHum	chaîne de caractères	Variable qui contient l'équipement du joueur humain tout au long de la partie	Calculé mais prédéfini a NON_EQUIPE	Non
scoreLancerCourant	entier	Score de chaque lancer de l'humain ou de la machine	Calculé	Non
scoreJoueurHum	entier	Score de l'humain tout au long de la partie	Calculé	Non
scoreJoueurMachine	entier	Score de la machine tout au long de la partie	Calculé	Non

equipMach	chaîne de caractères	Variable qui contient l'équipement de la machine tout au long de la partie	Calculé mais prédéfini a NON_EQUIPE	Non
verifBateau	bool	Vérifie si le bateau a déjà été tiré	Prédéfini sur faux	Non
verifCapitaine	bool	Vérifie si le capitaine a déjà été tiré	Prédéfini sur faux	Non
verifEquipage	bool	Vérifie si l'équipage a déjà été tiré	Prédéfini sur faux	Non
continuer	bool	Savoir si le joueur veut continuer à jouer	Prédéfini sur vrai	Non
reponse	caractère	Réponse du joueur pour savoir s'il veut continuer à jouer	Saisie	non
joueurActu	chaîne de caractère	Définit le joueur qui doit jouer	Prédéfini	Non
deTirer	chaîne de caractère	Dés tirés par le joueur au cours d'un lancer	Prédéfini	Non
nombreAlea	entier	Nombre aléatoire tiré au hasard	Prédéfini	Non
de	entier	Dé correspondant au nombre aléatoire tiré	Prédéfini	non

Nous avons décidé d'initialiser toutes nos variables en début d'algorithme c'est pour cela que nous vous avons affiché tout le dictionnaire des variables

3. INITIALISER LA PARTIE :



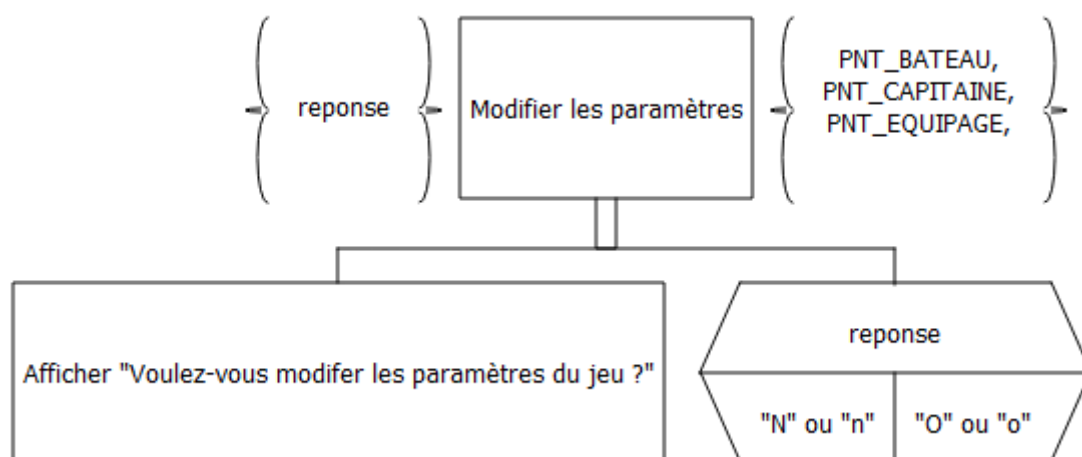
L'initialisation de la partie se décompose en 4 étapes :

- Il s'agit d'initialiser les éléments permettant de commencer la partie.
- Dans notre cas, il s'agit de définir les points du Capitaine, Bateau, Equipage par défaut.
- Initialiser le compteurManche à 0
- Modifier les paramètres des points du Capitaine, Bateau, Equipage si on le souhaite et d'afficher le pseudo que l'on souhaite à l'aide d'une saisie.

Nom	Type	Signification	Mode d'initialisation	Exportation
compteurManche	entier	Variable qui compte les manches au fur et à mesure de la partie	Prédéfinie à 0	Non
PTN_BATEAU	entier	Variable qui contient le nombre de point qu'un bateau vaut dans la partie, les points sont donnés lorsque un « 6 » est tiré	Prédéfini a 600 mais peut être modifiée	Non
PNT_CAPITAINE	entier	Variable qui contient le nombre de point qu'un capitaine vaut dans la partie, les points sont donnés lorsque un « 5 » est tiré	Prédéfini a 500 mais peut être modifiée	Non
PTN_EQUIPAGE	entier	Variable qui contient le nombre de point qu'un capitaine vaut dans la partie, les points sont donnés lorsque un « 4 » est tiré	Prédéfini a 400 mais peut être modifiée	Non
pseudoHumain	chaîne de caractères	Variable qui contient le pseudo choisi par la personne qui joue contre la machine	Saisie	Non
reponse	Caractère	Réponse du joueur pour savoir s'il veut continuer à jouer	Saisie	non

4. MODIFIER LES PARAMETRES :

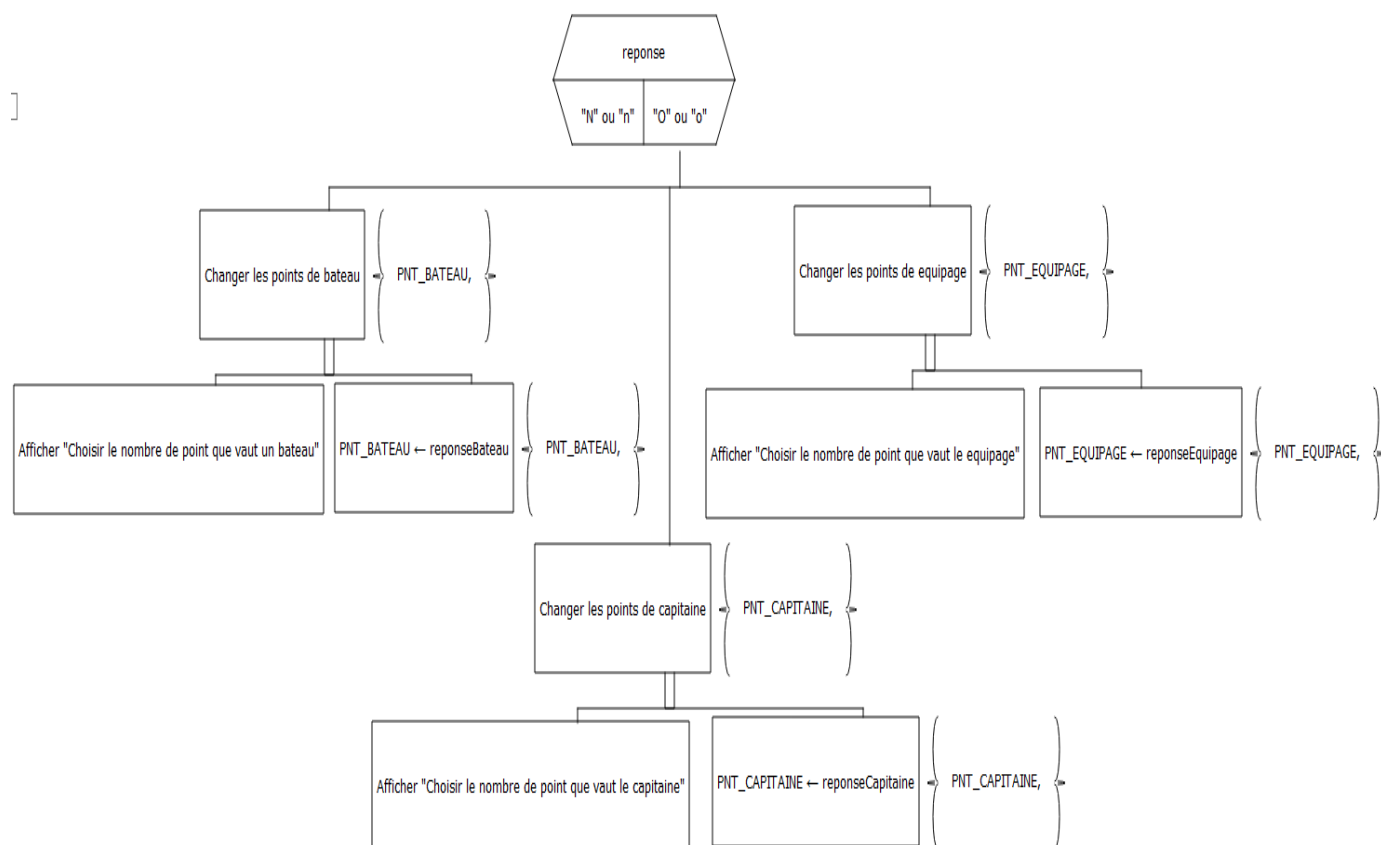
Cette étape va permettre de demander à l'utilisateur s'il veut changer les paramètres de la partie.



PTN_BATEAU	entier	Variable qui contient le nombre de point qu'un bateau vaut dans la partie, les points sont donnés lorsque un « 6 » est tiré	Prédéfini a 600 mais peut être modifiée	Non
PNT_CAPITAINE	entier	Variable qui contient le nombre de point qu'un capitaine vaut dans la partie, les points sont donnés lorsque un « 5 » est tiré	Prédéfini a 500 mais peut être modifiée	Non
PTN_EQUIPAGE	entier	Variable qui contient le nombre de point qu'un capitaine vaut dans la partie, les points sont donnés lorsque un « 4 » est tiré	Prédéfini a 400 mais peut être modifiée	Non
reponse	Caractère	Réponse du joueur pour savoir s'il veut continuer à jouer	Saisie	non

