

Séquence 4 : Algorithmes et programmation

I- Etudier un programme simple

- 1) Environnement de Scratch
- 2) Construire une première animation (Traverser la ville en auto)
- 3) Construire un script qui répond à un événement
 - a) Notion de boucle : Activités sans ordinateur (P4)
 - b) Activité avec Scratch (Danser avec Khalid)
- 4) Programmer des scripts se déroulant en parallèle
- 5) Faire appel à un sous-programme

Faire les exercices : P540 (exercice 3)

II- Connaître les instructions de contrôle

- 1) Les conditions :
 - a) Activités sans ordinateur (P5)
 - b) Boucles et conditions imbriquées : Activités sans ordinateur (P6 fusée)
 - c) Avec l'ordinateur : Réaliser un jeu de Pong
- 2) Les variables
 - a) Activités sans ordinateur (P7)
 - b) Utiliser une boucle, étudier une variable : Activité avec l'ordinateur (Construire une spirale)

Programmer des déplacements déclenchés par un événement (Jeu du Labyrinthe)

Travailler avec des variables (compter le nombre d'apparition d'un caractère dans une phrase)

Programmer des actions déclenchées par un événement (Jeu de tir)

Faire les exercices 2 et 3 page 546

III- Etudier la logique algorithmique d'un programme

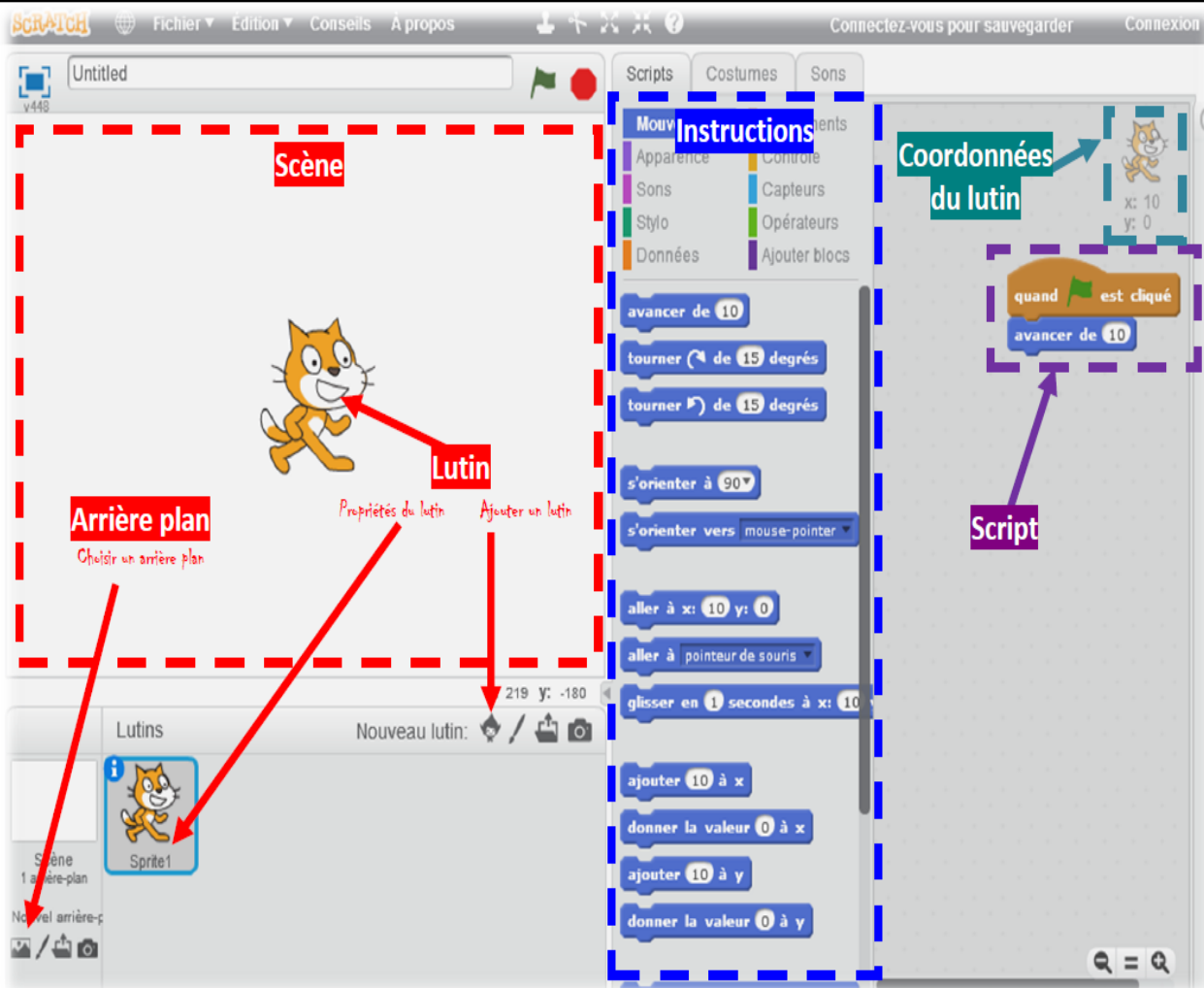
- 1) Décrire une démarche à l'aide d'un algorithme
- 2) Distinguer les étapes d'un programme

