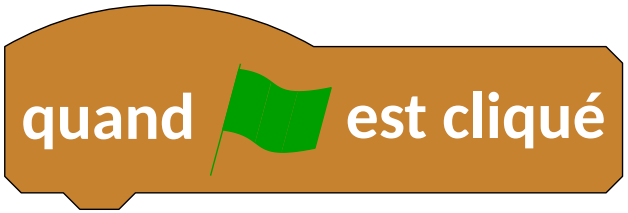
MP-Scratch

METAPOST au service de l'algorithmique

CHRISTOPHE POULAIN
christophe.poulain@melusine.eu.org
Collège Paul Eluard

Un exemple



Un exemple

Un exemple


quand est cliqué
répéter indéfiniment
dire Ce stage est super ! Vive METAPOST ! $\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$

Un exemple


```
input mp-scratch

beginfig(1);
draw Drapeau;
draw RepeterI;
draw Dire("\opSimple{Ce␣stage␣est␣super␣!␣Vive␣\MP␣!␣$\displaystyle\sum_{n=1}^\infty
␣␣\frac1{n^2}=\frac{\pi^2}6$}");
draw FinBlocRepeterI;
endfig;
end;
```

Algorithme en LaTeX

- package **listings**

Algorithme en LaTeX

- package **listings**

```
Demander un nombre
Ajouter 2
Multiplier par 3
Retrancher 2
Afficher le r\'esultat
```

Algorithme en LaTeX

- package **listings**
- package **algorithms**

Algorithme en LaTeX

- package **listings**
- package **algorithms**

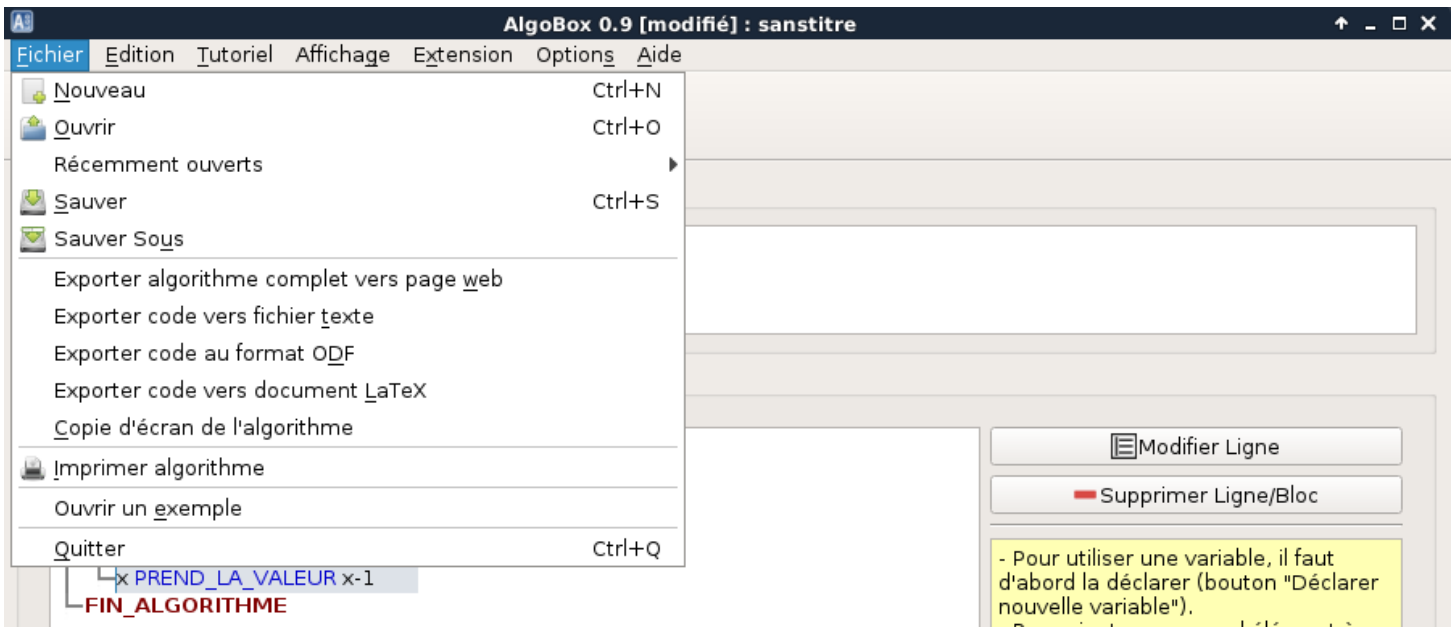
1: Demander un nombre n
2: $n+2 \rightarrow p$
3: $p*3 \rightarrow q$
4: $q-2 \rightarrow r$
5: Afficher le résultat r

Algorithme en LaTeX

- package **listings**
- package **algorithms**
- **algobox**

Algorithme en LaTeX

- package **listings**
- package **algorithms**
- **algobox**



Algorithme en LaTeX

- package **listings**
- package **algorithms**
- **algobox**

```
1: VARIABLES
2: x EST_DU_TYPE NOMBRE
3: DEBUT_ALGORITHME
4:     x PREND_LA_VALEUR x+1
5:     x PREND_LA_VALEUR 3x
6:     x PREND_LA_VALEUR x-1
7: FIN_ALGORITHME
```

Algorithme en LaTeX

- package **listings**
- package **algorithms**
- **algobox**
- captures d'écran

Algorithme en LaTeX

- package **listings**
- package **algorithms**
- **algobox**
- captures d'écran

Pourquoi MP-Scratch ?

Pourquoi MP-Scratch ?

- Pourquoi pas ? :)

Pourquoi MP-Scratch ?

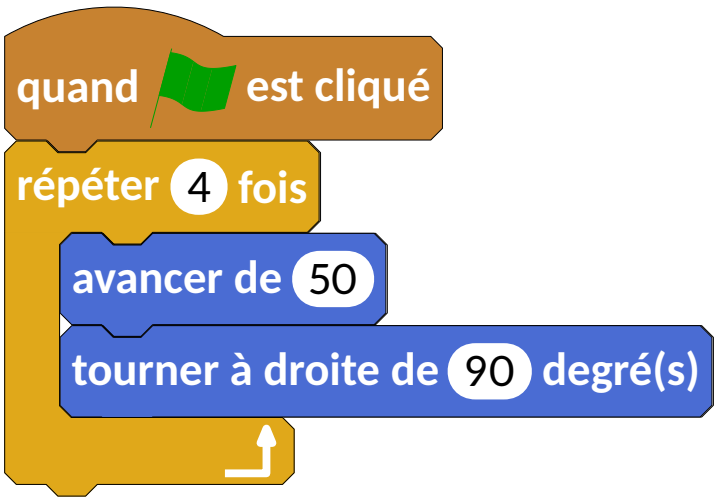
- Pourquoi pas ? :)
- Basé sur METAPOST

Pourquoi MP-Scratch ?

- Pourquoi pas ? :)
- Basé sur METAPOST
- Introduction de l'algorithmique au collège.

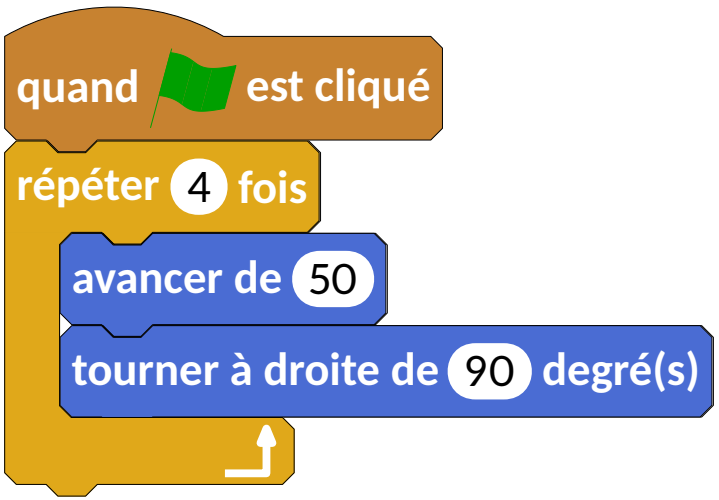
Pourquoi MP-Scratch ?

- Pourquoi pas ? :)
- Basé sur METAPOST
- Introduction de l'algorithmique au collège.
- Capture d'écran



Pourquoi MP-Scratch ?

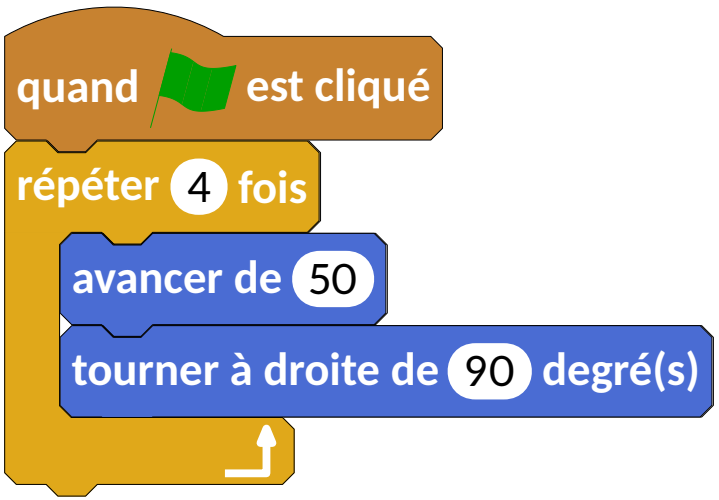
- Pourquoi pas ? :)
- Basé sur METAPOST
- Introduction de l'algorithmique au collège.
- Capture d'écran



- Typographie, qualité et homogénéïté des documents.

Pourquoi MP-Scratch ?

- Pourquoi pas ? :)
- Basé sur METAPOST
- Introduction de l'algorithmique au collège.
- Capture d'écran



- Typographie, qualité et homogénéïté des documents.
- Algorithmique débranché

Autres packages disponibles sous LaTeX

- basé sur Ti*k*Z :

Autres packages disponibles sous LaTeX

- basé sur Ti*k*Z :
  - scratch : `https://ctan.org/pkg/scratch`

Autres packages disponibles sous LaTeX

- basé sur Ti*k*Z :
  - scratch : `https://ctan.org/pkg/scratch`
  - scratchx : `https://www.ctan.org/pkg/scratchx`

Autres packages disponibles sous LaTeX

- basé sur Ti*k*Z :
  - scratch : `https://ctan.org/pkg/scratch`
  - scratchx : `https://www.ctan.org/pkg/scratchx`
  - tikzcodeblocks : `https://www.ctan.org/pkg/tikzcodeblocks`

Autres packages disponibles sous LaTeX

- basé sur Ti*k*Z :
  - scratch : `https://ctan.org/pkg/scratch`
  - scratchx : `https://www.ctan.org/pkg/scratchx`
  - tikzcodeblocks : `https://www.ctan.org/pkg/tikzcodeblocks`
- basé sur pstricks :

Autres packages disponibles sous LaTeX

- basé sur Ti*k*Z :
  - scratch : `https://ctan.org/pkg/scratch`
  - scratchx : `https://www.ctan.org/pkg/scratchx`
  - tikzcodeblocks : `https://www.ctan.org/pkg/tikzcodeblocks`
- basé sur pstricks :
  - scratchTeX : `https://github.com/nicolaspoulain/scratchTeX`

Installation

- Récupérer l'archive.

  `http://melusine.eu.org/syracuse/G/mp-scratch/`

Installation

- Récupérer l'archive.

  `http://melusine.eu.org/syracuse/G/mp-scratch/`

- Vérification de la disponibilité des packages particuliers.

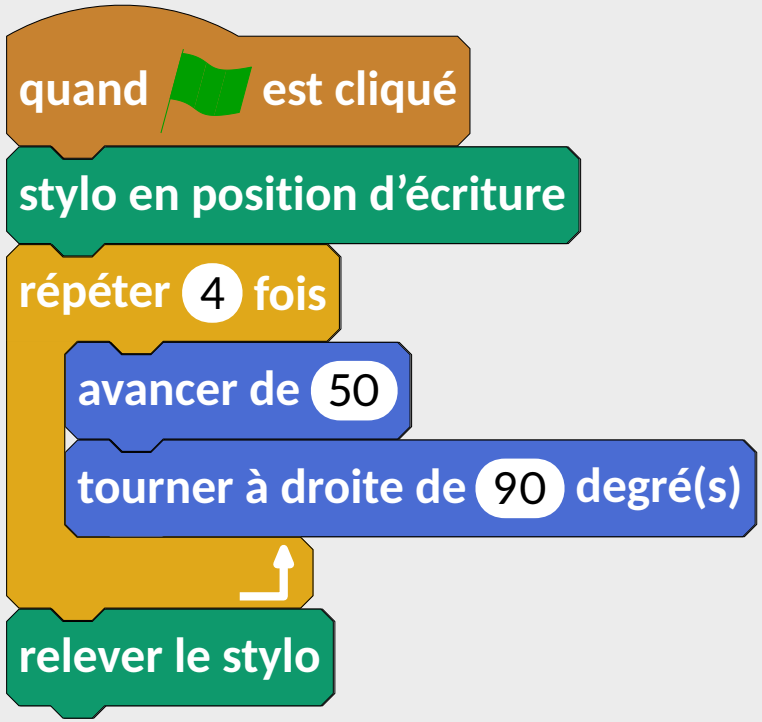
Installation

- Récupérer l'archive.