5. REPONSE MODIFIER LES PARAMETRES :

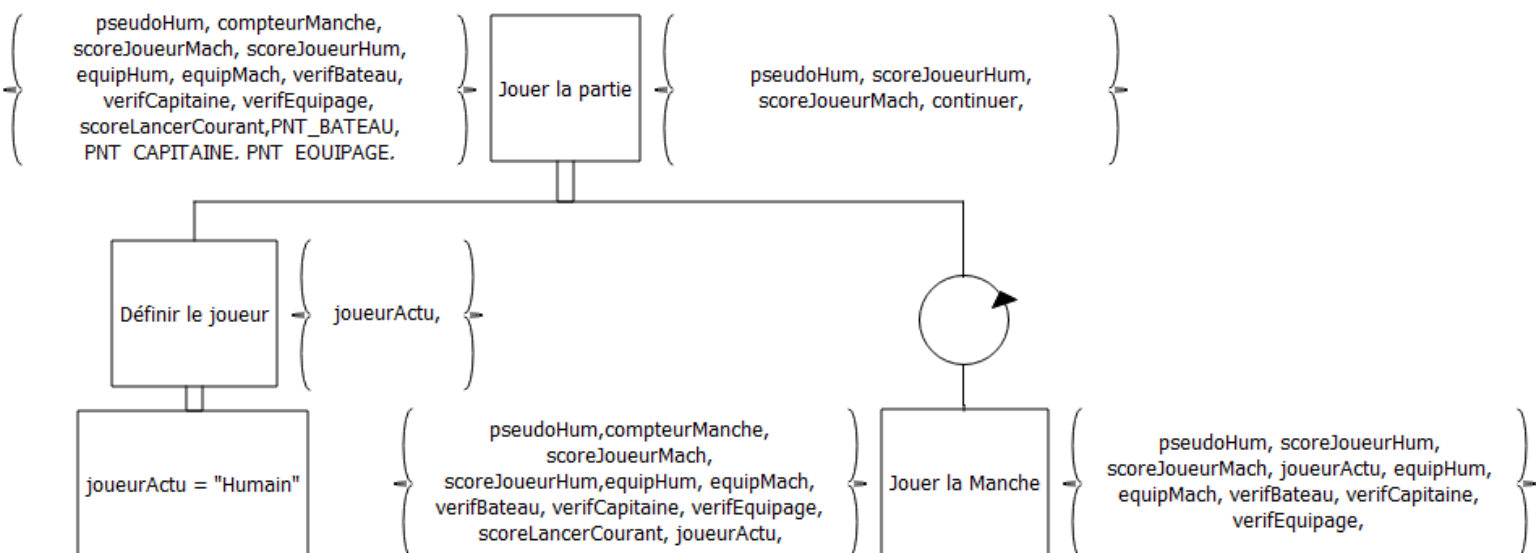
Cette étape va permettre de changer les paramètres si l'utilisateur le veut.



PTN_BATEAU	entier	Variable qui contient le nombre de point qu'un bateau vaut dans la partie, les points sont donnés lorsque un « 6 » est tiré	Prédéfini a 600 mais peut être modifiée	Non
PNT_CAPITAINE	entier	Variable qui contient le nombre de point qu'un capitaine vaut dans la partie, les points sont donnés lorsque un « 5 » est tiré	Prédéfini a 500 mais peut être modifiée	Non
PTN_EQUIPAGE	entier	Variable qui contient le nombre de point qu'un capitaine vaut dans la partie, les points sont donnés lorsque un « 4 » est tiré	Prédéfini a 400 mais peut être modifiée	Non

6. JOUER LA PARTIE :

Le sous-problème Jouer la partie consiste à répéter 2 étapes marquant le déroulement d'une manche.



Les 2 étapes consistent à :

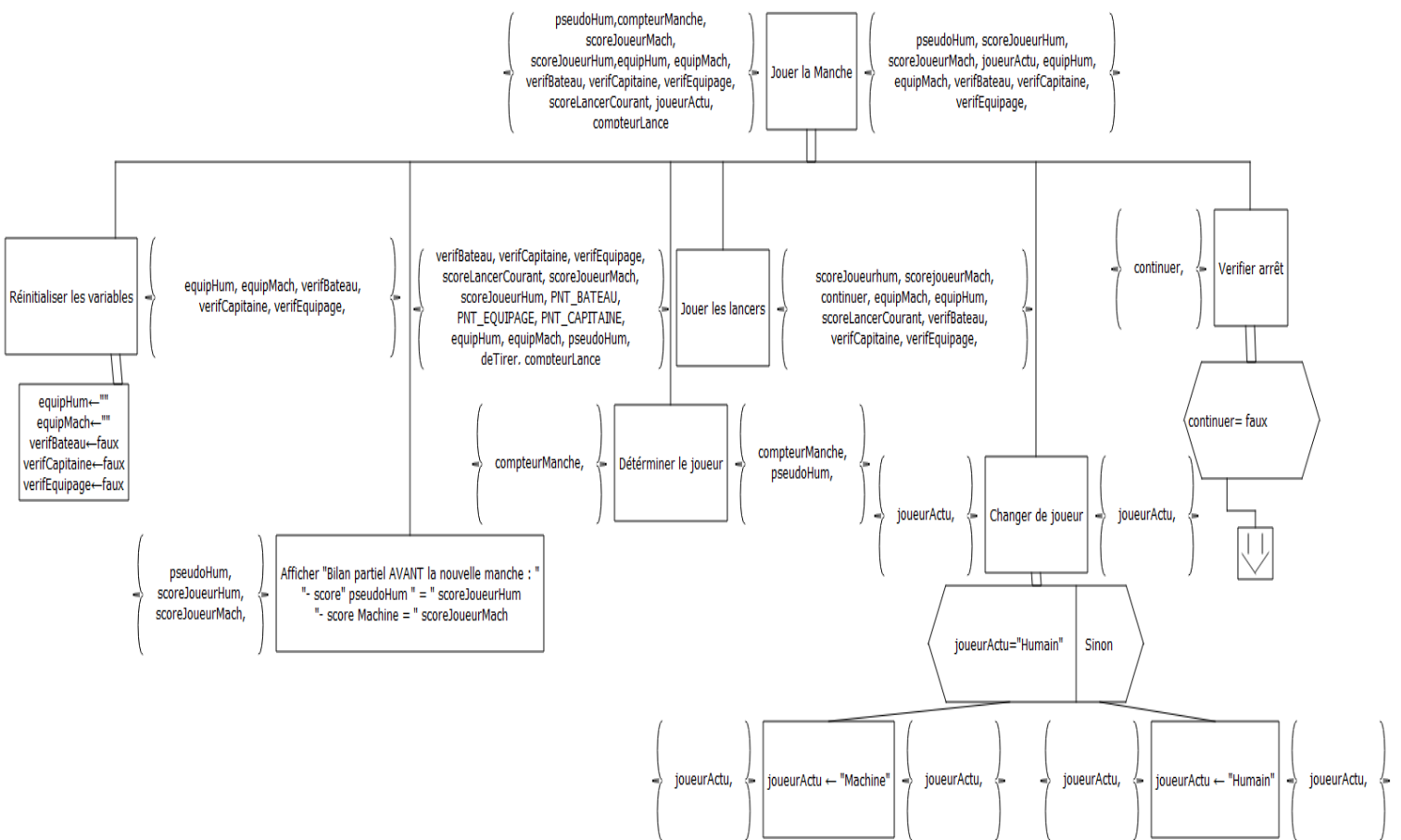
- Définir le joueur qui va débiter la manche
- Jouer la Manche

Nom	Type	Signification	Mode d'initialisation	Exportation
compteurManche	entier	Variable qui compte les manches au fur et à mesure de la partie	Prédéfinie à 0	Non
compteurLance	entier	Variable qui compte les lancés au fur et à mesure de la partie	Prédéfini a 0	Non
PTN_BATEAU	entier	Variable qui contient le nombre de point qu'un bateau vaut dans la partie, les points sont donnés lorsque un « 6 » est tiré	Prédéfini a 600 mais peut être modifiée	Non
PNT_CAPITAINE	entier	Variable qui contient le nombre de point qu'un capitaine vaut dans la partie, les points sont donnés lorsque un « 5 » est tiré	Prédéfini a 500 mais peut être modifiée	Non

PTN_EQUIPAGE	entier	Variable qui contient le nombre de point qu'un capitaine vaut dans la partie, les points sont donnés lorsque un « 4 » est tiré	Prédéfini a 400 mais peut être modifiée	Non
pseudoHum	chaîne de caractères	Variable qui contient le pseudo choisi par la personne qui joue contre la machine	Saisie	Non
equipHum	chaîne de caractères	Variable qui contient l'équipement du joueur humain tout au long de la partie	Calculé mais prédéfini a NON_EQUIPE	Non
scoreLancerCoulant	entier	Score de chaque lancer de l'humain ou de la machine	Calculé	Non
scoreJoueurHum	entier	Score de l'humain tout au long de la partie	Calculé	Non
scoreJoueurMachine	entier	Score de la machine tout au long de la partie	Calculé	Non
equipMachine	chaîne de caractères	Variable qui contient l'équipement de la machine tout au long de la partie	Calculé mais prédéfini a NON_EQUIPE	Non
verifBateau	bool	Vérifie si le bateau a déjà été tiré	Prédéfini sur faux	Non
verifCapitaine	bool	Vérifie si le capitaine a déjà été tiré	Prédéfini sur faux	Non
verifEquipage	bool	Vérifie si l'équipage a déjà été tiré	Prédéfini sur faux	Non
continuer	bool	Savoir si le joueur veut continuer à jouer	Prédéfini sur vrai	Non
joueurActuel	chaîne de caractère	Définit le joueur qui doit jouer	Prédéfini	non