Séquence 4 : Algorithme et programmation

I- Etudier un programme simple

1) Environnement de Scratch

SCRATCH : vocabulaire



SCRATCH : liste des commandes (instructions)

MOUVEMENT	APPARENCE	SON	STYLO
<p>avancer de 10</p> <p>tourner (↶) de 15 degrés</p> <p>tourner (↷) de 15 degrés</p> <p>s'orienter à 90</p> <p>s'orienter vers mouse-pointer</p> <p>aller à x: 271 y: 0</p> <p>aller à pointeur de souris</p> <p>glisser en 1 secondes à x: 271</p> <p>ajouter 10 à x</p> <p>donner la valeur 0 à x</p> <p>ajouter 10 à y</p> <p>donner la valeur 0 à y</p> <p>rebondir si le bord est atteint</p> <p>fixer le sens de rotation position</p> <p><input type="checkbox"/> abscisse x</p> <p><input type="checkbox"/> ordonnée y</p> <p><input type="checkbox"/> direction</p>	<p>dire Hello! pendant 2 secondes</p> <p>dire Hello!</p> <p>penser à Hmm... pendant 2 secondes</p> <p>penser à Hmm...</p> <p>montrer</p> <p>cacher</p> <p>basculer sur costume costume2</p> <p>costume suivant</p> <p>basculer sur l'arrière-plan arrière-plan</p> <p>ajouter à l'effet couleur 25</p> <p>mettre l'effet couleur à 0</p> <p>annuler les effets graphiques</p> <p>ajouter 10 à la taille</p> <p>mettre à 100 % de la taille initiale</p> <p>envoyer au premier plan</p> <p>déplacer de 1 plans arrière</p> <p><input type="checkbox"/> costume n°</p> <p><input type="checkbox"/> nom de l'arrière-plan</p> <p><input type="checkbox"/> taille</p>	<p>jouer le son miaou</p> <p>jouer le son miaou jusqu'au bout</p> <p>arrêter tous les sons</p> <p>jouer du tambour 1 pendant 0.25</p> <p>faire une pause pour 0.25 temps</p> <p>jouer la note 60 pendant 0.5</p> <p>choisir l'instrument n° 1</p> <p>ajouter -10 au volume</p> <p>mettre le volume au niveau 100</p> <p><input type="checkbox"/> volume</p> <p>ajouter 20 au tempo</p> <p>mettre le tempo à 60 bpm</p> <p><input type="checkbox"/> tempo</p>	<p>effacer tout</p> <p>estampiller</p> <p>stylo en position d'écriture</p> <p>relever le stylo</p> <p>choisir la couleur pour le stylo</p> <p>ajouter 10 à couleur du stylo</p> <p>mettre la couleur du stylo à 0</p> <p>ajouter 10 à l'intensité du stylo</p> <p>choisir l'intensité 50 pour le stylo</p> <p>ajouter 1 à la taille du stylo</p> <p>choisir la taille 1 pour le stylo</p>

SCRATCH : liste des commandes (suite)

DONNEES	CONTROLE	CAPTEURS	OPERATEURS
Créer une variable Créer une liste	attendre 1 secondes répéter 10 fois répéter indéfiniment si alors si alors sinon attendre jusqu'à répéter jusqu'à stop tout quand je commence comme un clone créer un clone de moi-même supprimer ce clone	touché? couleur touchée? couleur touchée ? distance de demander What's your name? et attendre réponse touche espace pressée? souris pressée? souris x souris y volume sonore video mouvement sur ce lutin activer la vidéo Activé mettre la transparence vidéo à chronomètre réinitialiser le chronomètre abscisse x de Lutin1 actuel minute jours depuis 2000 nom d'utilisateur	+ - * / nombre aléatoire entre 1 et 10 < = > et ou non regroupe hello world lettre 1 de world longueur de world modulo arrondi de racine de 9
EVENEMENTS			AJOUTER BLOCS
quand cliqué quand espace est cliqué quand ce lutin est cliqué quand l'arrière-plan bascule sur quand volume sonore > 10 quand je reçois message1 envoyer à tous message1 envoyer à tous message1 et attendre			Créer un bloc Ajouter une extension

Trace écrite :

Scratch est un **langage de programmation**. Il en existe beaucoup d'autres : Python, Visual Basic... Un **langage** permet d'écrire des **programmes exécutés par l'ordinateur**.

Les **objets du langage Scratch** sont : Un lutin, un arrière plan de la scène,.....