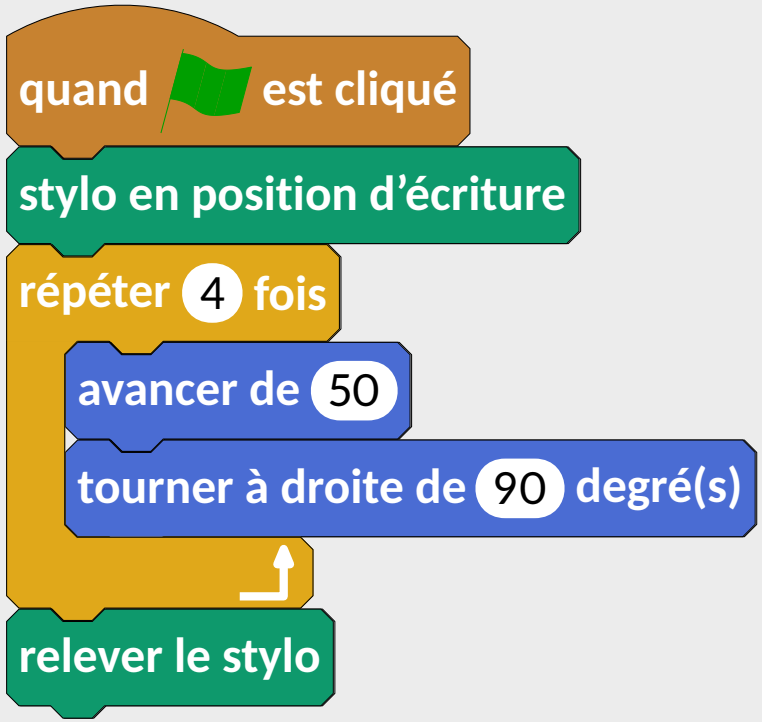
  `http://melusine.eu.org/syracuse/G/mp-scratch/`

- Vérification de la disponibilité des packages particuliers.
- Installation dans *un répertoire local*.

À vous de jouer

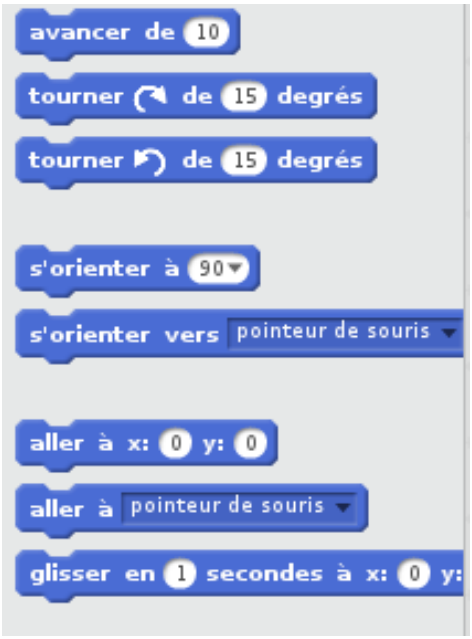


À vous de jouer

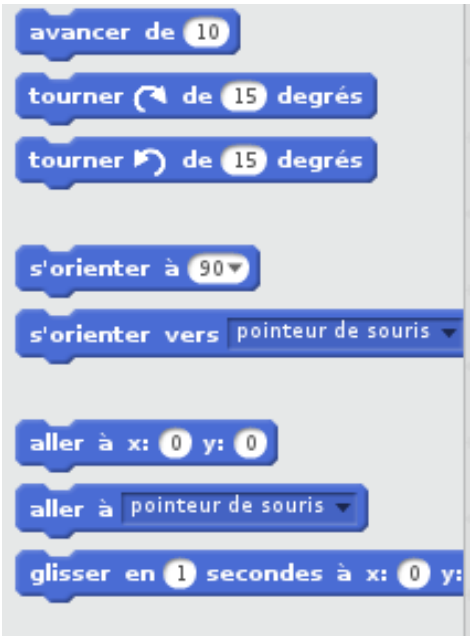



```
draw Drapeau;
draw PoserStylo;
draw Repeter("4");
draw Avancer("50");
draw Tournerd("90");
draw FinBlocRepeter;
draw ReleverStylo;
```

Groupe « Mouvement »



Groupe « Mouvement »



- **draw** Avancer("10");
- **draw** Tournerd("90");
- **draw** Tournerg("90");
- **draw** Orienter("90");

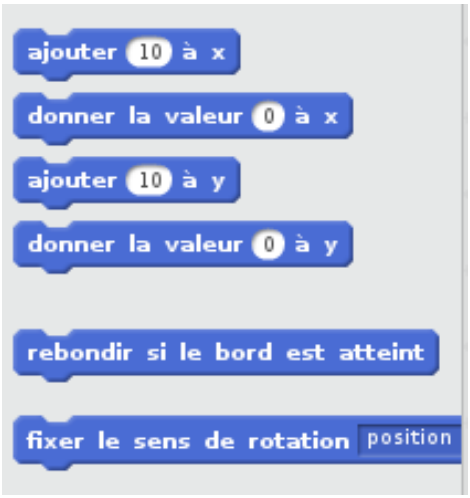
- **draw** Orienterdirection("pointeur_de_souris");

- **draw** Aller("50","100");

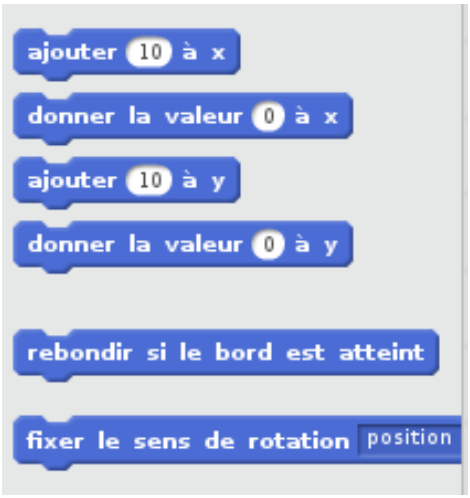
- **draw** Allera("pointeur_de_souris");

- **draw** Glisser("2","50","100");

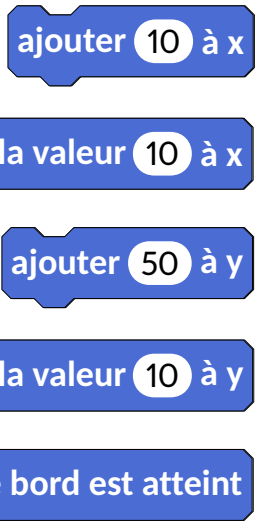
Groupe « Mouvement »



Groupe « Mouvement »

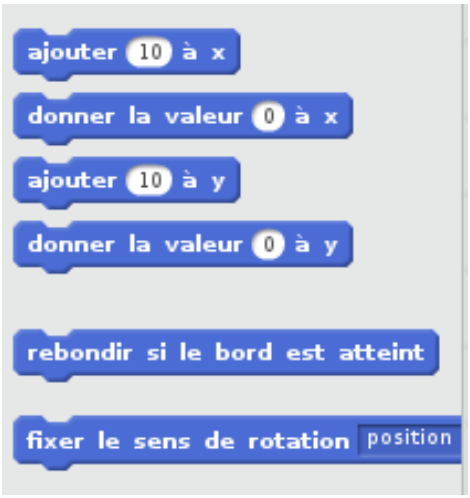


- **draw** Ajouter("10","x");
- **draw** Mettre("10","x");
- **draw** Ajouter("50","y");
- **draw** Mettre("10","y");
- **draw** Rebondir;



- **draw** FixerSensRotation("position␣\'a␣gauche␣ou␣\'a␣droite");

Groupe « Mouvement »



- **draw** Ajouter("10","x");

- **draw** Mettre("10","x");

- **draw** Ajouter("50","y");

- **draw** Mettre("10","y");

- **draw** Rebondir;

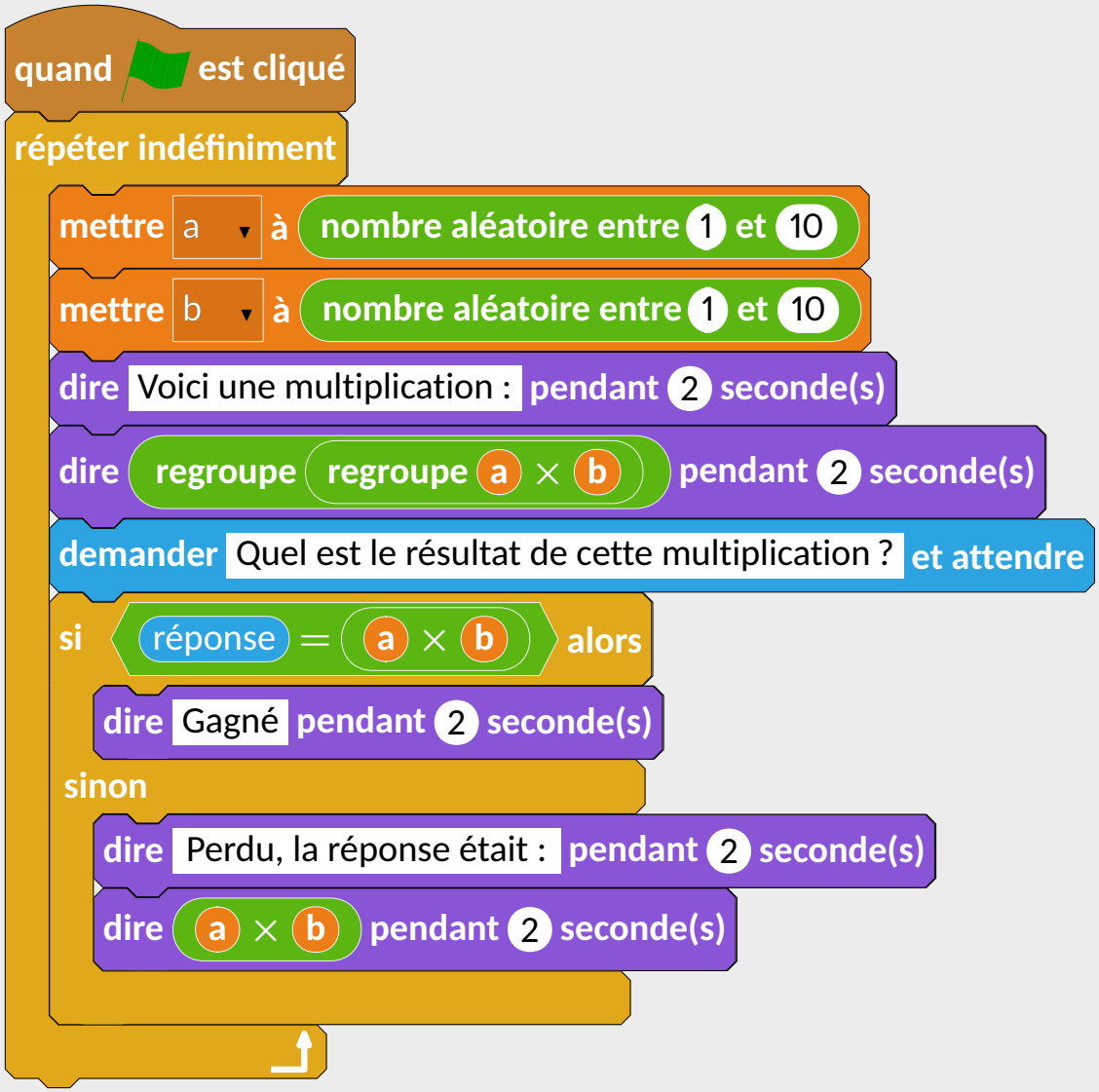
- **draw** FixerSensRotation("position_\‘a_gauche_ou_\‘a_droite");

- **draw** Avancer("\opOp{$\opMouv{abscisse\_x}\bm{+}\opSimple{10}$}");

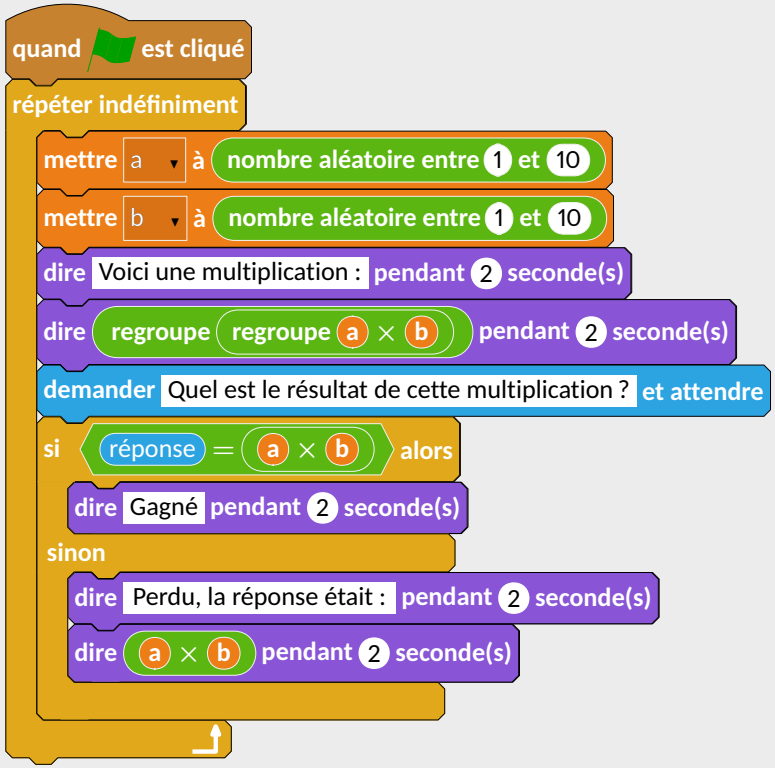
- **draw** Mettre("\opMouv{abscisse_x}","y");