7. JOUER LA MANCHE SE DECOMPOSE EN 5 ETAPES :

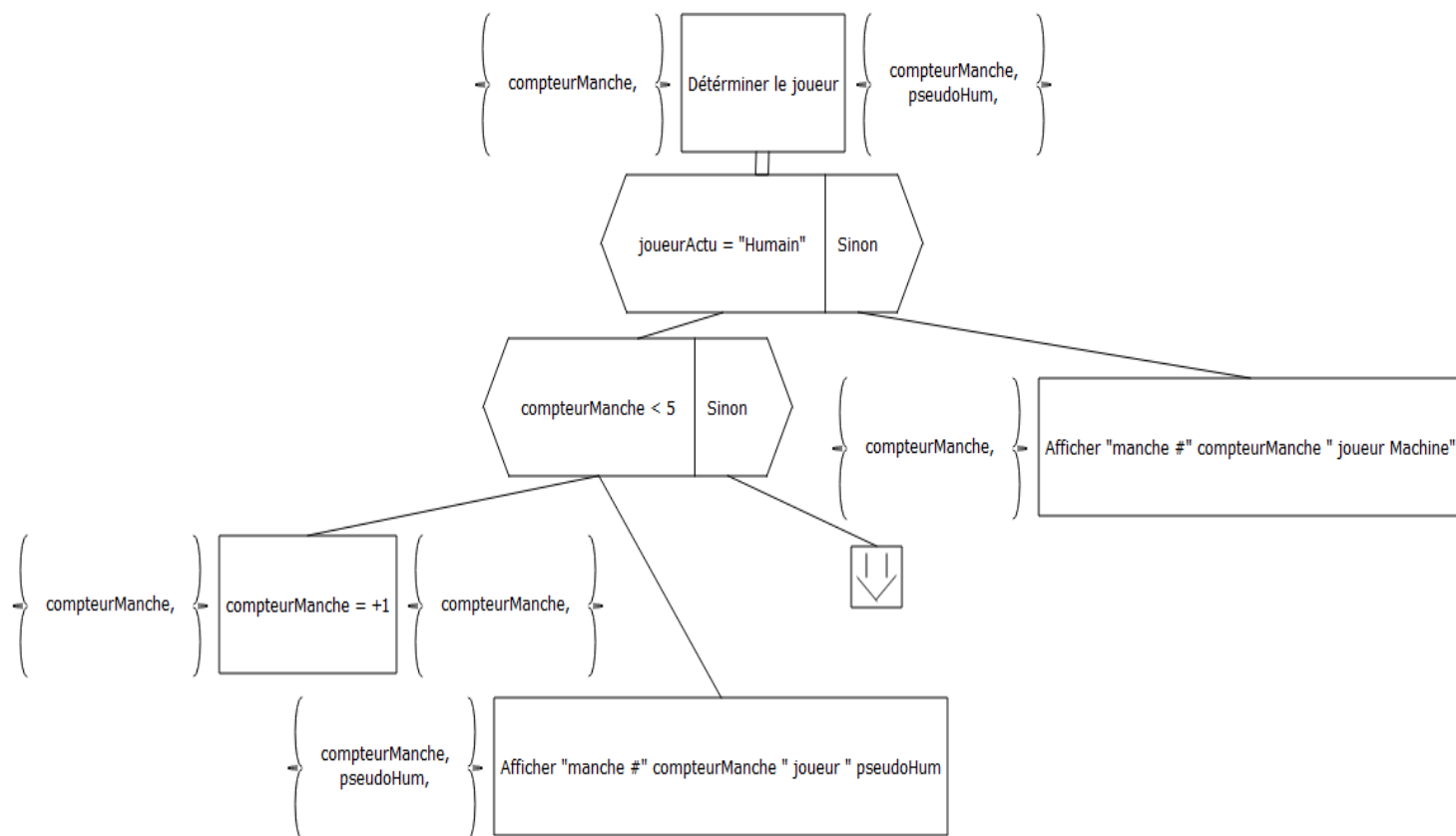
- Réinitialiser les variables
- Afficher "Bilan partiel AVANT la nouvelle manche : "
- Déterminer le joueur
- Jouer des Lancers
- Vérifier si le joueur continue la partie
- Changer de Joueur



pseudoHum	chaîne de caractères	Variable qui contient le pseudo choisi par la personne qui joue contre la machine	Saisie	Non
equipHum	chaîne de caractères	Variable qui contient l'équipement du joueur humain tout au long de la partie	Calculé mais prédéfini à NON_EQUIPE	Non
scoreLancer Courant	entier	Score de chaque lancer de l'humain ou de la machine	Calculé	Non
scoreJoueur Hum	entier	Score de l'humain tout au long de la partie	Calculé	Non
scoreJoueur Mach	entier	Score de la machine tout au long de la partie	Calculé	Non
equipMach	chaîne de caractères	Variable qui contient l'équipement de la machine tout au long de la partie	Calculé mais prédéfini à NON_EQUIPE	Non
verifBateau	bool	Vérifie si le bateau a déjà été tiré	Prédéfini sur faux	Non
verifCapitaine	bool	Vérifie si le capitaine a déjà été tiré	Prédéfini sur faux	Non
verifEquipage	bool	Vérifie si l'équipage a déjà été tiré	Prédéfini sur faux	Non
PTN_BATEAU	entier	Variable qui contient le nombre de point qu'un bateau vaut dans la partie, les points sont donnés lorsque un « 6 » est tiré	Prédéfini à 600 mais peut être modifiée	Non
PNT_CAPITAINE	entier	Variable qui contient le nombre de point qu'un capitaine vaut dans la partie,	Prédéfini à 500 mais peut être modifiée	Non

		les points sont donnés lorsque un « 5 » est tiré		
PTN_EQUIPAGE	entier	Variable qui contient le nombre de point qu'un capitaine vaut dans la partie, les points sont donnés lorsque un « 4 » est tiré	Prédéfini a 400 mais peut être modifiée	Non
compteurManche	entier	Variable qui compte les manches au fur et à mesure de la partie	Prédéfinie à 0	Non
compteurLance	entier	Variable qui compte les lancers au fur et à mesure de la partie	Prédéfini a 0	Non
joueurActuel	chaîne de caractère	Définit le joueur qui doit jouer	Prédéfini	non
continuer	bool	Savoir si le joueur veut continuer à jouer	Prédéfini sur vrai	Non

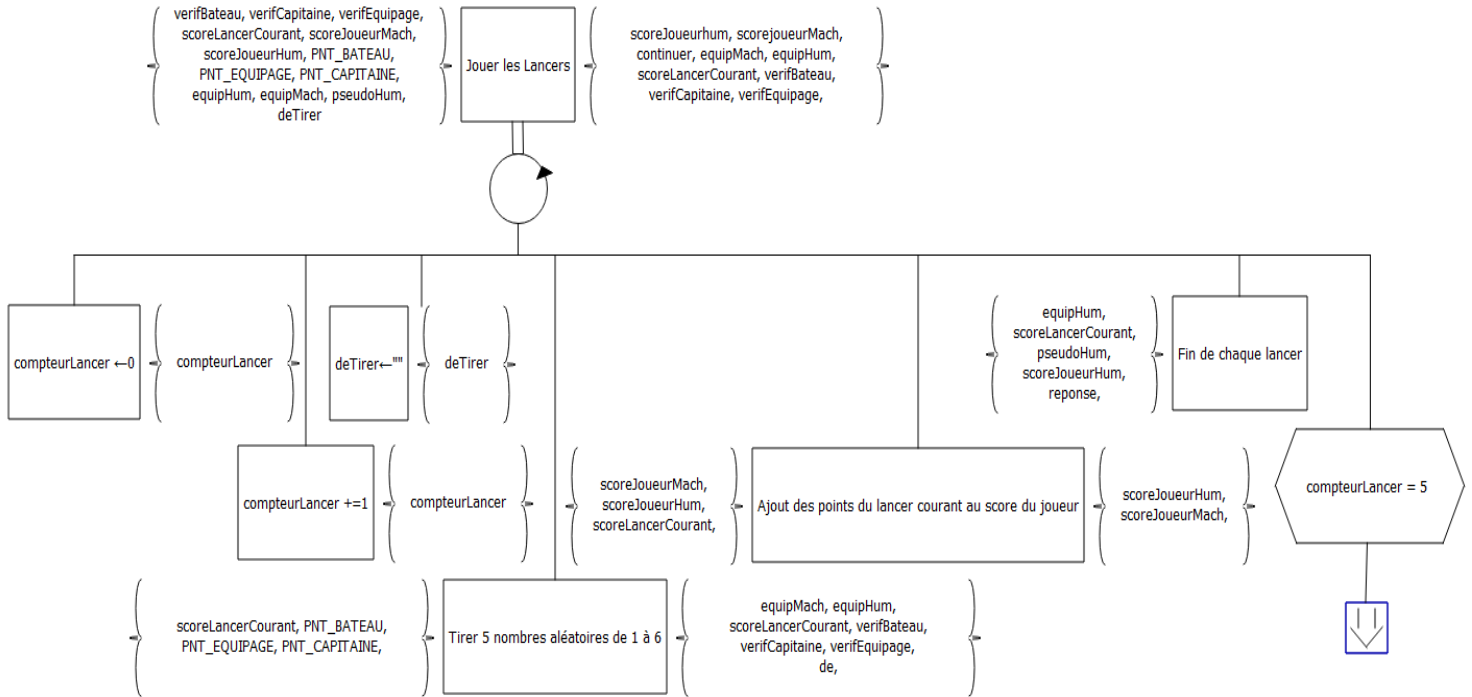
8. DETERMINER LE JOUEUR :



Cette étape va permettre de déterminer le joueur qui va jouer la manche

Nom	Type	Signification	Mode d'initialisation	Exportation
compteurManche	entier	Variable qui compte les manches au fur et à mesure de la partie	Prédéfinie à 0	Non
pseudoHum	chaîne de caractères	Variable qui contient le pseudo choisi par la personne qui joue contre la machine	Saisie	Non

9. JOUER LES LANCERS :



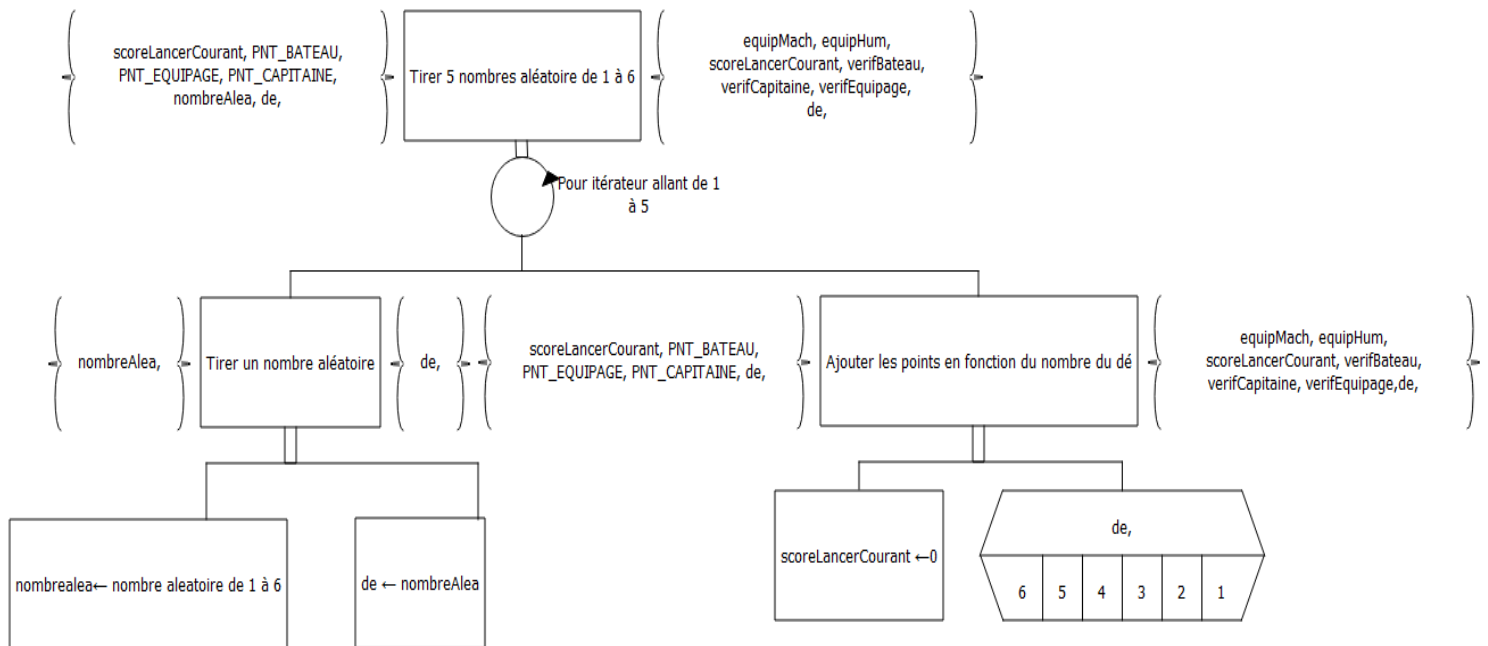
Jouer les lancers va se décomposer en 7 étapes :