On peut associer à chacun d'eux des scripts, des costumes (ou arrière plans) et des sons

2) Construire une première animation

Mouvement (P1)

Pour se déplacer

choisir une seule instruction proposée

```

quand cliqué
  avancer de 10
  avancer de -10
  tourner de 15 degrés
  aller à x: 219 y: -162
  glisser en 1 secondes à x: 219 y: -162
  
```

Apparence (P2)

Pour parler, penser...

choisir une seule instruction proposée

```

quand espace est cliqué
  dire Hello! pendant 2 secondes
  dire Hello!
  penser à Hmm... pendant 2 secondes
  penser à Hmm...
  
```

Son (P3)

Pour créer un son, une mélodie...

choisir une seule instruction proposée

```

quand ce lutin est cliqué
  jouer le son miaou
  jouer du tambour pendant 0.25 temps
  jouer la note 60 pendant 0.5 temps
  choisir l'instrument n° 1
  
```

Avec deux lutins (P4)

Programmer deux scripts différents - Faire interagir 2 lutins

Sélectionner lutin 1

Sélectionner lutin 2

choisir une autre commande de départ pour faire évoluer les lutins séparément par ex :

```

quand cliqué
  avancer de 100
  
```

```

quand cliqué
  dire Hello!
  
```

```

quand espace est cliqué
  
```

Contrôle (P5 → P11)

Commandes « Répéter »

P5)

```

quand cliqué
  avancer de 10
  
```

Le lutin avance d'une certaine distance (10 graduations) puis s'arrête. Si on appuie encore sur le drapeau vert, alors il avance encore de 10, etc...

P6) Répéter 10 fois

```

quand cliqué
  répéter 10 fois
    avancer de 10
  
```

Le lutin avance de 100 graduations puis s'arrête.

P7) Remise à l'origine du lutin

```

quand cliqué
  aller à x: 0 y: 0
  répéter 10 fois
    avancer de 10
  
```

Le lutin revient à sa position initiale, puis avance de 100 graduations.

P8) Répéter indéfiniment

```

quand cliqué
  aller à x: 0 y: 0
  répéter indéfiniment
    avancer de 10
  
```

Le lutin avance sans jamais s'arrêter.

P9) Répéter « jusqu'à »

```

quand cliqué
  répéter jusqu'à bord touché?
    avancer de 10
  
```

Le lutin avance jusqu'au bord de la scène, puis s'arrête (il revient à sa position initiale dans le 2^{ème} cas).

```

quand cliqué
  aller à x: 0 y: 0
  répéter jusqu'à bord touché?
    avancer de 10
  
```

P10) Répéter indéfiniment, puis rebondir si bord atteint

```

quand cliqué
  aller à x: 0 y: 0
  répéter indéfiniment
    avancer de 10
    rebondir si le bord est atteint
  
```

Le lutin avance jusqu'au bord, puis rebondit pour avancer dans l'autre sens... indéfiniment.

P11) Avancer de manière « aléatoire »

```

quand cliqué
  tourner de 15 degrés
  répéter indéfiniment
    avancer de 10
    rebondir si le bord est atteint
  
```

Le lutin avance sans arrêt dans « n'importe quel sens » en rebondissant dès qu'il touche le bord de la scène.

Construire une première animation (Traverser la ville en auto)

Faire l'activité 2 page 536 du manuel TransMath

3) Construire un script qui répond à un événement.

a) Notion de boucle : Activités sans ordinateur.

Algorithme et Positions

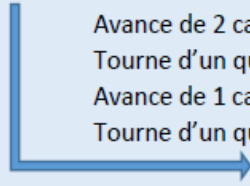
: Place le bateau

Exercice : Le bateau qui est sur le quadrillage va appliquer les consignes de l'algorithme. Sur cette feuille, dessine le bateau dans sa position finale.

Algorithme :

REPETE 2 FOIS

Avance de 2 cases
Tourne d'un quart de tour à droite
Avance de 1 case
Tourne d'un quart de tour à gauche



Remarque1 : Le bateau avance dans le sens de la marche (voir flèche)