- **draw** Ajouter("\opMouv{direction}","y");

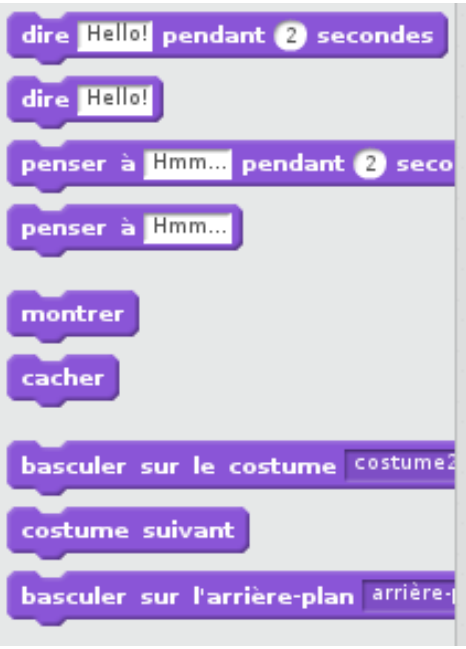
À vous de jouer !



```
draw Drapeau;
draw RepeterI;
picture BB[];
BB1=OvalOp("nombre aléatoire entre ",OvalNb("1"),"
    et ",OvalNb("10"));
draw MettreVar("a",BB1);
draw MettreVar("b",BB1);
draw DireT("Voici une multiplication :","2");
BB2=OvalOp("regroupe",OvalVar("a"),"
    $\bm{\times}$ ",OvalVar("b"));
draw DireT(OvalOp("regroupe",BB2),"2");
draw Demander("Quel est le résultat de cette
    multiplication ?");
BB3=OvalOp(OvalVar("a")," $\bm{\times}$
    ",OvalVar("b"));
BB4=TestOp(OvalCap("réponse"),"$\bm{=}$",BB3);
draw Si(BB4);
draw DireT("Gagné","2");
draw Sinon;
draw DireT("Perdu, la réponse était : ","2");
draw DireT(BB3,"2");
draw FinBlocSi;
draw FinBlocRepeter;
```



Groupe « Apparence »



Groupe « Apparence »

- **draw** DireT("Hello",2);

dire Hello pendant 2 seconde(s)

- **draw** Dire("Hello");

dire Hello

- **draw** PenserT("Hmm...","2");

penser Hmm... pendant 2 seconde(s)

- **draw** Penser("Hmm...");

penser Hmm...

- **draw** Montrer;

montrer

- **draw** Cacher;

cacher

- **draw** Basculer("costume2");

basculer sur le costume costume2

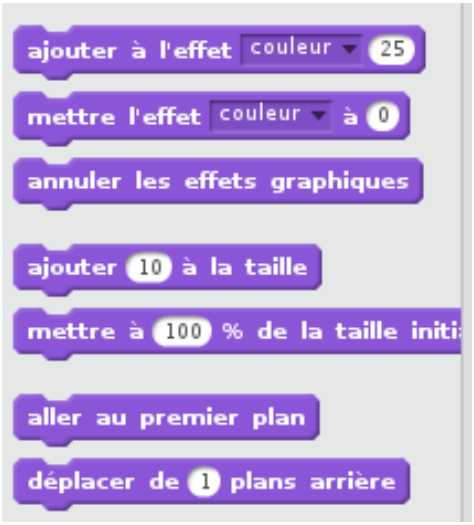
- **draw** CostumeSuivant;

costume suivant

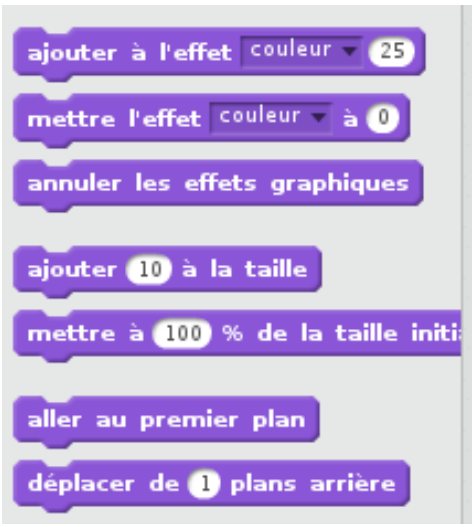
- **draw** BasculerAR("arriere−plan2");

basculer sur l'arrière-plan arrière-plan2

Groupe « Apparence »

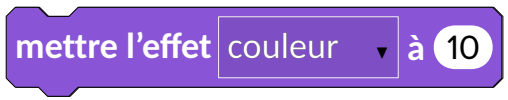


Groupe « Apparence »



- **draw** AjouterEffet("couleur","10");

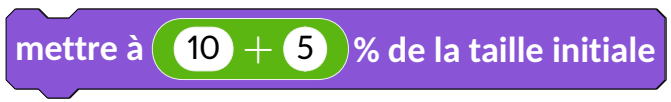
- **draw** MettreEffet("couleur","10");



- **draw** AnnulerEffet;

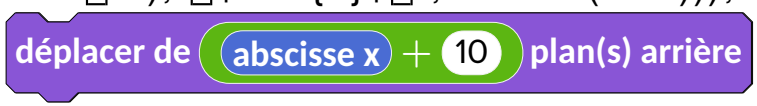
- **draw** AjouterTaille("10");

- **draw** MettreA(OvalOp(OvalNb("10"),"$\bm{+}$",OvalNb("5")));



- **draw** AllerPPlan;

- **draw** DeplacerAP(OvalOp(OvalMouv("abscisse␣x"),"␣$\bm{+}$␣",OvalNb("10")));



Groupe « Apparence »

Quand « la scène » est sélectionnée, on dispose également des commandes :

- **draw** BasculerARA("arriere−plan2");

- **draw** ARSuivant;

Groupe « Apparence »

Quand « la scène » est sélectionnée, on dispose également des commandes :

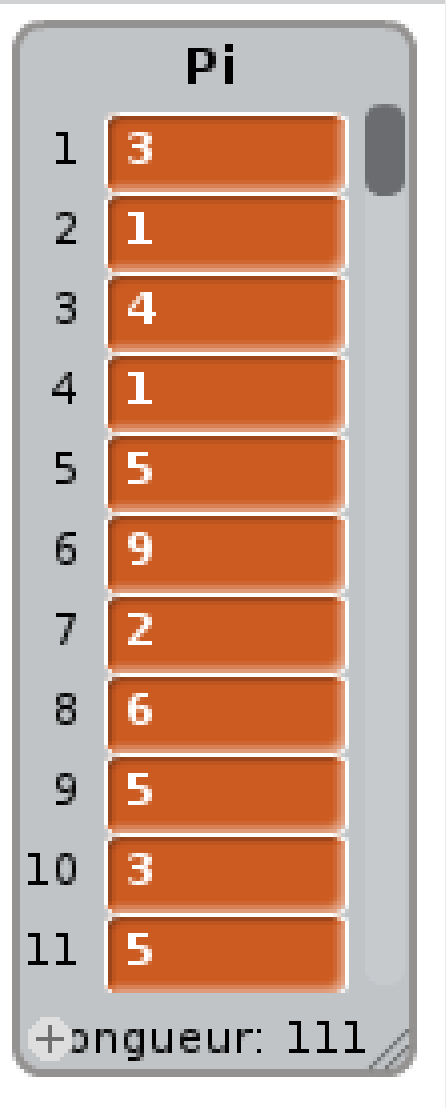
- **draw** BasculerARA("arriere−plan2");

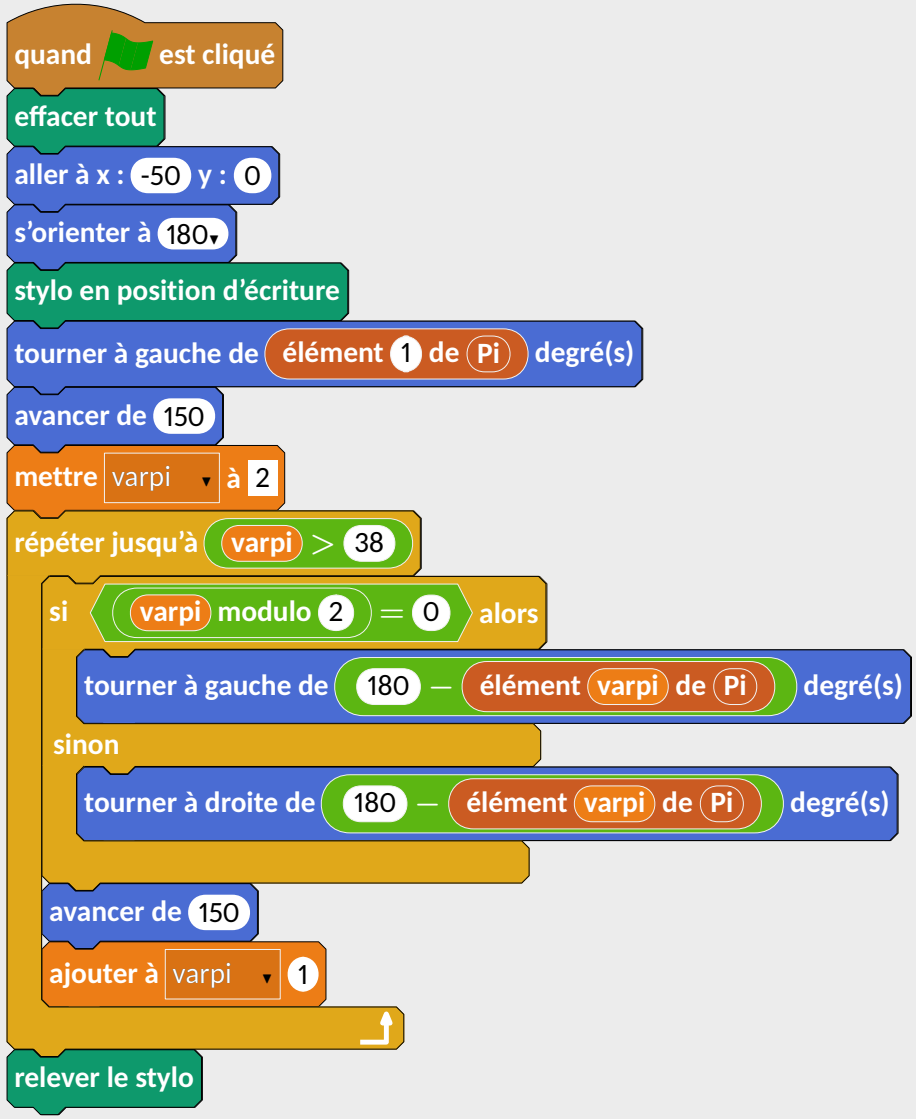
- **draw** ARSuivant;

Les « opérateurs »

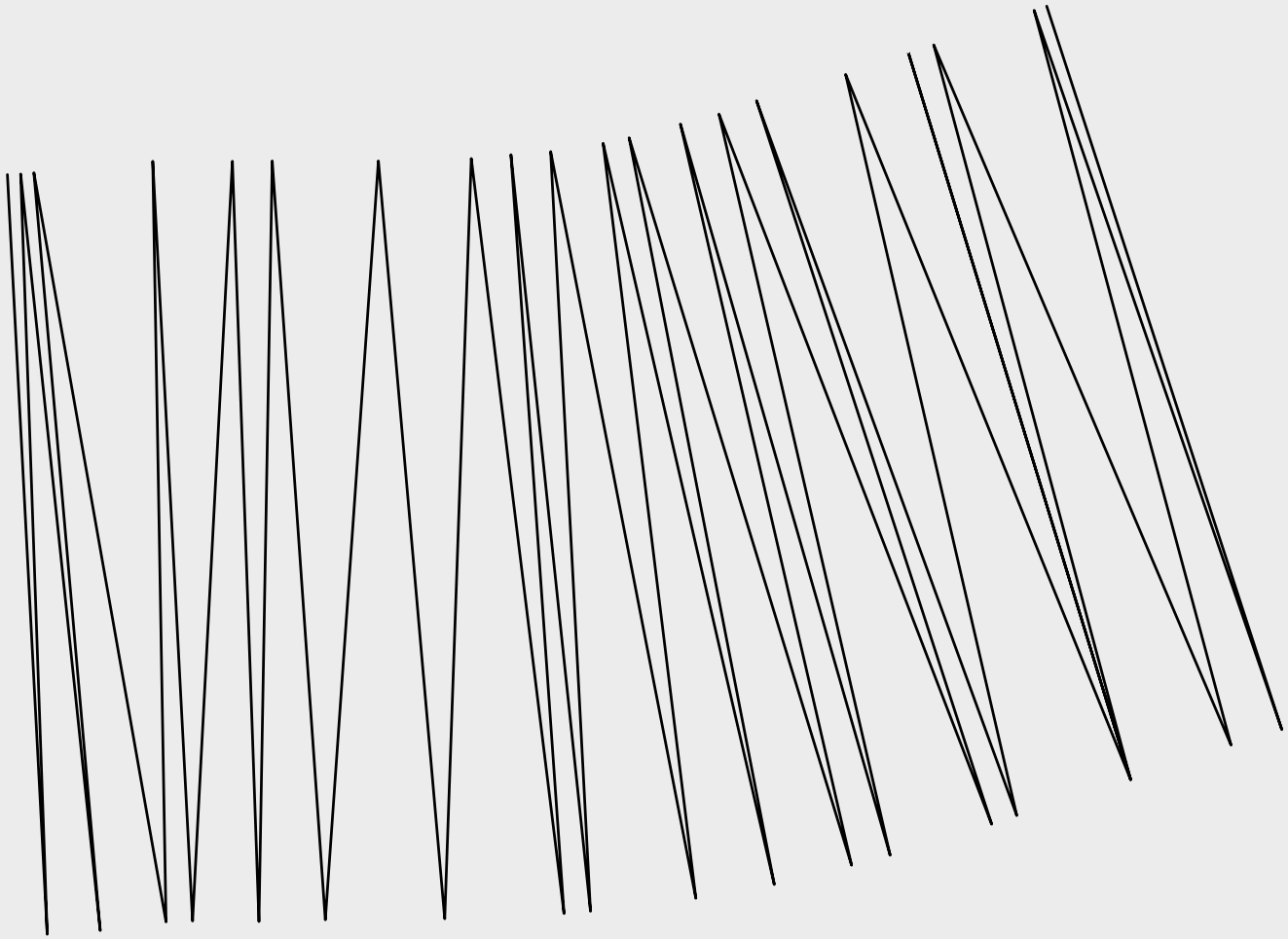
s'obtiennent avec la commande OvalApp().