- Initialiser le compteurLancer à 0
- Ajouter 1 à chaque fois qu'on lance le dé
- Mettre à vide la chaine de caractère deTirer
- Tirer 5 nombres aléatoires de 1 à 6.
- Ajouter des points du lancer courant au score du joueur
- Mettre fin à chaque lancer
- Quand le compteurLance = 5 on sort de la boucle

equipHum	chaîne de caractères	Variable qui contient l'équipement du joueur humain tout au long de la partie	Calculé mais prédéfini a NON_EQUIPE	Non
scoreLancerCourant	entier	Score de chaque lancer de l'humain ou de la machine	Calculé	Non
scoreJoueurHum	entier	Score de l'humain tout au long de la partie	Calculé	Non
scoreJoueurMach	entier	Score de la machine tout au long de la partie	Calculé	Non
equipMach	chaîne de caractères	Variable qui contient l'équipement de la machine tout au long de la partie	Calculé mais prédéfini a NON_EQUIPE	Non
verifBateau	bool	Vérifie si le bateau a déjà été tiré	Prédéfini sur faux	Non
verifCapitaine	bool	Vérifie si le capitaine a déjà été tiré	Prédéfini sur faux	Non
verifEquipage	bool	Vérifie si l'équipage a déjà été tiré	Prédéfini sur faux	Non
PTN_BATEAU	entier	Variable qui contient le nombre de point qu'un bateau vaut dans la partie, les points sont donnés lorsque un « 6 » est tiré	Prédéfini a 600 mais peut être modifiée	Non
PNT_CAPITAINE	entier	Variable qui contient le nombre de point qu'un capitaine vaut dans la partie, les points sont donnés lorsque un « 5 » est tiré	Prédéfini a 500 mais peut être modifiée	Non
PTN_EQUIPAGE	entier	Variable qui contient le nombre de point qu'un équipage vaut dans la partie, les points sont donnés lorsque un « 4 » est tiré	Prédéfini a 400 mais peut être modifiée	Non

deTirer	Chaine de caractere	Dés tirés par le joueur au cours d'un lancer	Prédéfini	Non
pseudoHum	Chaine de caractères	Variable qui contient le pseudo choisi par la personne qui joue contre la machine	Saisie	Non

10. TIRER 5 NOMBRES ALEATOIRES DE 1 A 6 :



Cette boucle va se décomposer en 2 étapes :

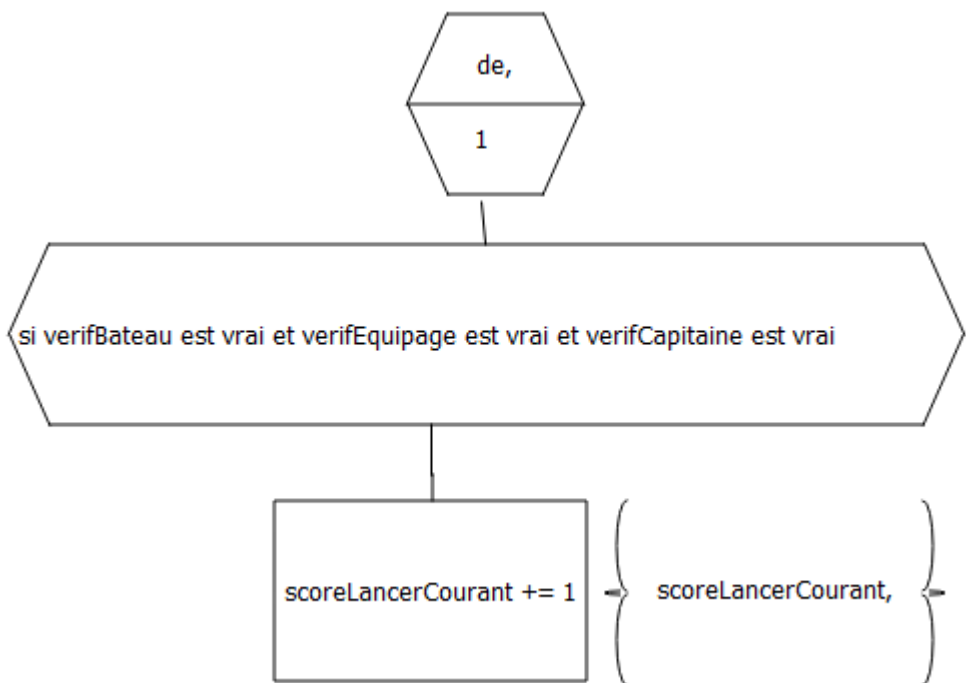
- Tirer un nombre aléatoire de 1 à 6
- Ajouter les points en fonction du dé tiré

nombreAlea	entier	Nombre aleatoire tirer au hasard	Prédéfini	Non
de	entier	Dé correspondant au nombre aleatoire tiré	Prédéfini	Non
equipMach	chaîne de caractères	Variable qui contient l'équipement de la machine tout au long de la partie	Calculé mais prédéfini a NON_EQUIPE	Non
verifBateau	bool	Vérifie si le bateau a déjà été tiré	Prédéfini sur faux	Non
verifCapitaine	bool	Vérifie si le capitaine a déjà été tiré	Prédéfini sur faux	Non
verifEquipage	bool	Vérifie si l'équipage a déjà été tiré	Prédéfini sur faux	Non
PTN_BATEAU	entier	Variable qui contient le nombre de point qu'un bateau vaut dans la partie, les points sont donnés lorsqu'un « 6 » est tiré	Prédéfini a 600 mais peut être modifiée	Non
PNT_CAPITAINE	entier	Variable qui contient le nombre de point qu'un capitaine vaut dans la partie, les points sont donnés lorsqu'un « 5 » est tiré	Prédéfini a 500 mais peut être modifiée	Non
PTN_EQUIPAGE	entier	Variable qui contient le nombre de point qu'un capitaine vaut dans la partie, les points sont donnés lorsqu'un « 4 » est tiré	Prédéfini a 400 mais peut être modifiée	Non
scoreLancerCourant	entier	Score de chaque lancer de l'humain ou de la machine	Calculé	Non
equipHum	chaîne de caractères	Variable qui contient l'équipement du joueur humain tout au long de la partie	Calculé mais prédéfini a NON_EQUIPE	Non

11. SWITCH DU DE 1 A 6 :

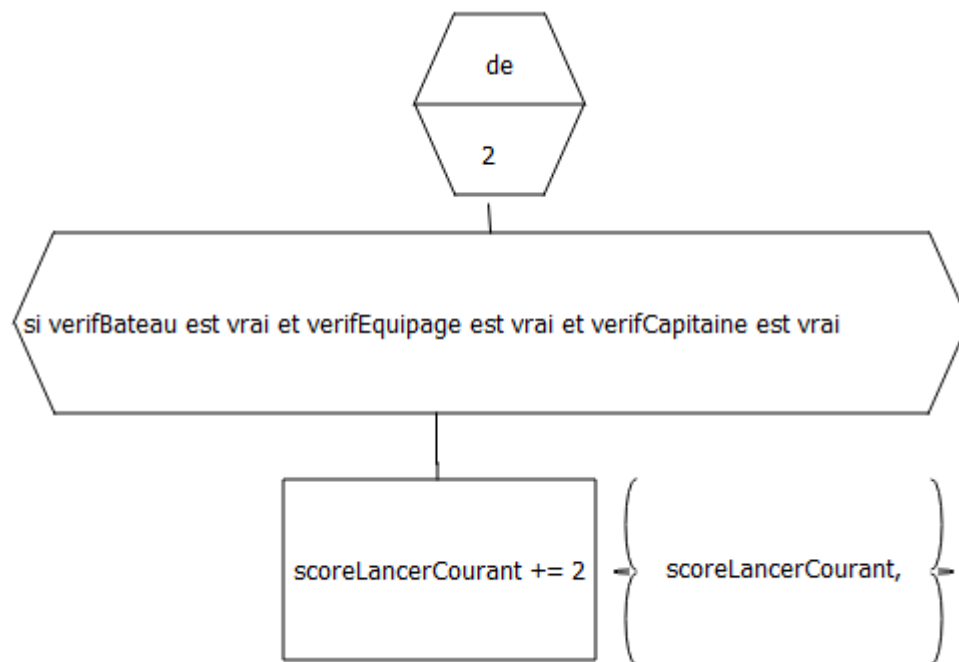
Cette étape va permettre d’attribuer les valeurs des dés au point qui leurs sont attribués.