Remarque 2 : Le cas du bateau qui tourne d'un quart de tour à droite



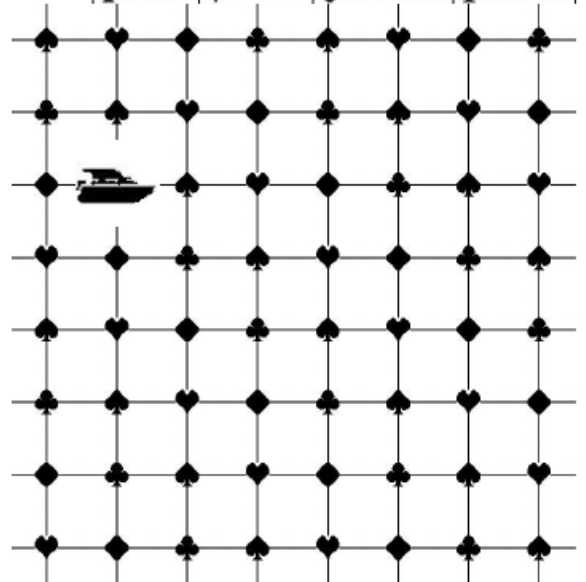
devient

Remarque 2 : Le cas du bateau qui tourne d'un quart de tour à gauche



devient

♠ PIQUE ♥ COEUR ♦ CARREAU ♣ TREFLE



Algorithme et Positions

: Place le plongeur

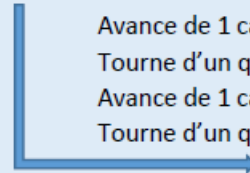
Exercice : Le plongeur qui est sur le quadrillage va appliquer les consignes de l'algorithme. Sur cette feuille, dessine le plongeur dans sa position finale.

Algorithme :

Avance de 1 cases

REPETE 3 FOIS

Avance de 1 case
Tourne d'un quart de tour à droite
Avance de 1 case
Tourne d'un quart de tour à gauche

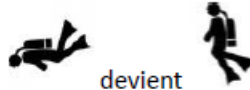


Avance de 2 cases

Remarque1 : Le plongeur avance dans le sens de la flèche.



Remarque 2 : Le cas du plongeur qui tourne d'un quart de tour à droite

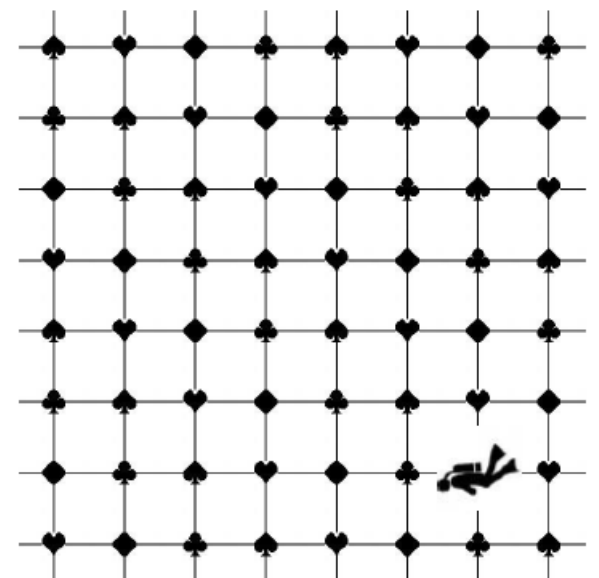


devient

Remarque 3 : Le cas du plongeur qui tourne d'un quart de tour à gauche



devient



b) Activité avec le langage Scratch (Danser avec Khalid)

Faire l'activité 3 page 537 du manuel TransMath

Trace écrite :

- Une **boucle** est une répétition d'une série d'instructions
- L'exécution d'un programme peut être déclenchée par un événement (l'appui sur une touche de clavier ou un bouton de la souris. Un même événement peut déclencher plusieurs scripts. Ces programmes sont prêts à réagir et sont en attente de la réalisation de l'événement choisi par le programmeur.
- La programmation de ces scripts est appelée **programmation événementielle**.

4) Programmer des scripts se déroulant en parallèle

Faire l'activité 4 page 538 du manuel TransMath

Trace écrite :

- A chaque objet du langage on peut associer un script.
- Chaque script peut être exécuté seul. Cela permet de valider son fonctionnement, mais on peut aussi provoquer leur exécution en **parallèle**.

5) Faire appel à un sous-programme

Faire l'activité 5 page 539 du manuel TransMath

Trace écrite :

- Un **sous-programme** permet de réaliser une tâche particulière
- D'autres programmes peuvent alors appeler ce sous-programme afin d'exécuter cette tâche.
- Programmer ainsi permet de décomposer un problème en sous-problèmes et de structurer les programmes.

Faire les exercices : P540 (exercice 3)

II- Connaitre les instructions de contrôle

1) Les conditions :

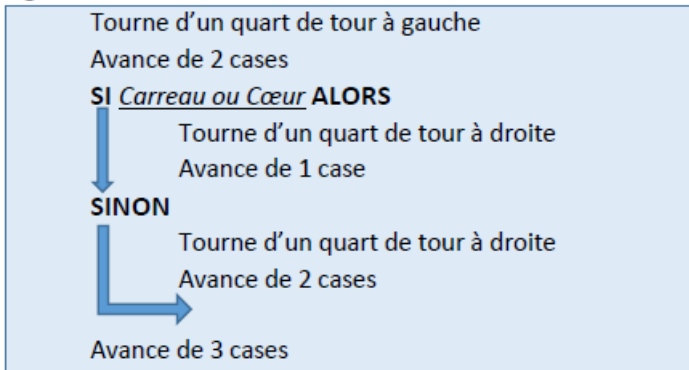
a) Activités sans ordinateur

Algorithme et Positions

: Place le vélo

Exercice : Le vélo qui est sur le quadrillage va appliquer les consignes de l'algorithme. Sur cette feuille, dessine le vélo dans sa position finale.