À vous de jouer !

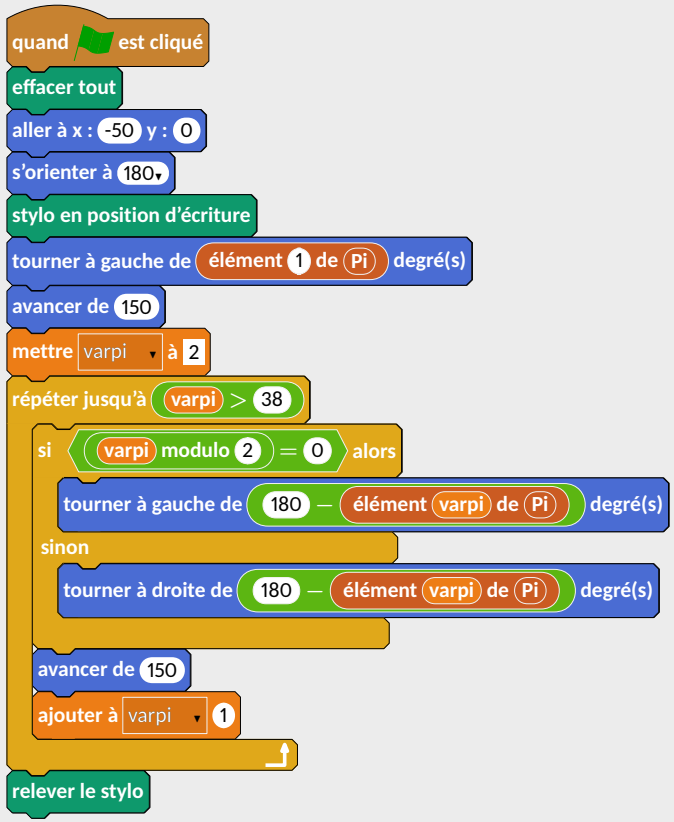




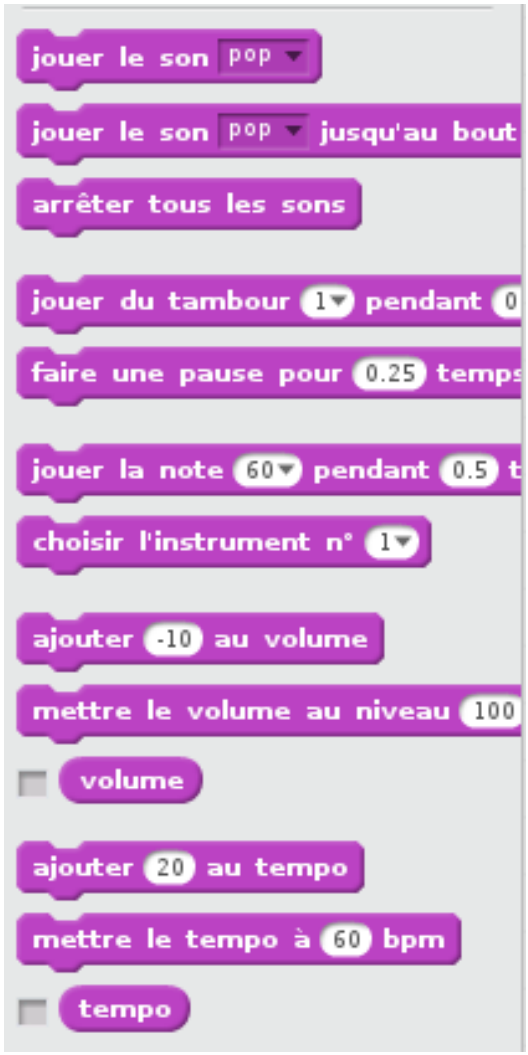
À vous de jouer !



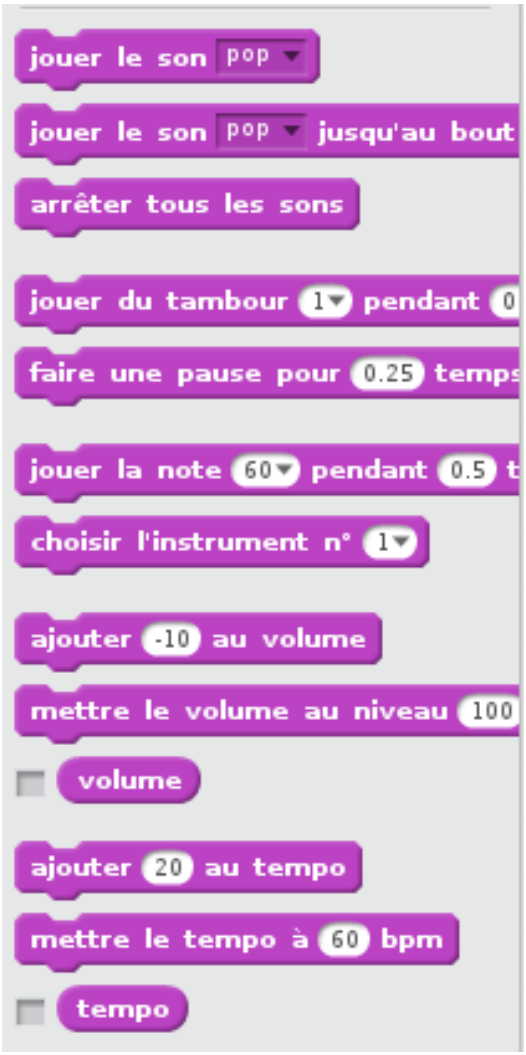
```
draw Drapeau;
draw Effacer;
draw Aller("-50","0");
draw Orienter("180");
draw PoserStylo;
picture BB[];
BB1=OvalListMulti("élément ",OvalNb("1")," de ",
    OvalList("Pi"));
draw Tournerg(BB1);
draw Avancer("150");
draw MettreVar("varpi","2");
draw RepeterJ(OvalOp(OvalVar("varpi")," $\bm{>}$ ",
      OvalNb("38")));
BB2=OvalOp(OvalVar("varpi")," modulo ",OvalNb("2"));
BB3=TestOp(BB2," $\bm{=}$ ",OvalNb("0"));
draw Si(BB3);
BB4=OvalListMulti("élément ",OvalVar("varpi")," de ",
OvalList("Pi"));
draw Tournerg(OvalOp(OvalNb("180")," $\bm{-}$ ",BB4));
draw Sinon;
draw Tournerd(OvalOp(OvalNb("180")," $\bm{-}$ ",BB4));
draw FinBlocSi;
draw Avancer("150");
draw AjouterVar("varpi","1");
draw FinBlocRepeter;
draw ReleverStylo;
```



Groupe « Son »


jouer le son pop
jouer le son pop jusqu'au bout
arrêter tous les sons
jouer du tambour 1 pendant 0
faire une pause pour 0.25 temps
jouer la note 60 pendant 0.5 t
choisir l'instrument n° 1
ajouter -10 au volume
mettre le volume au niveau 100
volume
ajouter 20 au tempo
mettre le tempo à 60 bpm
tempo

Groupe « Son »



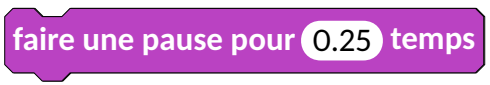
- **draw** Jouer("miaou");

- **draw** JouerT("miaou");

- **draw** ArreterSon;

- **draw** Tambour("2","0.25");

- **draw** Pause("0.25");

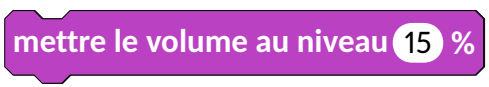


- **draw** JouerNote("50","0.25");

- **draw** ChoisirInstrument("17");

- **draw** AjouterVol("−10");

- **draw** MettreVol("15");



- **draw** AjouterTempo("20");

- **draw** MettreTempo("15");

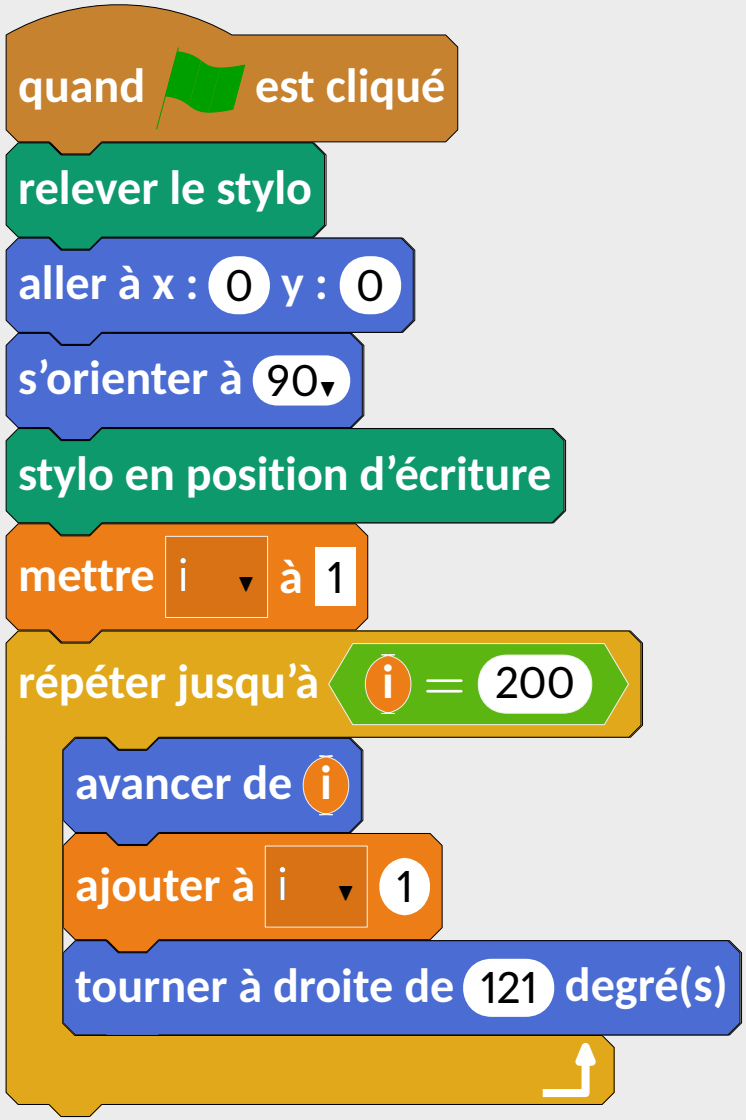
Les « opérateurs »

volume tempo

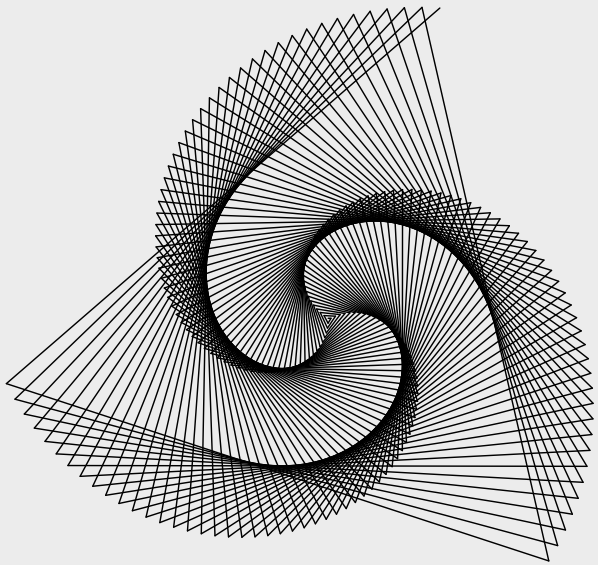
s'obtiennent par la commande OvalSon().