Switch 1



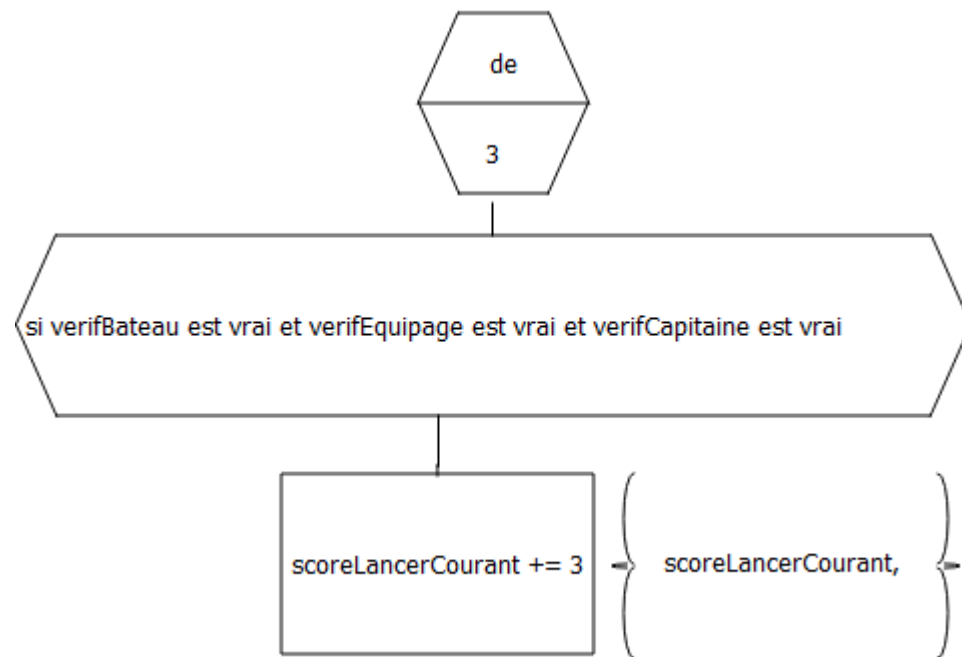
scoreLancerCourant	entier	Score de chaque lancer de l'humain ou de la machine	Calculé	Non
--------------------	--------	---	---------	-----

Switch 2



scoreLancerCourant	entier	Score de chaque lancer de l'humain ou de la machine	Calculé	Non
--------------------	--------	---	---------	-----

Switch 3

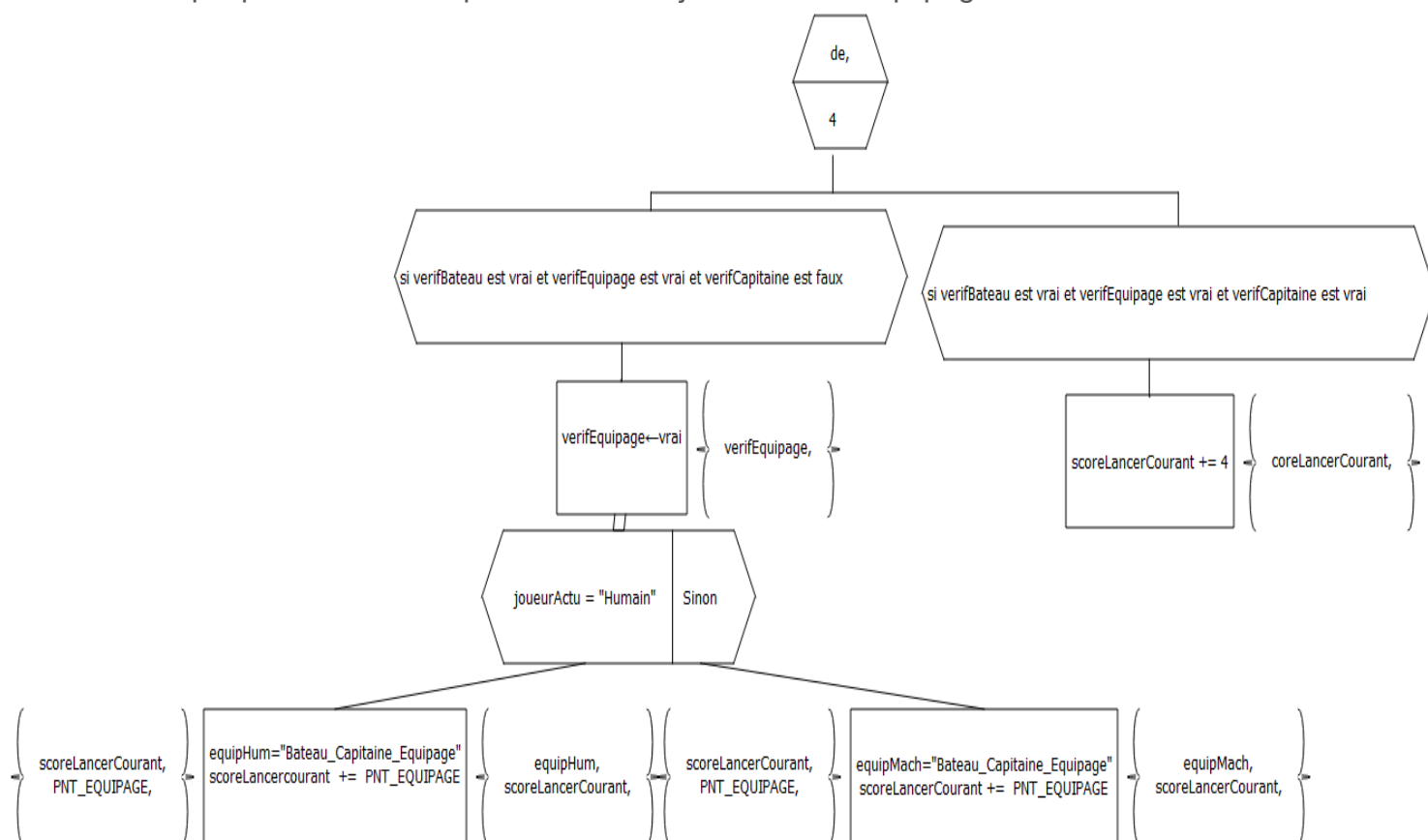


scoreLancerCourant	entier	Score de chaque lancer de l'humain ou de la machine	Calculé	Non

Switch 4

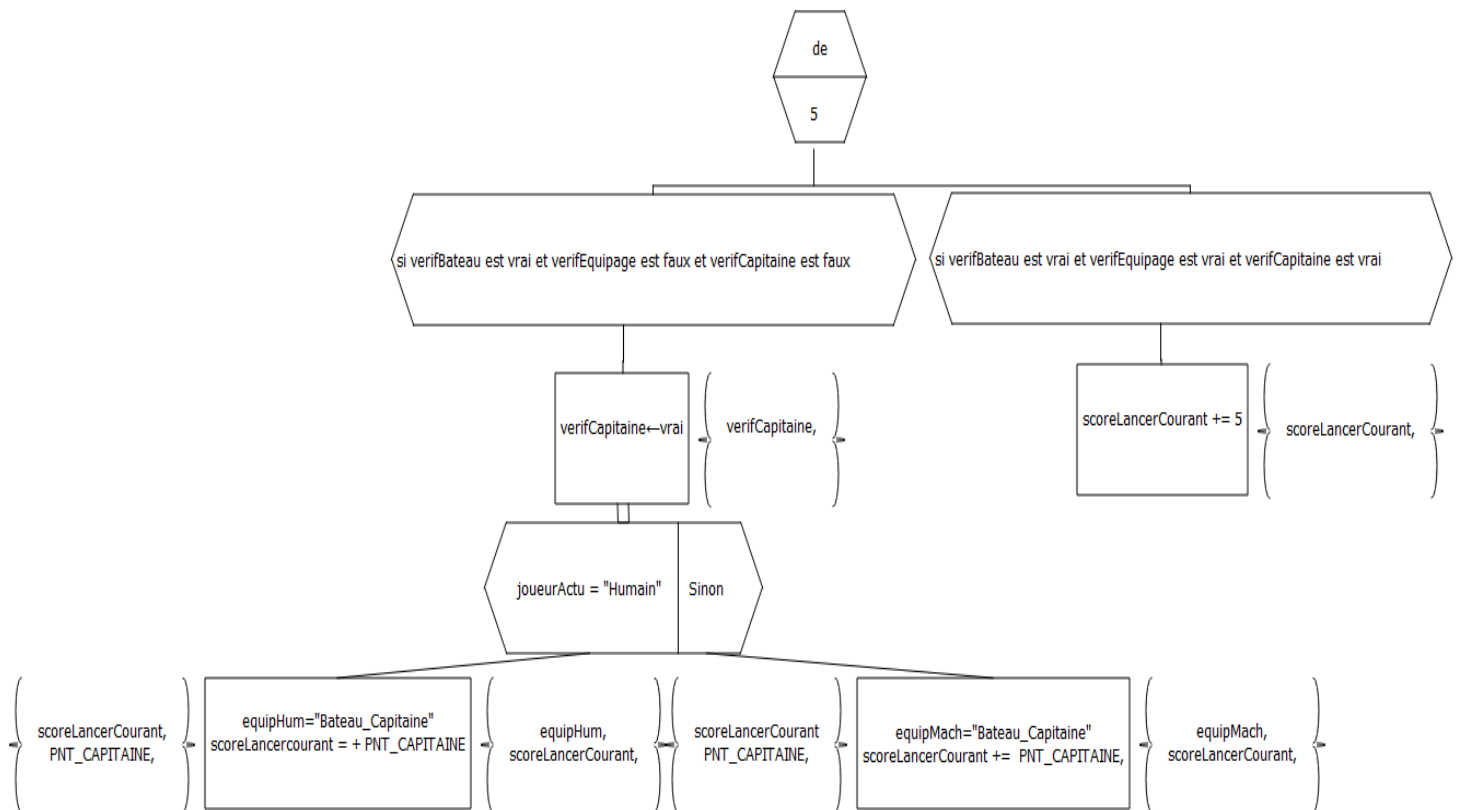
Les dés 4 à 6 ont un équipement bien précis donc on va vérifier à chaque fois si on n'a pas tiré un équipement qui va permettre de déverrouiller l'équipement suivant.

Par exemple pour avoir un capitaine il faut déjà avoir tiré l'équipage.