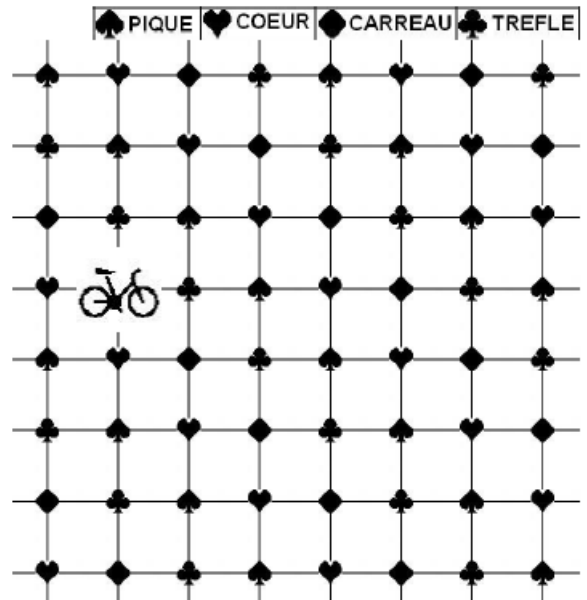
Algorithme :



Remarque1 : « SI Carreau ou Cœur » signifie « SI le vélo est positionné sur un Carreau ou bien un Cœur »

Remarque2 : Si le vélo est positionné sur un carreau ou un cœur alors il faut exécuter les instructions qui sont entre ALORS et SINON et ne pas exécuter les instructions qui sont entre SINON et la flèche horizontale.

Remarque3 : Si le vélo n'est pas positionné sur un carreau ou un cœur alors il ne faut pas exécuter les instructions qui sont entre ALORS et SINON et il faut exécuter les instructions qui sont entre SINON et la flèche horizontale.

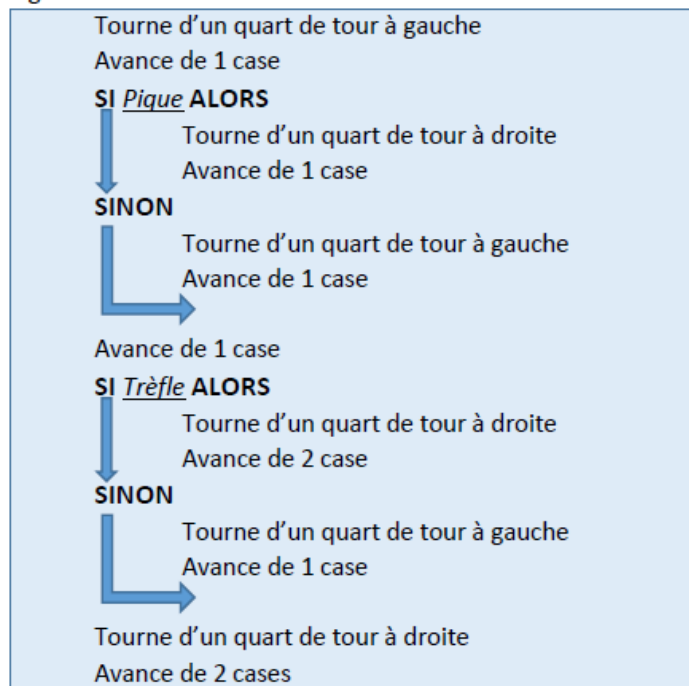


Algorithme et Positions

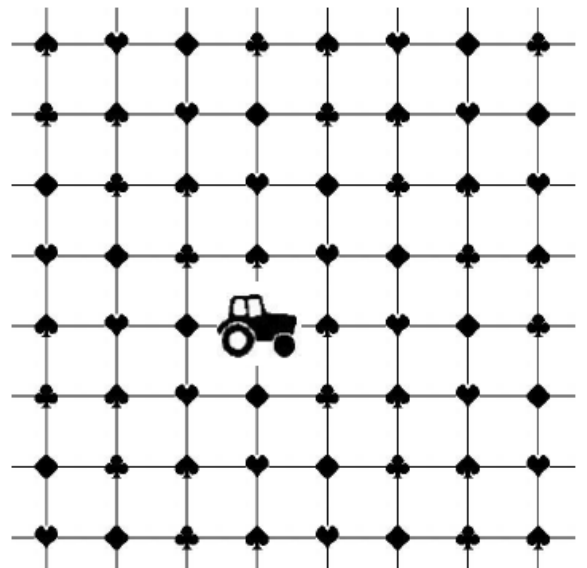
: Place le tracteur

Exercice : Le tracteur qui est sur le quadrillage va appliquer les consignes de l'algorithme. Sur cette feuille, dessine le tracteur dans sa position finale.