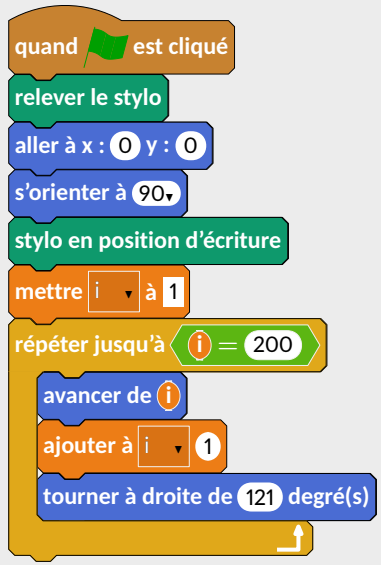

```
draw DireT(OvalOp("regroupe",RecText("Le volume actuel est
    "),OvalSon("Volume")),"2");
```

À vous de jouer !



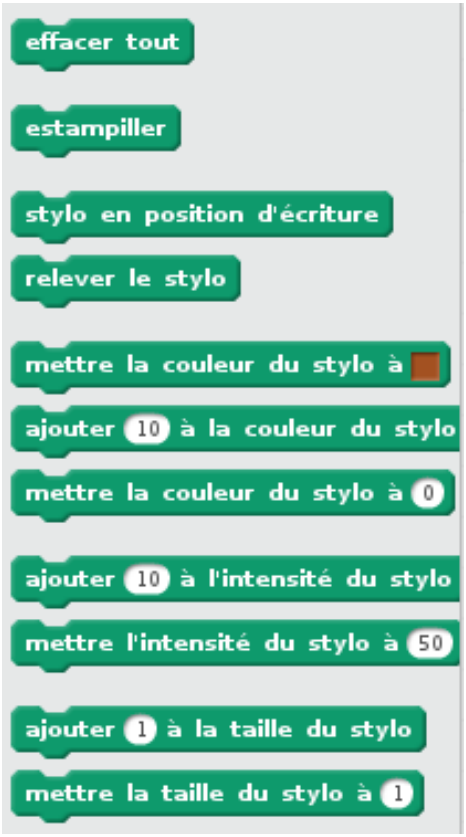
À vous de jouer !



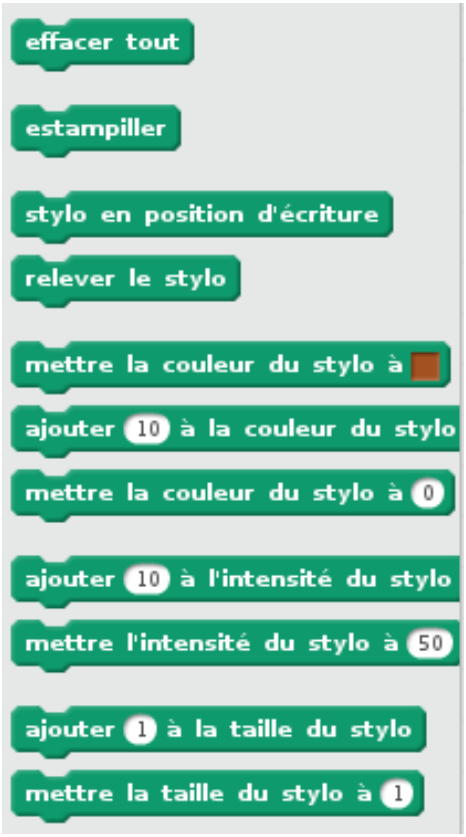



```
draw Drapeau;
draw ReleverStylo;
draw Aller("0","0");
draw Orienter("90");
draw PoserStylo;
draw MettreVar("i","1");
draw RepeterJ(TestOp(OvalVar("i"),"␣$\bm{=}␣$",OvalNb("200")));
draw Avancer(OvalVar("i"));
draw AjouterVar("i","1");
draw Tournerd("121");
draw FinBlocRepeter;
```

Groupe « Stylo »



Groupe « Stylo »



- **draw** Effacer;

- **draw** Estampiller;

- **draw** PoserStylo;

- **draw** releverStylo;

- **draw** MettreCouleur("Magenta",1,0,1);

- **draw** MettreCS("25");

- **draw** MettreIS("15");

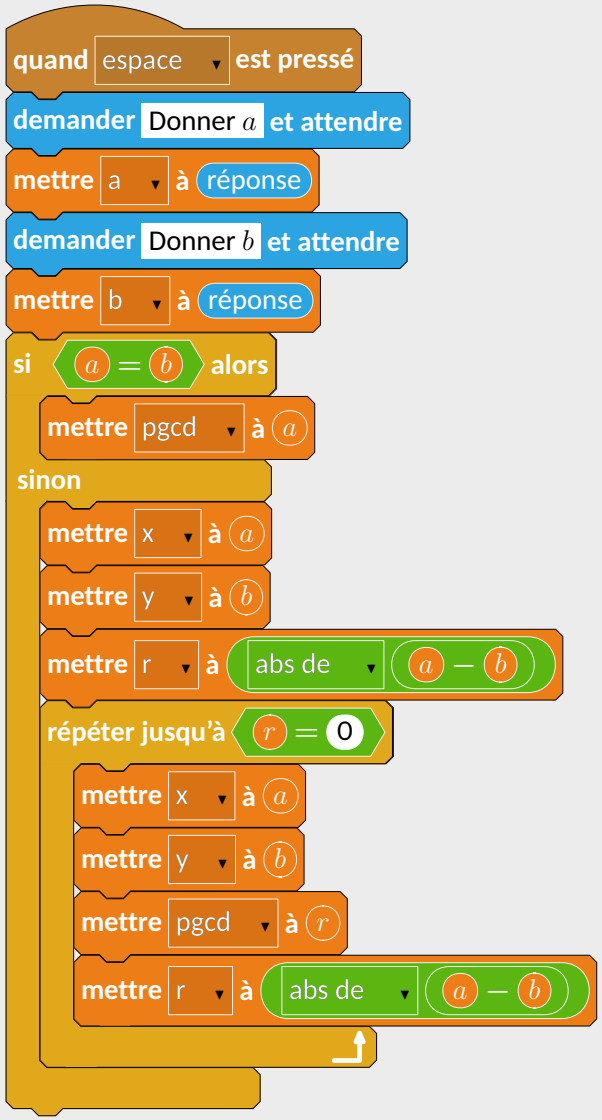
- **draw** AjouterTS("12");

- **draw** AjouterCS("15");

- **draw** AjouterIS(OvalOp(OvalNb("15"),"$\bm{−}$",OvalNb("10")));

- **draw** MettreTS(OvalOp(OvalNb("15"),"$\bm{\times}$",OvalNb("10")));

À vous de jouer !



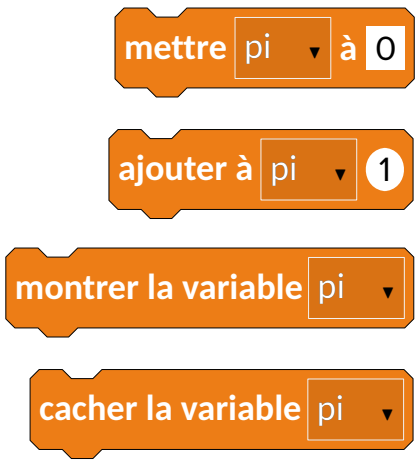
À vous de jouer !

```
draw QPresse("espace");
draw Demander("Donner␣$a$");
draw MettreVar("a",OvalCap("réponse"));
draw Demander("Donner␣$b$");
draw MettreVar("b",OvalCap("réponse"));
draw Si(TestOp(OvalVar("$a$"),"$\bm{=}$",OvalVar("$b$")));
draw MettreVar("pgcd",OvalVar("$a$"));
draw Sinon;
draw MettreVar("x",OvalVar("$a$"));
draw MettreVar("y",OvalVar("$b$"));
picture BB[];
BB1=OvalOp(OvalVar("$a$"),"$\bm{-}$",OvalVar("$b$"));
draw MettreVar("r",OvalOp(RecMenuOp("abs de "),BB1));
draw RepeterJ(TestOp(OvalVar("$r$"),"$\bm{=}$",OvalNb("0")));
draw MettreVar("x",OvalVar("$a$"));
draw MettreVar("y",OvalVar("$b$"));
draw MettreVar("pgcd",OvalVar("$r$"));
draw MettreVar("r",OvalOp(RecMenuOp("abs de "),BB1));
draw FinBlocRepeter;
draw FinBlocSi;
```

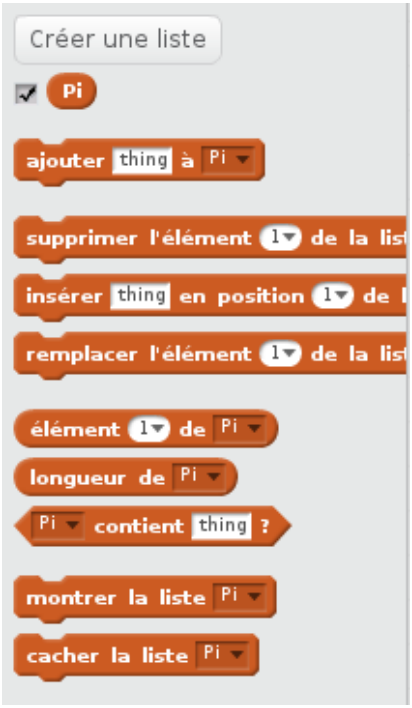
Groupe « Données »- Variables

Groupe « Données »- Variables

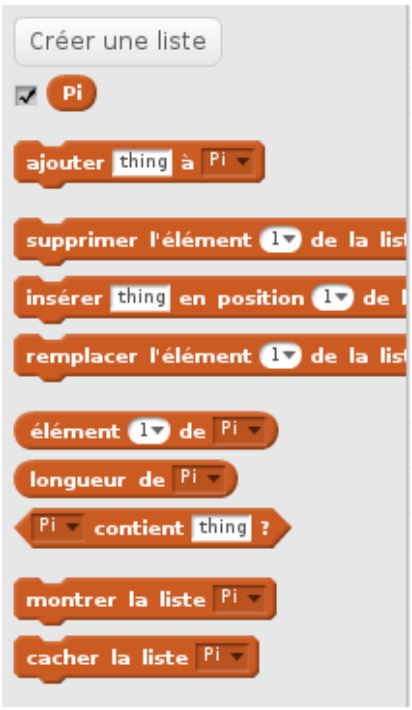
- **draw** MettreVar("pi","0") ;
- **draw** AjouterVar("pi","1") ;
- **draw** MontrerVar("pi");
- **draw** CacherVar("pi");



Groupe « Données »- Listes



Groupe « Données »- Listes



- **draw** AjouterList(RecText("\LaTeX"),"Listepi") ;

- **draw** SupprimerList(RecText("\LaTeX"),"Listepi");

- **draw** InsererList(RecText("\MP"),"1"," Listepi ") ;

- **draw** RemplacerList(3,"Listepi",OvalOp(OvalNb("4")," $\bm{+}$",OvalNb("5")));

- **draw** MontrerList("Listepi") ;

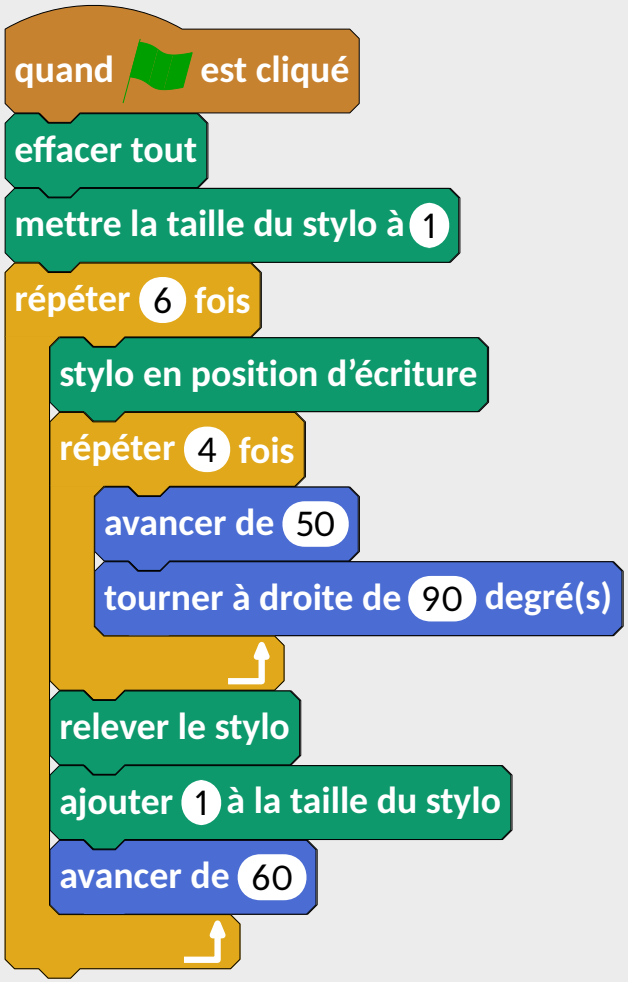
- **draw** CacherList("Listepi") ;

Les « opérateurs »

s'obtiennent par les commandes \OvalVar() et \OvalListMulti().