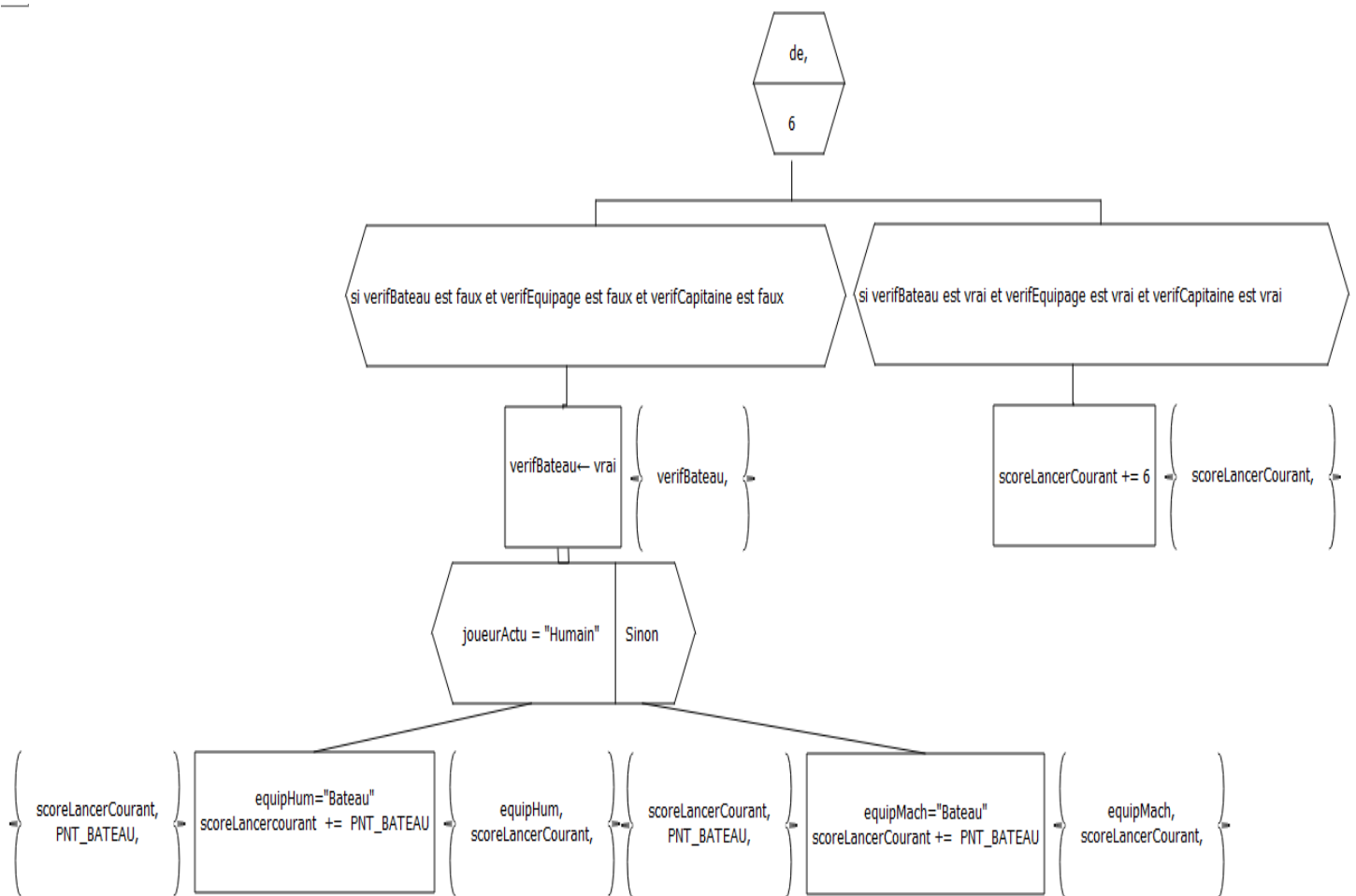
PTN_EQUIPAGE	entier	Variable qui contient le nombre de point qu'un capitaine vaut dans la partie, les points sont donnés lorsque un « 4 » est tiré	Prédéfini a 400 mais peut être modifiée	Non
equipHum	chaîne de caractères	Variable qui contient l'équipement du joueur humain tout au long de la partie	Calculé mais prédéfini a NON_EQUIPE	Non
scoreLancerCourant	entier	Score de chaque lancer de l'humain ou de la machine	Calculé	Non
equipMach	chaîne de caractères	Variable qui contient l'équipement de la machine tout au long de la partie	Calculé mais prédéfini a NON_EQUIPE	Non
verifEquipage	bool	Vérifie si l'équipage a déjà été tiré	Prédéfini sur faux	Non

Switch 5



equipMach	chaîne de caractères	Variable qui contient l'équipement de la machine tout au long de la partie	Calculé mais prédéfini a NON_EQUIPE	Non
equipHum	chaîne de caractères	Variable qui contient l'équipement du joueur humain tout au long de la partie	Calculé mais prédéfini a NON_EQUIPE	Non
PNT_CAPITAINE	entier	Variable qui contient le nombre de point qu'un capitaine vaut dans la partie, les points sont donnés lorsque un « 5 » est tiré	Prédéfini a 500 mais peut être modifiée	Non
verifCapitaine	bool	Vérifie si le capitaine a déjà été tiré	Prédéfini sur faux	Non
scoreLancerCourant	entier	Score de chaque lancer de l'humain ou de la machine	Calculé	Non

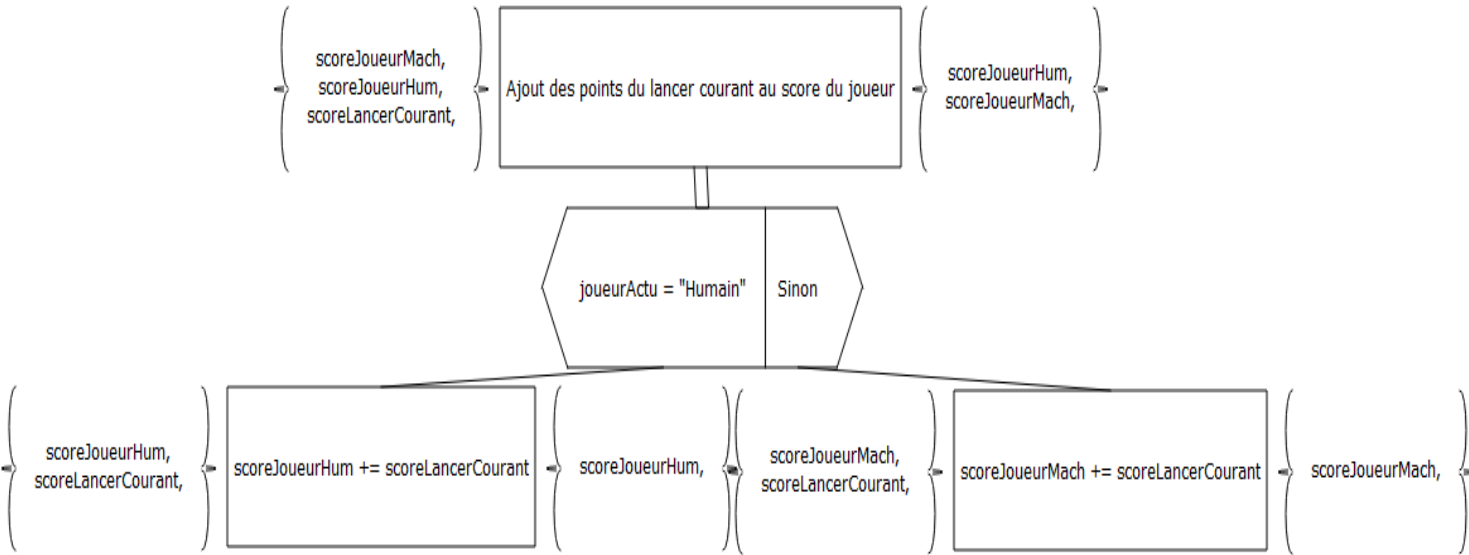
Switch 6



scoreLancerCourant	entier	Score de chaque lancer de l'humain ou de la machine	Calculé	Non
equipMach	chaîne de caractères	Variable qui contient l'équipement de la machine tout au long de la partie	Calculé mais prédéfini a NON_EQUIPE	Non
equipHum	chaîne de caractères	Variable qui contient l'équipement du joueur humain tout au long de la partie	Calculé mais prédéfini a NON_EQUIPE	Non
PTN_BATEAU	entier	Variable qui contient le nombre de point qu'un bateau vaut dans la partie, les points sont donnés lorsque un « 6 » est tiré	Prédéfini a 600 mais peut être modifiée	Non
verifBateau	bool	Vérifie si le bateau a déjà été tiré	Prédéfini sur faux	Non

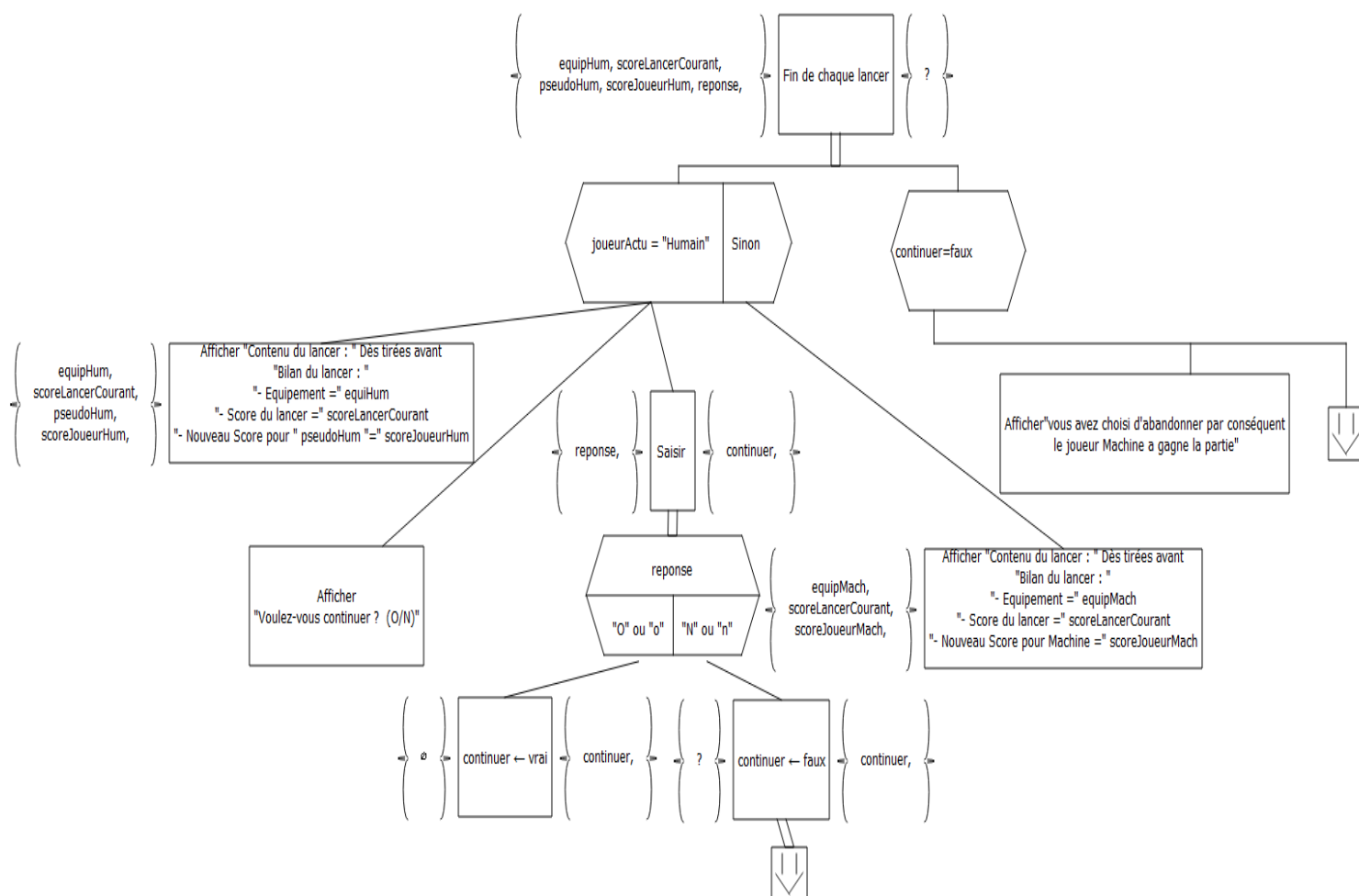
12. AJOUT DES POINTS DU LANCER COURANT AU SCORE DU JOUEUR :