Algorithme :



PIQUE COEUR CARREAU TREFLE



b) **Boucles et conditions imbriquées** (appel sous-programme ou séquence):

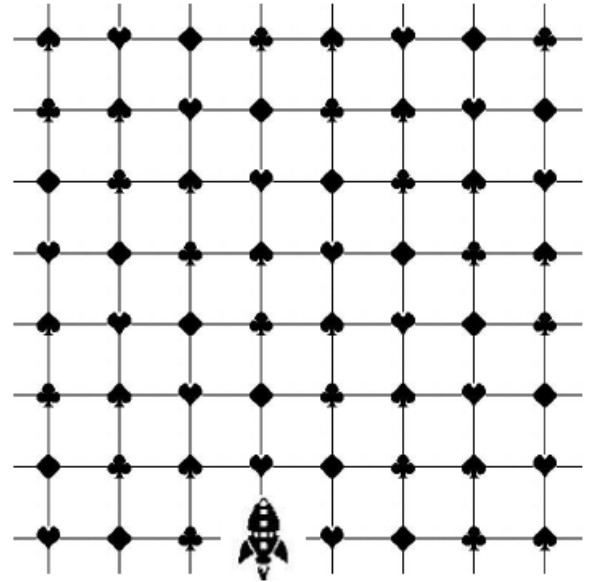
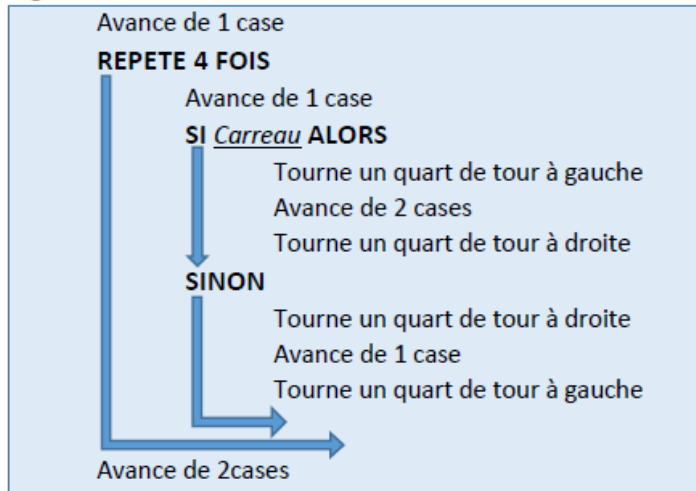
Activités sans ordinateur

Algorithme et Positions

: Place la fusée

Exercice : La fusée qui est sur le quadrillage va appliquer les consignes de l'algorithme. Sur cette feuille, dessine la fusée dans sa position finale.

Algorithme :

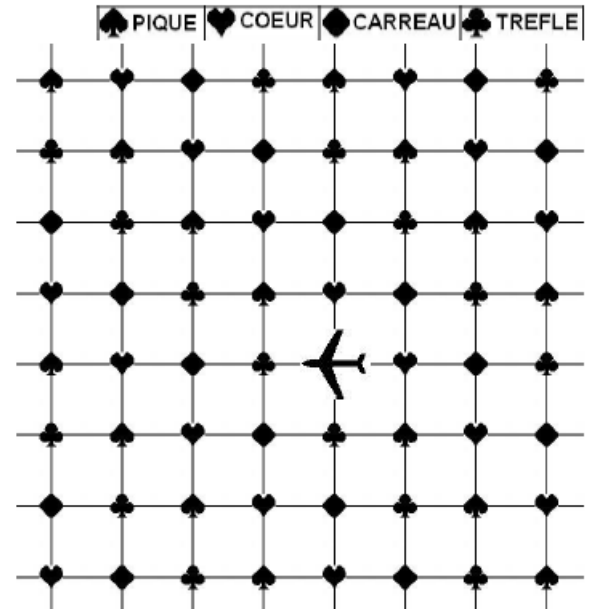
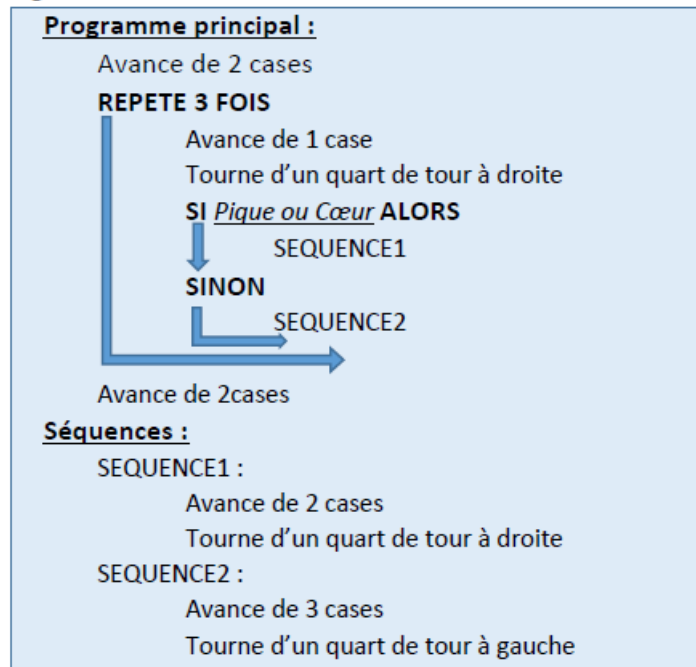


Algorithme et Positions

: Place l'avion

Exercice : L'avion qui est sur le quadrillage va appliquer les consignes de l'algorithme. Sur cette feuille, dessine l'avion dans sa position finale.