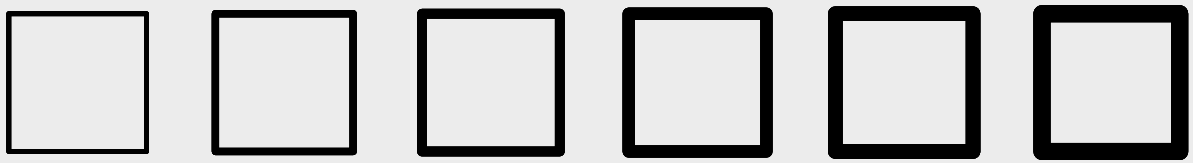

```
draw OvalListMulti(RecMenuList("Pythagore"),"␣
    contient␣",RecText("(3;4;5)␣?"));
```

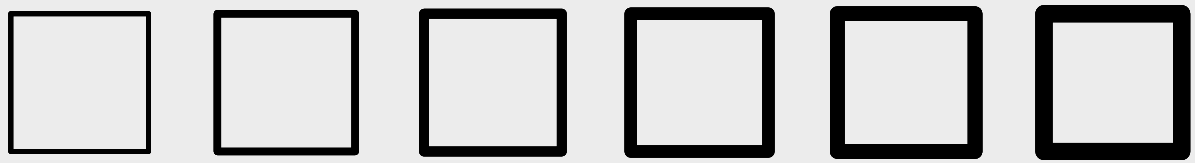
À vous de jouer !




quand est cliqué
effacer tout
mettre la taille du stylo à 1
répéter 6 fois
stylo en position d'écriture
répéter 4 fois
avancer de 50
tourner à droite de 90 degré(s)
relever le stylo
ajouter 1 à la taille du stylo
avancer de 60


quand est cliqué
effacer tout
mettre la taille du stylo à 1
répéter 6 fois
stylo en position d'écriture
répéter 4 fois
avancer de 50
tourner à droite de 90 degré(s)
relever le stylo
ajouter 1 à la taille du stylo
avancer de 60





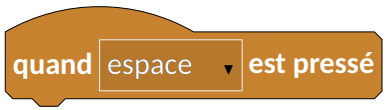
```
draw Drapeau;
draw Effacer;
draw MettreTS("1");
draw Repeter("6");
draw PoserStylo;
draw Repeter("4");
draw Avancer("50");
draw Tournerd("90");
draw FinBlocRepeter;
draw ReleverStylo;
draw AjouterTS("1");
draw Avancer("60");
draw FinBlocRepeter;
```

Groupe « Évènements »

Groupe « Évènements »

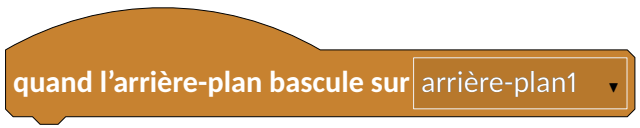
- **draw** Drapeau;

- **draw** QPresse("espace");



- **draw** QLutinPresse;

- **draw** QBasculeAR("arriere−plan1");



- **draw** QVolumeSup("Volume␣sonore",OvalNb("10"));

- **draw** QRecevoirMessage("message1");

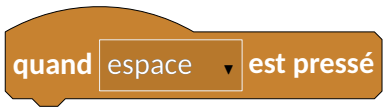
- **draw** EnvoyerMessage("message1");

- **draw** EnvoyerMessageA("message1");

Groupe « Évènements »

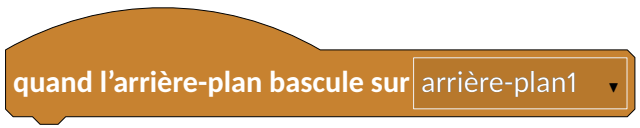
- **draw** Drapeau;

- **draw** QPresse("espace");



- **draw** QLutinPresse;

- **draw** QBasculeAR("arriere−plan1");



- **draw** QVolumeSup("Volume␣sonore",OvalNb("10"));

- **draw** QRecevoirMessage("message1");

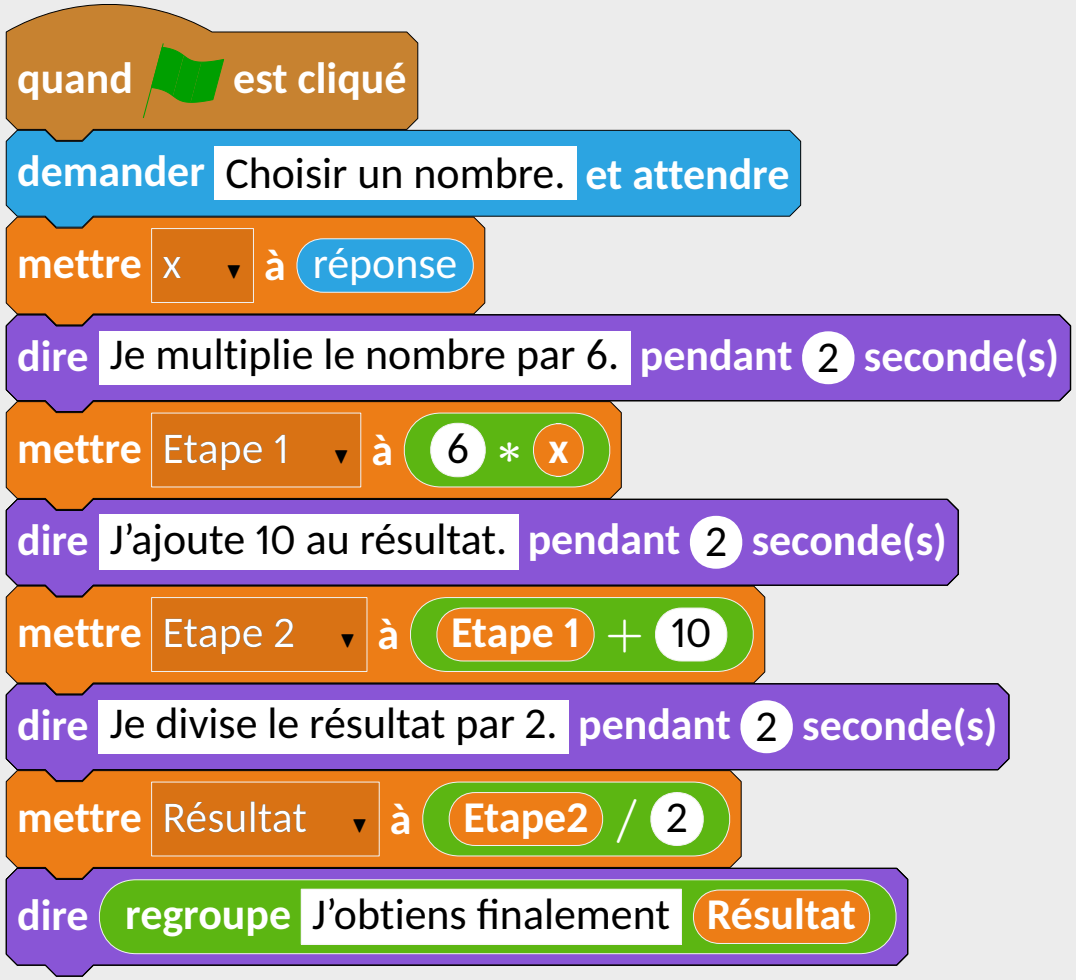
- **draw** EnvoyerMessage("message1");

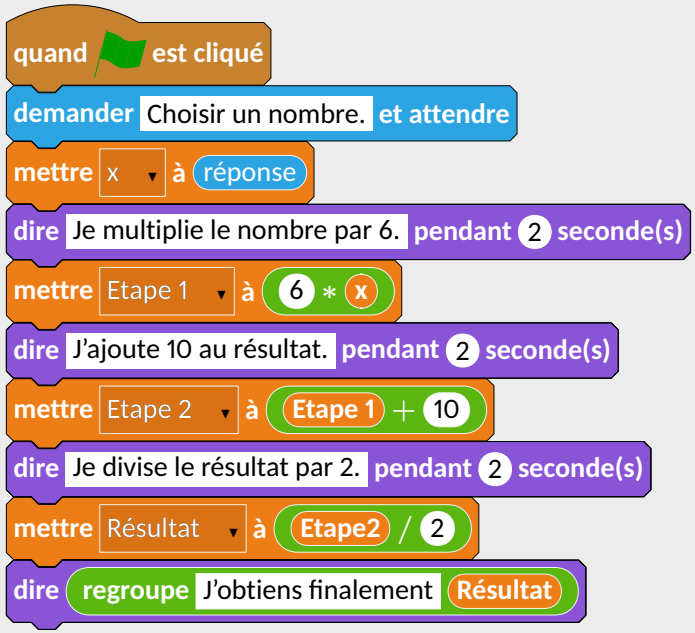
- **draw** EnvoyerMessageA("message1");

Quand « la scène » est sélectionnée, on dispose de :

- **draw** QScenePressee;

À vous de jouer !






```
draw Drapeau;
draw Demander("Choisir un nombre.");
draw MettreVar("x",OvalCap("réponse"));
draw DireT("Je multiplie le nombre par 6.","2");
draw MettreVar("Etape 1",OvalOp(OvalNb("6"),"$\bm{*}$",OvalVar("x")));
draw DireT("J'ajoute 10 au résultat.","2");
draw MettreVar("Etape 2",OvalOp(OvalVar("Etape 1"),"$\bm{+}$",OvalNb("10")));
draw DireT("Je divise le résultat par 2.","2");
draw MettreVar("Résultat",OvalOp(OvalVar("Etape2"),"$\bm{/}$",OvalNb("2")));
draw Dire(OvalOp("regroupe",RecText("J'obtiens finalement"),OvalVar("Résultat")));
```

Groupe « Contrôle »

Groupe « Contrôle »

- **draw** Attendre("1");

  draw Repeter(10);
- **draw** LigneVide;
  draw FinBlocRepeter;

  draw RepeterI;
- **draw** LigneVide;
  draw FinBlocRepeterI;

Groupe « Contrôle »

- **draw** Attendre("1");

  draw Repeter(10);
- **draw** LigneVide;
  draw FinBlocRepeter;

  draw RepeterI;
- **draw** LigneVide;
  draw FinBlocRepeterI;

- **draw** Si(TestOp(OvalVar("x"),
  "␣$\bm{=}$␣",OvalNb("20
  ")));
  draw LigneVide;
  draw FinBlocSi;

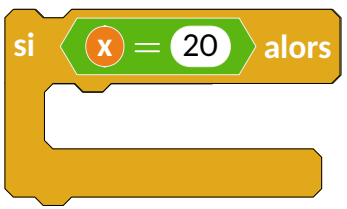
Groupe « Contrôle »

- **draw** Attendre("1");

draw Repeter(10);
- **draw** LigneVide;
  draw FinBlocRepeter;

draw RepeterI;
- **draw** LigneVide;
  draw FinBlocRepeterI;

- **draw** Si(TestOp(OvalVar("x"),
  "␣$\bm{=}$␣",OvalNb("20
  ")));
  draw LigneVide;
  draw FinBlocSi;



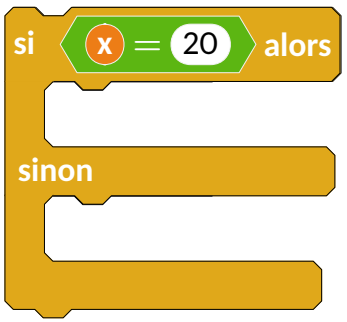
Groupe « Contrôle »

- **draw** Attendre("1");

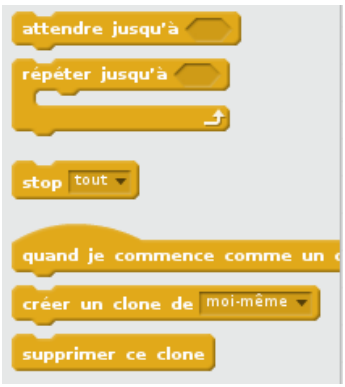
  draw Repeter(10);
- **draw** LigneVide;
  draw FinBlocRepeter;

  draw RepeterI;
- **draw** LigneVide;
  draw FinBlocRepeterI;

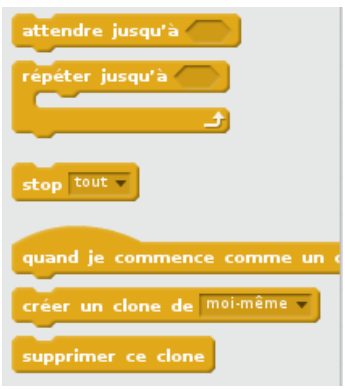
- **draw** Si(TestOp(OvalVar("x"), "␣$\bm{=}$␣",OvalNb("20")));
  draw LigneVide;
  draw Sinon;
  draw LigneVide;
  draw FinBlocSi;



Groupe « Contrôle »

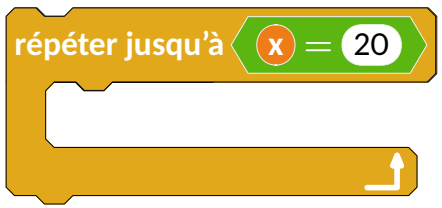


Groupe « Contrôle »