Cette étape va permettre d'ajouter les points du lancer en fonction des dés tirés lors du lancer.



scoreJoueurHum	entier	Score de l'humain tout au long de la partie	Calculé	Non
scoreJoueurMach	entier	Score de la machine tout au long de la partie	Calculé	Non
scoreLancerCourant	entier	Score de chaque lancer de l'humain ou de la machine	Calculé	Non

13. FIN DE CHAQUE LANCER :

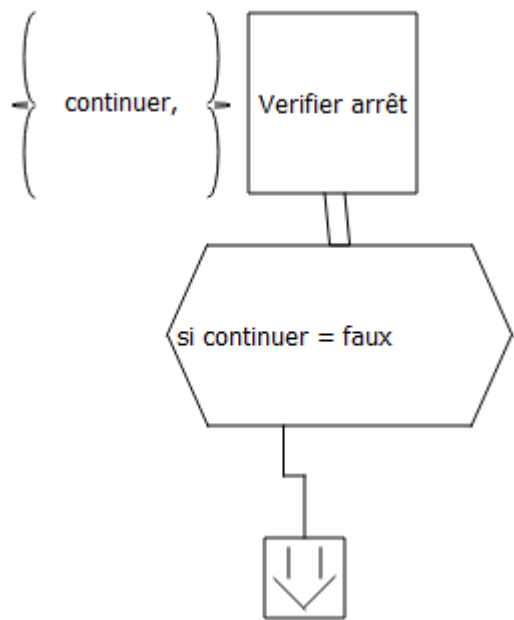


Cette étape va permettre de faire le bilan de chaque lancer et de demander au joueur s'il veut continuer la partie.

equipHum	chaîne de caractères	Variable qui contient l'équipement du joueur humain tout au long de la partie	Calculé mais prédéfini a NON_EQUIPE	Non
scoreLancerCourant	entier	Score de chaque lancer de l'humain ou de la machine	Calculé	Non
reponse	Caractère	Réponse du joueur pour savoir s'il veut continuer à jouer	Saisie	non
pseudoHum	chaîne de caractères	Variable qui contient le pseudo choisi par la personne qui joue contre la machine	Saisie	Non
scoreJoueurHum	entier	Score de l'humain tout au long de la partie	Calculé	Non

14. VERIF ARRET :

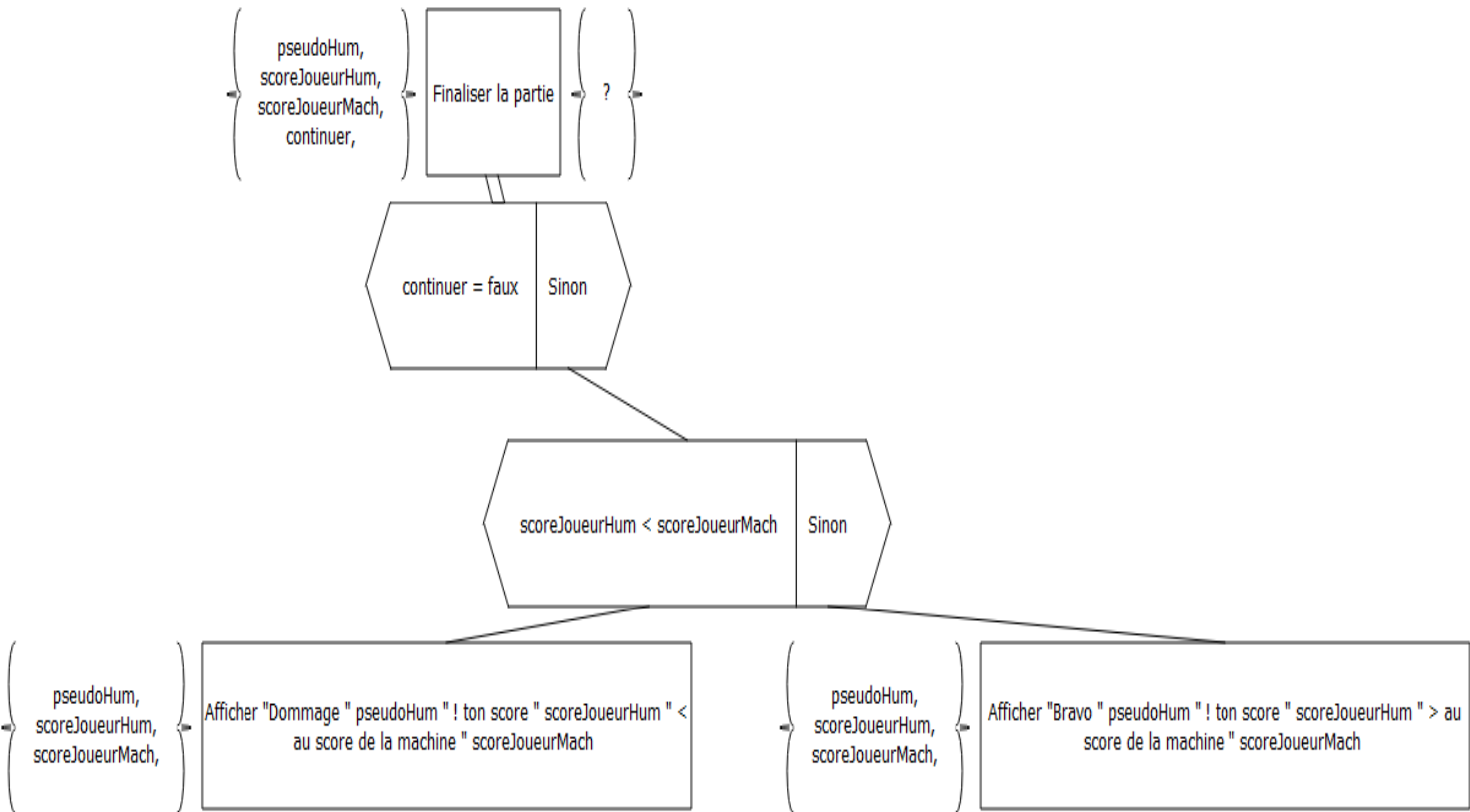
Cette étape va permettre de savoir si le joueur veut continuer la partie.



continuer	bool	Savoir si le joueur veut continuer à jouer	Prédéfini sur vrai	Non
-----------	------	--	--------------------	-----

15. FINALISER LA PARTIE :

Cette étape va consister à déterminer qui a gagné la partie entre le joueur et la machine et si le joueur a abandonné c'est la machine qui remporte la partie.



pseudoHum	chaîne de caractères	Variable qui contient le pseudo choisi par la personne qui joue contre la machine	Saisie	Non
scoreJoueurHum	entier	Score de l'humain tout au long de la partie	Calculé	Non
scoreJoueurMach	entier	Score de la machine tout au long de la partie	Calculé	Non
continuer	bool	Savoir si le joueur veut continuer à jouer	Prédéfini sur vrai	Non

16. CODE SOURCE :

```
/*
Programme ; Jeu de l'equipage
But : jouer au jeu de l'equipage
Date de derniere modification : 19/10/23
Remarques : code conforme aux specifications internes donnee en cours
*/
#include "game-tools.h"
#include <iostream>
using namespace std;

int main(void)
{
    // VARIABLES
    unsigned int compteurLance;           // nombre de lancés réalisés au fur et
à mesure de la partie
    string pseudoHum;                     // pseudo saisie par la personne qui
joue contre la machine
    string equipHum = "NON_EQUIPE";      // équipement du joueur humain tout au
long de la partie
    string equipMach = "NON_EQUIPE";     // équipement de la machine tout au
long de la partie
    unsigned int scoreLancerCourant = 0; // Score de chaque lancer de l'humain
ou de la machine
    unsigned int scoreJoueurHum = 0;     // Score de l'humain tout au long de la
partie
    unsigned int scoreJoueurMach = 0;    // Score de la machine tout au long de
la partie
    bool verifBateau = false;            // Vérifie si le bateau a déjà été tiré
    bool verifCapitaine = false;         // Vérifie si le capitaine a déjà été
tiré
    bool verifEquipage = false;          // Vérifie si l'équipage a déjà été
tiré
    bool continuer = false;              // Réponse du joueur pour savoir si il
veut continuer à jouer
    char reponse;                        // Reponse du joueur pour savoir si il
veut continuer
    string joueurActu;                   // Définit le joueur qui doit jouer
    unsigned short int nombreAlea;       // Nombre généré aléatoirement
    unsigned short int de;               // Valeur du dé
    string deTirer;                     // Liste de tout les dès tirers lors
d'un lancer
    unsigned short int iterateur = 0;    // Itérateur pour une boucle for
    // TRAITEMENTS

    // INITIALISER LA PARTIE

    int PNT_BATEAU = 600;                // nombre de point qu'un bateau vaut dans
la partie, les points sont donné lorsque un « 6 » est tiré
    int PNT_CAPITAINE = 500;             // nombre de point qu'un capitaine vaut
dans la partie, les points sont donné lorsque un « 5 »est tiré
    int PNT_EQUIPAGE = 400;              // nombre de point qu'un equipage vaut dans
la partie, les points sont donné lorsque un « 4 »est tiré
    unsigned int compteurManche = 0;     // nombre de manches réalisés au fur et à
mesure de la partie
    // MODIFIER LES PARAMETRES
    cout << "Voulez-vous modifier les parametres du jeu ? (O/N) " << endl;
```

```

cin >> reponse;
switch (reponse)
{
case 'O': // CAS OU LA REPONSE EST EN MAJUSCULE

    // CHANGER LES POINTS DE BATEAU
    cout << "Choisir le nombre de point que vaut un bateau " << endl;
    cin >> PNT_BATEAU;
    // CHANGER LES POINTS DU CAPITAINE
    cout << "Choisir le nombre de point que vaut un capitaine " << endl;
    cin >> PNT_CAPITAINE;
    // CHANGER LES POINTS DE L'EQUIPAGE
    cout << "Choisir le nombre de point que vaut un equipage " << endl;
    cin >> PNT_EQUIPAGE;
    break;

case 'o': // CAS OU LA REPONSE EST EN MINISCULE

    // CHANGER LES POINTS DE BATEAU
    cout << "Choisir le nombre de point que vaut un bateau " << endl;
    cin >> PNT_BATEAU;
    // CHANGER LES POINTS DU CAPITAINE
    cout << "Choisir le nombre de point que vaut un capitaine " << endl;
    cin >> PNT_CAPITAINE;
    // CHANGER LES POINTS DE L'EQUIPAGE
    cout << "Choisir le nombre de point que vaut un equipage " << endl;
    cin >> PNT_EQUIPAGE;
    break;

case 'N': // CAS AU CAS OU LA REPONSE EST EN MAJUSCULE
    break;

case 'n':
    break;

default:
    break;
}
// CHOISIR LE PSEUDO DE L'HUMAIN
cout << "Quel pseudo voulez-vous choisir ? " << endl;
cin >> pseudoHum;

// JOUER LA PARTIE

// DEFINIR LE JOUEUR
joueurActu = "Humain";
do
{
    // JOUER LA MANCHE
    // REMETTRE LES VARIABLES A DES VALEURS DE BASES POUR UN BON
FONCTIONNEMENT
    equipHum = "";
    equipMach = "";
    verifBateau = false;
    verifCapitaine = false;
    verifEquipage = false;
    cout << "Bilan partiel AVANT la nouvelle manche : " << endl;
    cout << "-score " << pseudoHum << " = " << scoreJoueurHum << endl;
    cout << "-score Machine = " << scoreJoueurMach << endl;
    // DETERMINER LE JOUEUR
    if (joueurActu == "Humain")