Algorithme :



c) **Avec l'ordinateur :** Réaliser un jeu de Pong

Faire l'activité 1 page 541 du manuel TransMath

Trace écrite :

- L'instruction conditionnelle : « **Si** condition **alors** instructions » permet d'exécuter certaines instructions seulement dans le cas où une condition est vérifiée.
- Sous la forme « **Si** condition **alors** instructions **sinon** instructions », elle précise aussi les instructions à appliquer dans le cas où la condition n'est pas vérifiée.

2) Les variables

a) Activités sans ordinateur

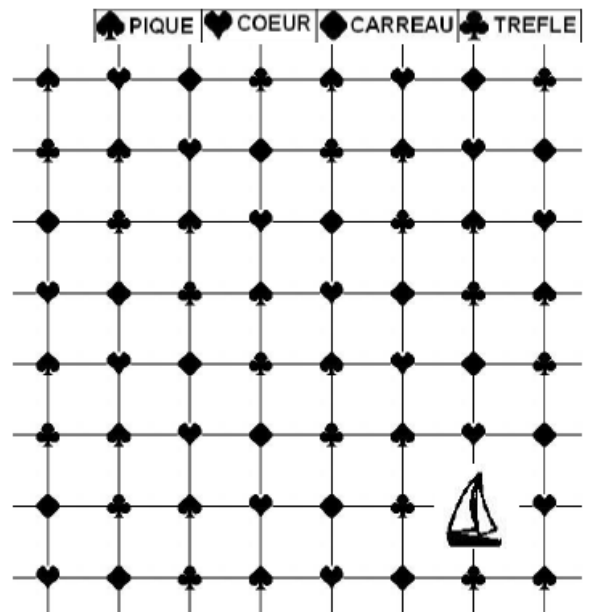
Algorithme et Positions

: Place le voilier

Exercice : Le voilier qui est sur le quadrillage va appliquer les consignes de l'algorithme. Sur cette feuille, dessine le voilier dans sa position finale.

Algorithme :

valeur est une variable
Mettre *valeur* à 2
Avance de 1 case
REPETE *valeur* **FOIS**
 Avance de 1 case
 Tourne d'un quart de tour à droite
 Avance de *valeur* cases
 Tourne d'un quart de tout à gauche
Avance de *valeur* cases



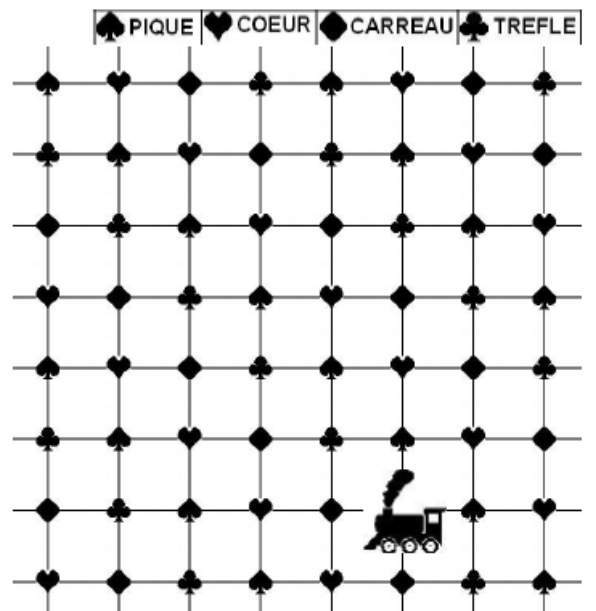
Algorithme et Positions

: Place la locomotive

Exercice : La locomotive qui est sur le quadrillage va appliquer les consignes de l'algorithme. Sur cette feuille, dessine la locomotive dans sa position finale.