- **draw** AttendreJ(TestOp(OvalVar("x"),"␣$\bm{=}$␣",OvalNb("20")))

- **draw** RepeterJ(TestOp(OvalVar("x"),"␣$\bm{=}$␣","OvalNb("20")));
  draw LigneVide;
  draw FinBlocRepeter;



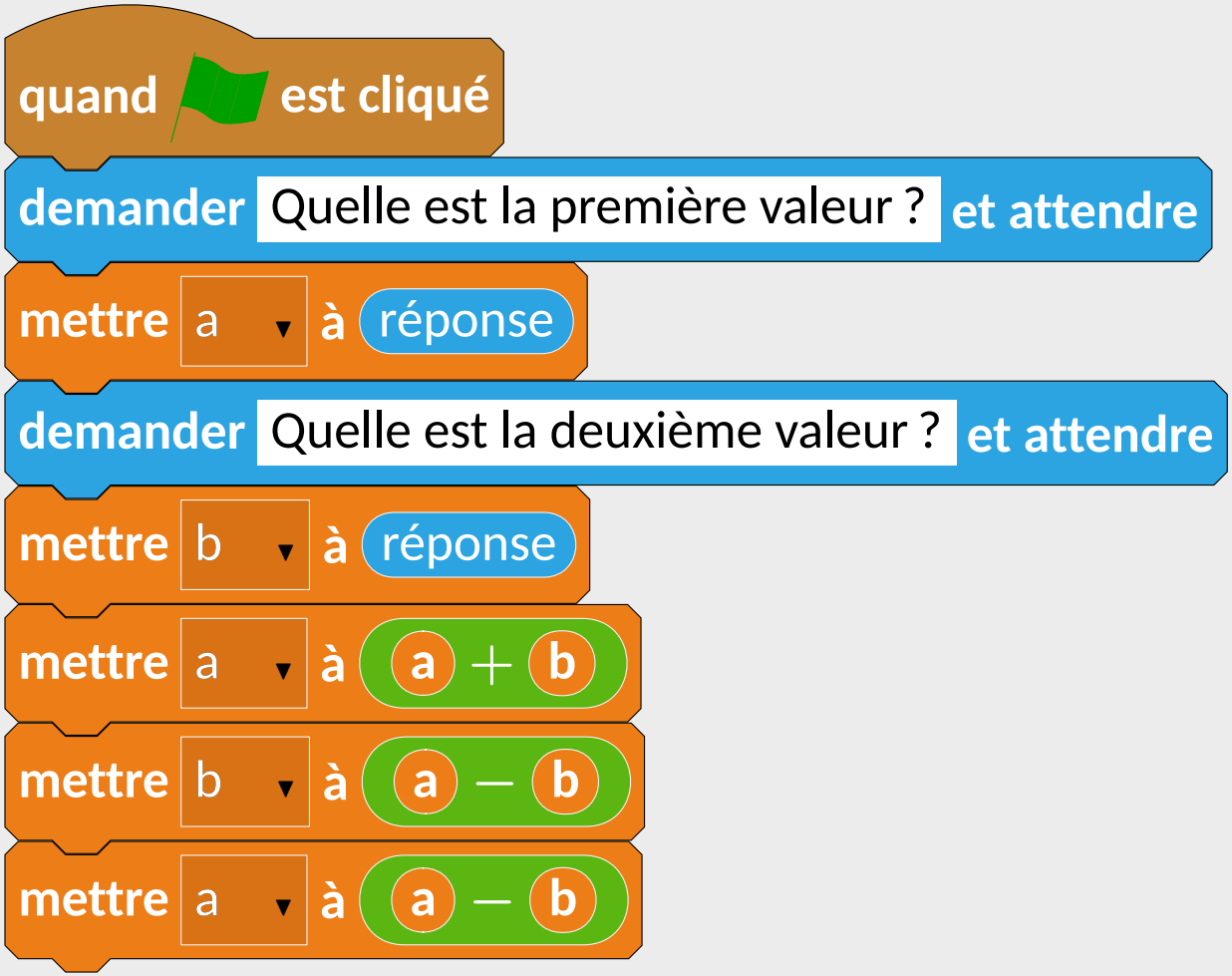
- **draw Stop**("ce␣script");

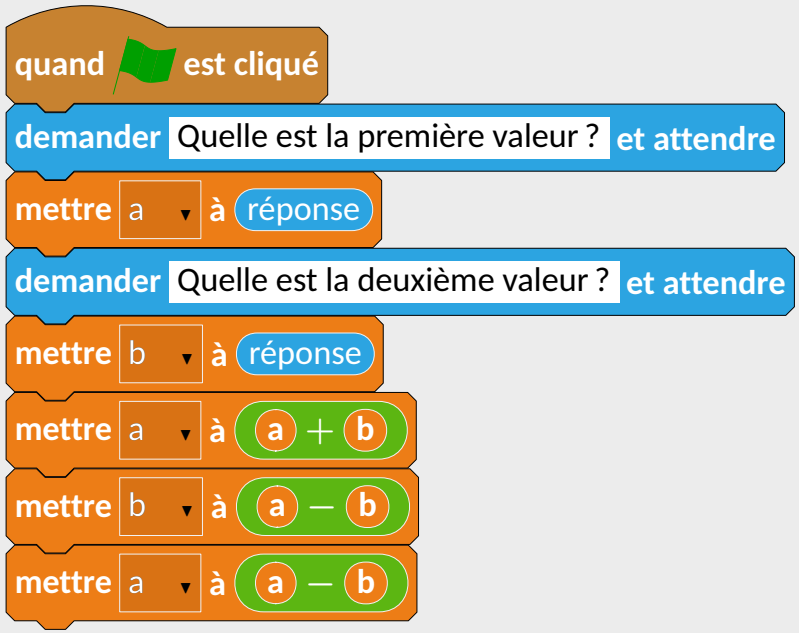
- **draw** CommencerClone;

- **draw** CreerClone("Lutin1");

- **draw** SupprimerClone;

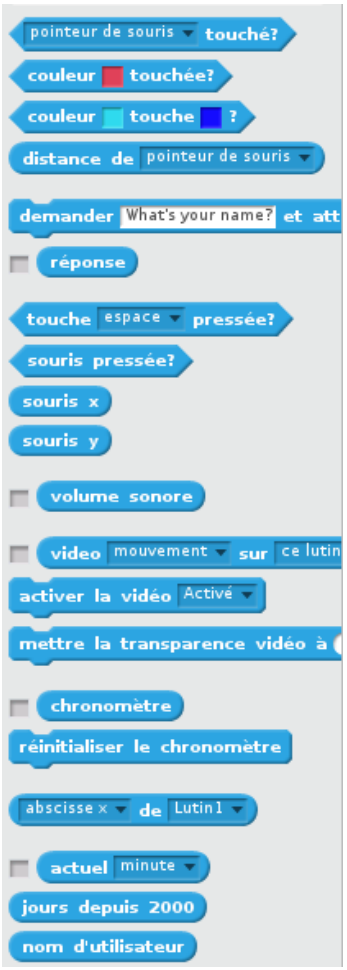
À vous de jouer !



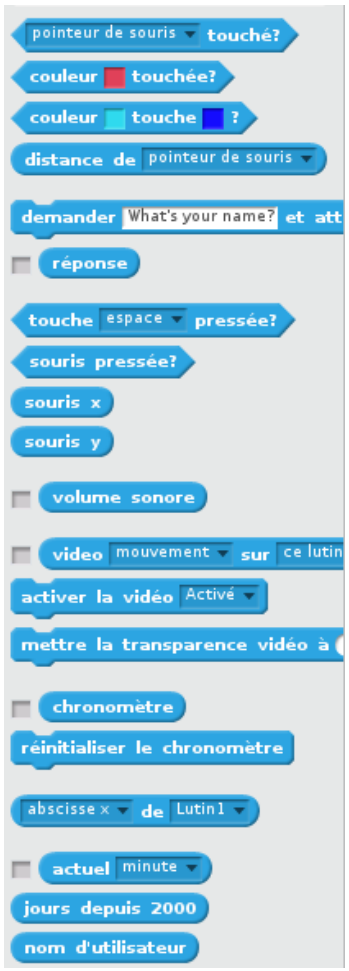



```
draw Drapeau;
draw Demander("Quelle est la première valeur ?");
draw MettreVar("a",OvalCap("réponse"));
draw Demander("Quelle est la deuxième valeur ?");
draw MettreVar("b",OvalCap("réponse"));
picture BB[];
BB1=OvalOp(OvalVar("a"),"$\bm{+}$",OvalVar("b"));
BB2=OvalOp(OvalVar("a"),"$\bm{-}$",OvalVar("b"));
draw MettreVar("a",BB1);
draw MettreVar("b",BB2);
draw MettreVar("a",BB2);
```

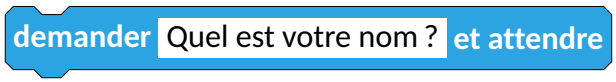
Groupe « Capteurs »



Groupe « Capteurs »



- **draw** Demander("Quel_est_votre_nom_?");

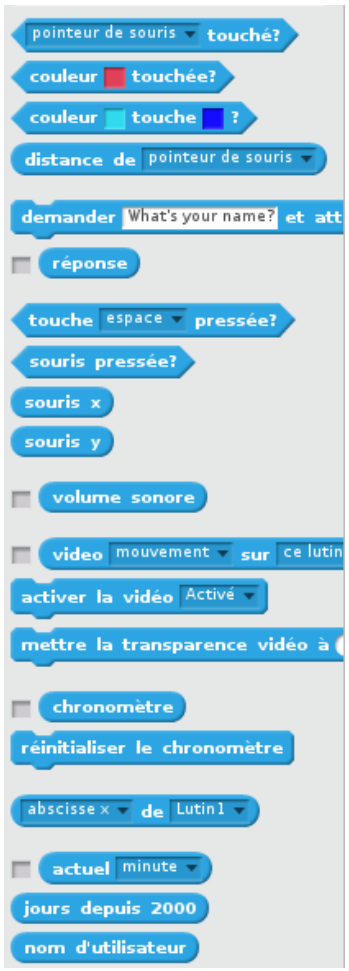


- **draw** ActiverVideo("activ\'e");

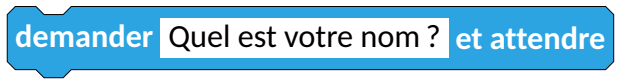
- **draw** TransparenceVideo("15");

- **draw** ReinitChrono;

Groupe « Capteurs »



- **draw** Demander("Quel␣est␣votre␣nom␣?");

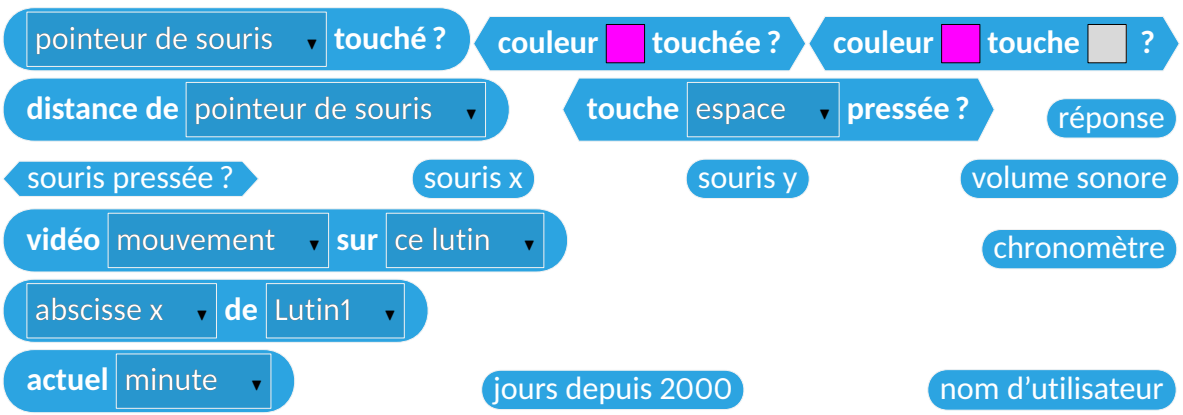


- **draw** ActiverVideo("activ\'e") ;

- **draw** TransparenceVideo("15");

- **draw** ReinitChrono;

Les « opérateurs »

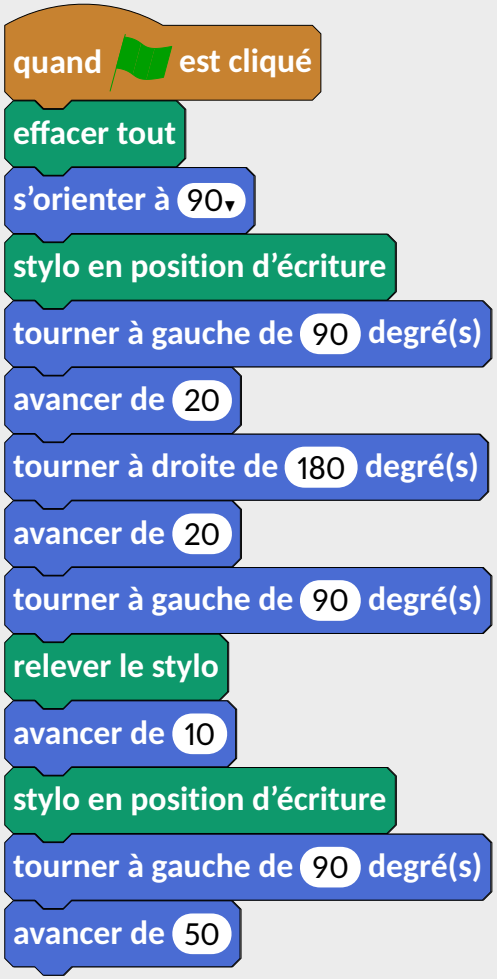


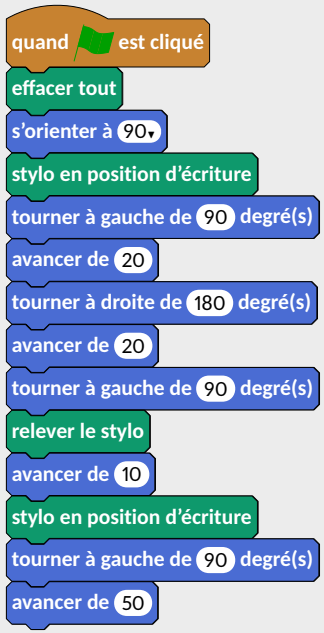
s'obtiennent par les commandes OvalCap et TestCap.