```

```

    {
        if (compteurManche < 5) // VERIFICATION QUE MOINS DE 5 MANCHES SE
SONT PASSER
        {
            compteurManche += 1;
            cout << "Manche # " << compteurManche << " joueur " << pseudoHum
<< endl;
        }
        else
        {
            break;
        }
    }
    else
    {
        cout << "Manche # " << compteurManche << " joueur Machine " << endl;
    }
    compteurLance = 0;
    // JOUER LES LANCERS
    do
    {
        scoreLancerCourant = 0;
        compteurLance += 1;
        deTirer = ""; //
        // TIRER 5 NOMBRES ALEATOIRES
        for (iterateur = 1; iterateur <= 5; iterateur++)
        {
            // TIRER UN NOMBRE ALEATOIRE
            nombreAlea = random(1, 6);
            de = nombreAlea; // ASSIGNATION DU NOMBRE ALEATOIRE AU DE

            // AJOUT DES POINTS EN FONCTION DE LA VALEUR DU DE

            switch (de)
            {
                case 6:
                    deTirer += "6 ";
                    if (verifBateau == false && verifCapitaine == false &&
verifEquipage == false) // VERIFICATION QUE LE BATEAU N'A PAS DEJA ETE TIRE POUR
AJOUTER LES POINTS D'UN BATEAU
                    {

                        verifBateau = true;
                        if (joueurActu == "Humain") // AJOUT DES POINTS AU SCORE
DU LANCER EN FONCTION DU JOUEUR
                        {
                            equipHum = "Bateau";
                            scoreLancerCourant += PNT_BATEAU;
                        }
                        else
                        {
                            equipMach = "Bateau";
                            scoreLancerCourant += PNT_BATEAU;
                        }
                    }
                    if (verifBateau == true && verifCapitaine == true &&
verifEquipage == true) /*VERIFICATION QUE LE BATEAU, LE CAPITAIN ET L'EQUIPAGE
ON DEJA ETE TIRE
POUR NE PAS RAJOUTER LES POINTS D'UN BATEAU MAIS LES POINTS NORMAUX*/
                    {

```

```

        scoreLancerCourant += 6;
    }

    break;
case 5:
    deTirer += "5 ";
    if (verifBateau == true && verifCapitaine == false &&
verifEquipage == false) // VERIFICATION QUE LE CAPITAINE N'A PAS DEJA ETE TIRE
POUR AJOUTER LES POINTS D'UN CAPITAINE
    {

        verifCapitaine = true;
        if (joueurActu == "Humain") // AJOUT DES POINTS AU SCORE
DU LANCER EN FONCTION DU JOUEUR
        {
            equipHum = "Bateau_Capitaine";
            scoreLancerCourant += PNT_CAPITAINE;
        }
        else
        {
            equipMach = "Bateau_Capitaine";
            scoreLancerCourant += PNT_CAPITAINE;
        }
    }
    if (verifBateau == true && verifCapitaine == true &&
verifEquipage == true) /*VERIFICATION QUE LE BATEAU, LE CAPITAINE ET L'EQUIPAGE
ON DEJA ETE TIRE

POUR NE PAS RAJOUTER LES POINTS D'UN CAPITAINE MAIS LES POINTS NORMAUX*/
    {

        scoreLancerCourant += 5;
    }
    break;
case 4:
    if (verifBateau == true && verifCapitaine == true &&
verifEquipage == true) /*VERIFICATION QUE LE BATEAU, LE CAPITAINE ET L'EQUIPAGE
ON DEJA ETE TIRE

POUR NE PAS RAJOUTER LES POINTS D'UN EQUIPAGE MAIS LES POINTS NORMAUX*/
    {

        scoreLancerCourant += 4;
    }
    deTirer += "4 ";
    if (verifBateau == true && verifCapitaine == true &&
verifEquipage == false) // VERIFICATION QUE LE EQUIPAGE N'A PAS DEJA ETE TIRE
POUR AJOUTER LES POINTS D'UN EQUIPAGE
    {

        verifEquipage = true;
        if (joueurActu == "Humain") // AJOUT DES POINTS AU SCORE
DU LANCER EN FONCTION DU JOUEUR
        {
            equipHum = "Bateau_Capitaine_Equipage";
            scoreLancerCourant += PNT_EQUIPAGE;
        }
        else
        {
            equipMach = "Bateau_Capitaine_Equipage";

```

```

        scoreLancerCourant += PNT_EQUIPAGE;
    }
}

break;
case 3:
    deTirer += "3 ";
    if (verifBateau == true && verifCapitaine == true &&
verifEquipage == true) /*VERIFICATION QUE LE BATEAU, LE CAPITAIN ET L'EQUIPAGE
ON DEJA ETE TIRE POUR AJOUTER LES POINTS NORMAUX*/
    {
        scoreLancerCourant += 3;
    }
    break;
case 2:
    deTirer += "2 ";
    if (verifBateau == true && verifCapitaine == true &&
verifEquipage == true) /*VERIFICATION QUE LE BATEAU, LE CAPITAIN ET L'EQUIPAGE
ON DEJA ETE TIRE POUR AJOUTER LES POINTS NORMAUX*/
    {
        scoreLancerCourant += 2;
    }
    break;
case 1:
    deTirer += "1 ";
    if (verifBateau == true && verifCapitaine == true &&
verifEquipage == true) /*VERIFICATION QUE LE BATEAU, LE CAPITAIN ET L'EQUIPAGE
ON DEJA ETE TIRE POUR AJOUTER LES POINTS NORMAUX*/
    {
        scoreLancerCourant += 1;
    }

default:
    break;
}

// AJOUT DES POINTS DU LANCER AU SCORE DU JOUEUR
}

// FIN DE CHAQUE LANCER
if (joueurActu == "Humain") // AJOUT DES POINTS DU LANCER COURANTS
AU SCORE DU JOUEUR ACTUEL
{
    scoreJoueurHum += scoreLancerCourant; // AJOUT DES POINTS DU
LANCER COURANTS AU SCORE DU JOUEUR HUMAIN
}
else
{
    scoreJoueurMach += scoreLancerCourant; // AJOUT DES POINTS DU
LANCER COURANTS AU SCORE DU JOUEUR MACHINE
}
if (joueurActu == "Humain")
{ // AFFICHER LE BILAN DU LANCER DU JOUEUR HUMAIN
    cout << "Lancer numero : " << compteurLance << endl;
    cout << "Contenu du lancer : " << deTirer << endl;
}

```

```

        cout << "-Bilan du lancer : " << endl;
        cout << "    -Equipement = " << equipHum << endl;
        cout << "    -Score du lancer = " << scoreLancerCourant << endl;
        cout << "    -Nouveau score pour " << pseudoHum << " = " <<
scoreJoueurHum << endl;
        cout << "Voulez-vous continuer (O/N) " << endl;
        cin >> reponse;

        switch (reponse)
        {
            case 'O':
                continuer = true;
                break;
            case 'o':
                continuer = true;
                break;
            case 'N':
                continuer = false;
                break;
            case 'n':
                continuer = false;
                break;

            default:
                break;
        }
    }
    else
    { // AFFICHER LE BILAN DU LANCER DU JOUEUR MACHINE
        cout << "Lancer numero : " << compteurLance << endl;
        cout << "Contenu du lancer : " << deTirer << endl;
        cout << "Bilan du lancer : " << endl;
        cout << "    -Equipement = " << equipMach << endl;
        cout << "    -Score du lancer = " << scoreLancerCourant << endl;
        cout << "    -Nouveau score pour Machine = " << scoreJoueurMach
<< endl;
    }
    if (continuer == false)
    {
        cout << "Vous avez choisi d'abandonner par consequent le joueur
Machine a gagne la partie" << endl;
        pause(5);
        break;
    }

    } while (compteurLance != 3); // BOUCLE POUR SAVOIR SI LES TROIS MANCHES
// CHANGER DE JOUEUR
    if (joueurActu == "Humain")
    {
        joueurActu = "Machine";
    }
    else
    {
        joueurActu = "Humain";
    }

    } while (continuer != false);
// FINALISER LA PARTIE
    if (continuer == false) // TEST POUR QUE QUAND LE JOUEUR HUMAIN ABANDONNE LE
PROGRAMME N'AFFICHE PAS LA FIN DE PARTIE NORMAL
    {

```

```

    }
    else
    { // FIN DE PARTIE SI LE JOUEUR HUMAIN N'A PAS ABANDONNE
        if (scoreJoueurHum < scoreJoueurMach)
        {
            cout << "Dommage " << pseudoHum << " ! ton score : " <<
scoreJoueurHum << " < au score de la machine : " << scoreJoueurMach << endl; //
AFFICHAGE DE FIN DE PARTIE SI LE JOUEUR HUMAIN A PERDU
        }
        else
        {
            cout << "Bravo " << pseudoHum << " ! ton score : " << scoreJoueurHum
<< " > au score de la machine : " << scoreJoueurMach << endl; // AFFICHAGE DE
FIN DE PARTIE SI LE JOUEUR HUMAIN A GAGNE
        }
    }

    return 0;
}

```