Algorithme :

distance est une variable
Mettre *distance* à 2
Avance de 1 case
SI *Trefle* **ALORS**
 REPETE 2 FOIS
 Tourne un quart de tour à gauche
 Avance de *distance* cases
 Tourne un quart de tour à droite
SINON
 REPETE 2 FOIS
 Tourne un quart de tour à droite
 Avance de *distance* cases
 Tourne un quart de tour à gauche
Avance de (*distance*+1) cases

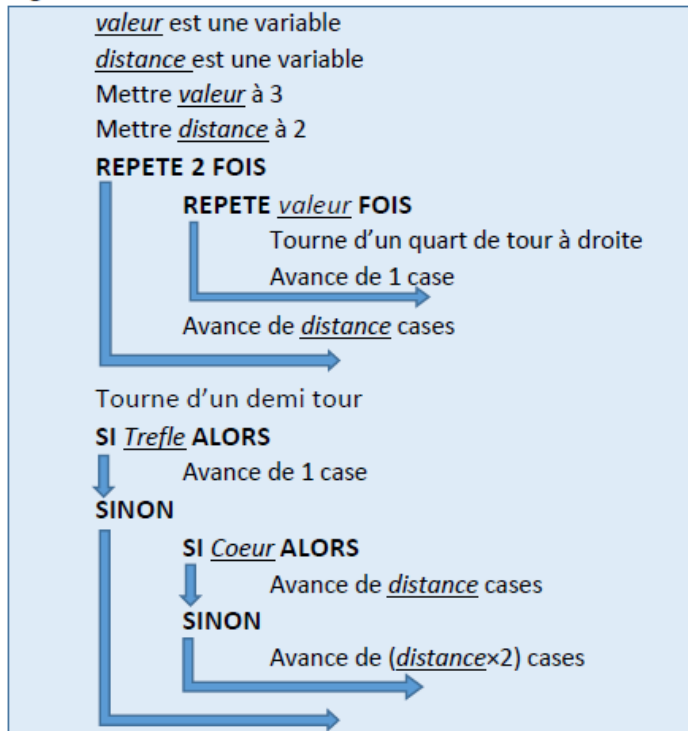


Algorithme et Positions

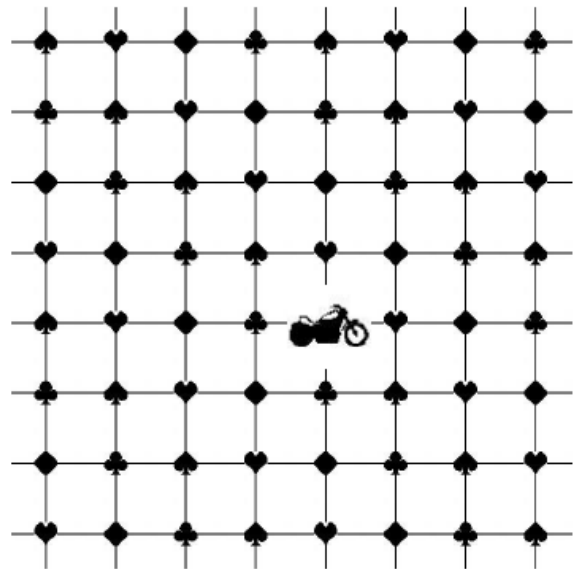
: Place la moto

Exercice : La moto qui est sur le quadrillage va appliquer les consignes de l'algorithme. Sur cette feuille, dessine la moto dans sa position finale.

Algorithme :



♠ PIQUE | ♥ COEUR | ♦ CARREAU | ♣ TREFLE



Ici les variables sont restées fixes

- b) **Utiliser une boucle, étudier une variable** : Activité avec l'ordinateur
 (Construire une spirale)

Faire l'activité 2 page 542 du manuel TransMath

Trace écrite :

- Une **variable** peut être assimilée à une boîte que le programme repère par son **nom**. Le **contenu** de la boîte évolue lorsque le programme est exécuté.
- Une boucle **répéter fois** permet de répéter des instructions un nombre de fois prévu à l'avance
- Sous la forme **condition** **répéter jusqu'à**, les instructions sont répétées jusqu'à ce qu'une vraie et le nombre de répétitions n'est en général pas prévu à l'avance.

Programmer des déplacements déclenchés par un événement (Jeu du Labyrinthe)

Faire l'activité 3 page 543 du manuel TransMath

Travailler avec des variables (compter le nombre d'apparition d'un caractère dans une phrase)

Faire l'activité 4 page 544 du manuel TransMath

Programmer des actions déclenchées par un événement (Jeu de tir)

Faire l'activité 1 page 547 du manuel TransMath

Faire les exercices 2 et 3 page 546

III- Etudier la logique algorithmique d'un programme

1) Décrire une démarche à l'aide d'un algorithme

Faire l'activité 2 page 548 du manuel TransMath

Trace écrite :

- Un **algorithme** décrit la démarche logique d'un programme. Il met en évidence la structure de ce programme et fait apparaître ses variables.
- Un fois mis au point, l'algorithme est codé dans un langage de programmation.

2) Distinguer les étapes d'un programme

Faire l'activité 4 page 550 du manuel TransMath

Trace écrite :

- Dans une démarche de programmation, on distingue les étapes successives du programme et on résout chacune de ces étapes.

Faire les exercices 1 et 5 pages 552 et 553