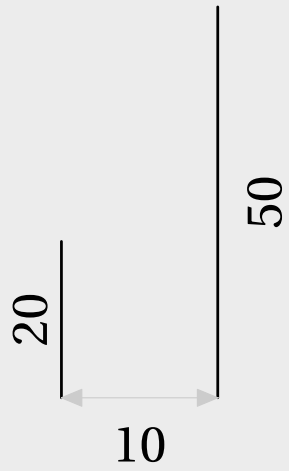

couleur
touche
?


```
draw TestCapMulti("couleur",RecCouleur(1,0,1),"touche",
     RecCouleur(0.85,0.85,0.85),"?");
```

À vous de jouer !





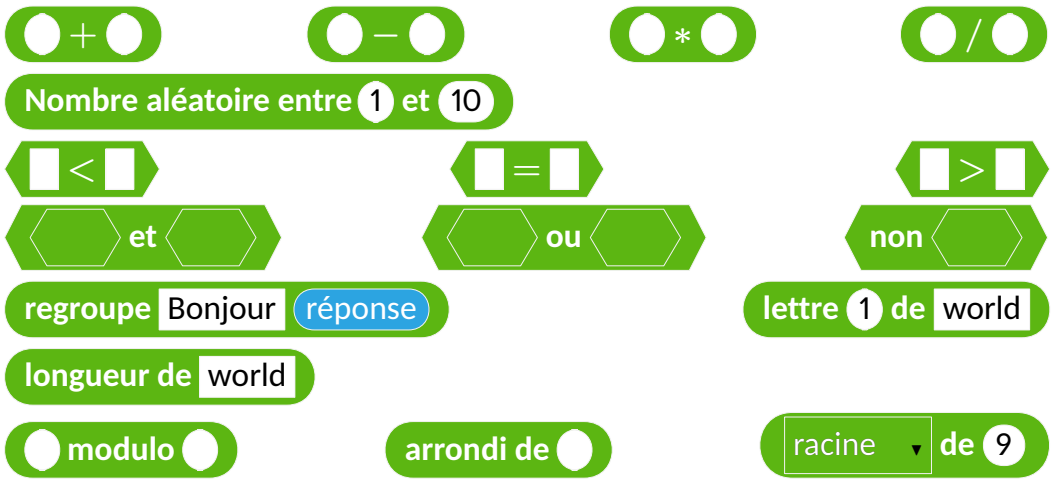



```
draw Drapeau;
draw Effacer;
draw Orienter("90");
draw PoserStylo;
draw Tournerg("90");
draw Avancer("20");
draw Tournerd("180");
draw Avancer("20");
draw Tournerg("90");
draw ReleverStylo;
draw Avancer("10");
draw PoserStylo;
draw Tournerg("90");
draw Avancer("50");
```

Groupe « Opérateurs »

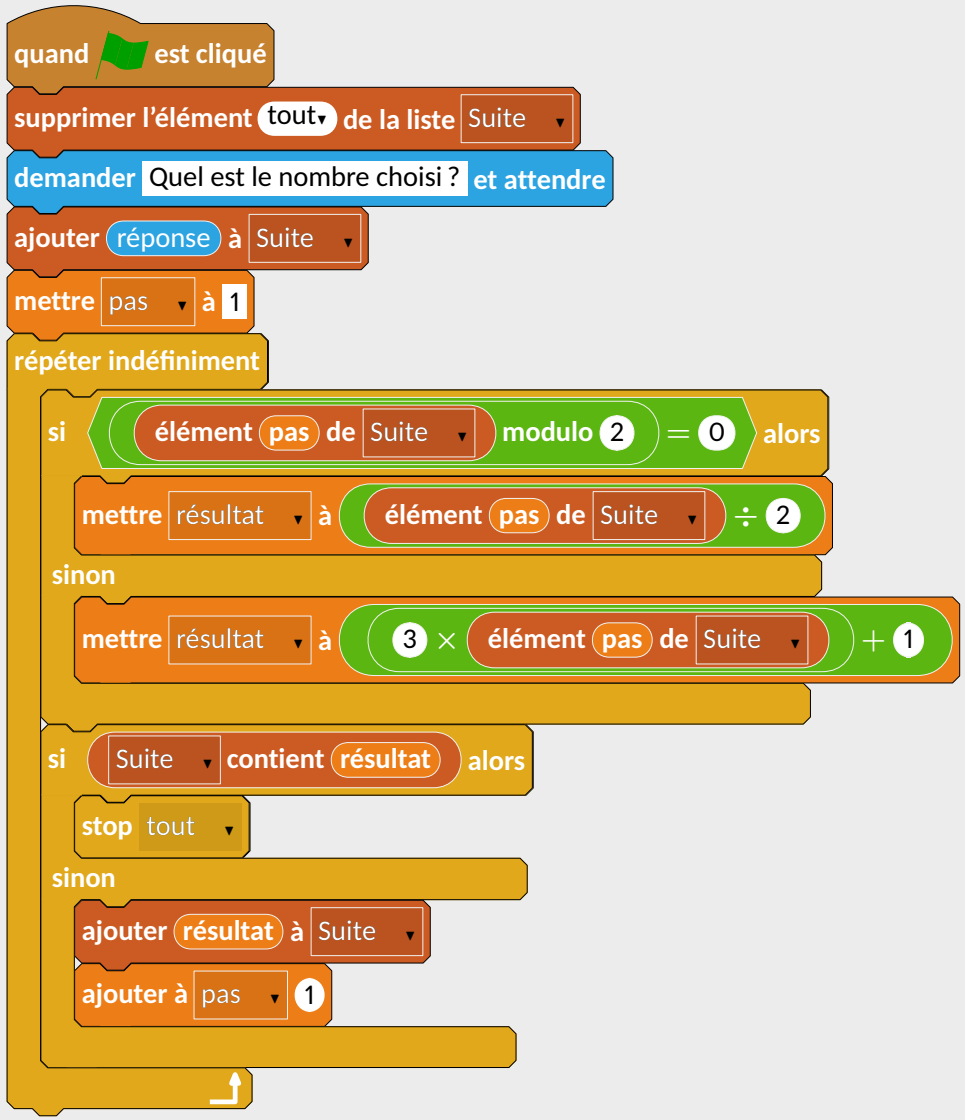
Groupe « Opérateurs »

Les éléments



s'obtiennent par les commandes OvalOp() et TestOp().

À vous de jouer !



```
draw Drapeau;
draw SupprimerListe("tout","Suite");
draw Demander("Quel␣est␣le␣nombre␣choisi␣?");
draw AjouterListe(OvalCap("réponse"),"Suite");
draw MettreVar("pas","1");
draw RepeterI;
picture BB[];
BB1=OvalListMulti("élément",OvalVar("pas"),"␣de␣",RecMenuList("Suite"));
BB2=OvalOp(BB1,"␣modulo␣",OvalNb("2"));
BB3=TestOp(BB2,"␣$\bm{=}$",OvalNb("0"));
BB4=OvalOp(BB1,"␣$\bm{\div}$␣",OvalNb("2"));
BB5=OvalOp(OvalNb("3"),"␣$\bm{\times}$␣",BB1);
BB6=OvalOp(BB5,"␣$\bm{+}$␣",OvalNb("1"));
draw Si(BB3);
draw MettreVar("résultat",BB4);
draw Sinon;
draw MettreVar("résultat",BB6);
draw FinBlocSi;
BB7=OvalListMulti(RecMenuList("Suite"),"␣contient␣",OvalVar("résultat"));
draw Si(BB7);
draw Stop("tout");
draw Sinon;
draw AjouterListe(OvalVar("résultat"),"Suite");
draw AjouterVar("pas","1");
draw FinBlocSi;
draw FinBlocRepeter;
```

Groupe « Ajouter blocs »

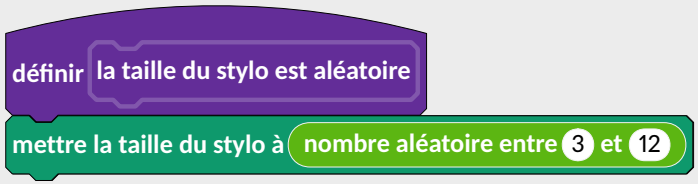
Groupe « Ajouter blocs »

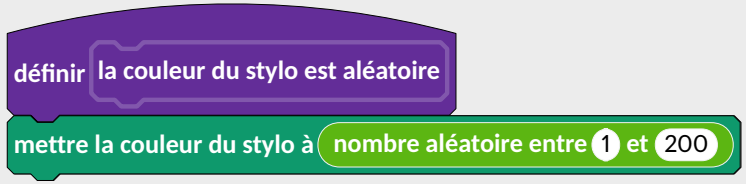
- **draw** NouveauBloc("Pentagone");

- **draw** NouveauBloc("Pentagone",OvalBloc("cote"));

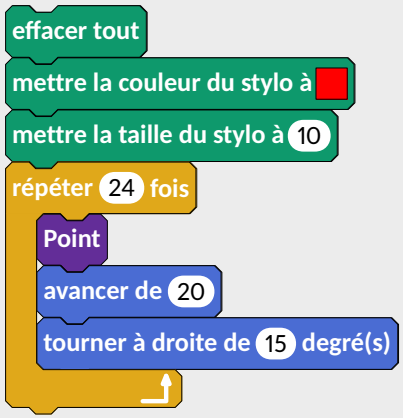
- **draw** Bloc("Pentagone");

À vous de jouer !

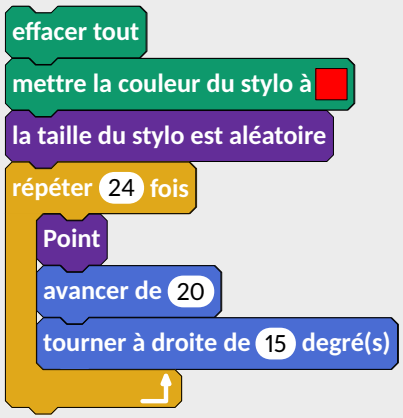




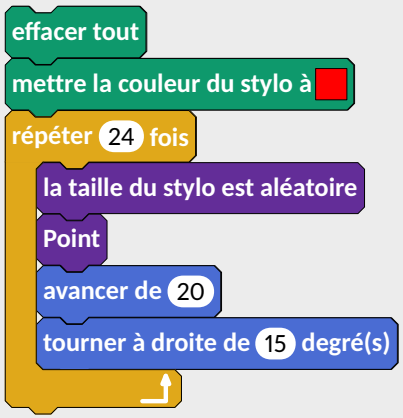
①



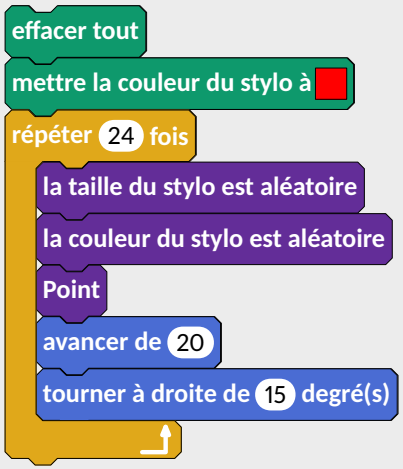
②



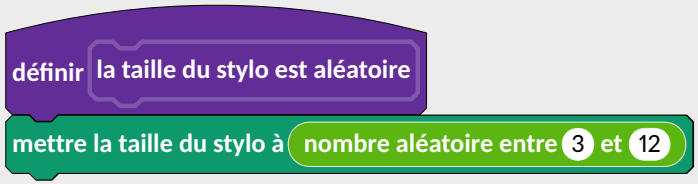
③

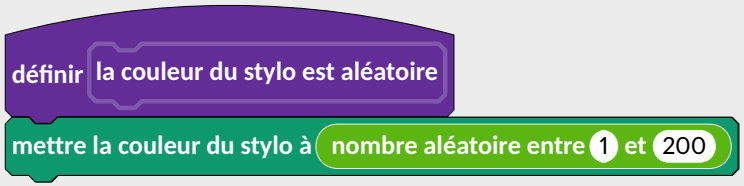


④



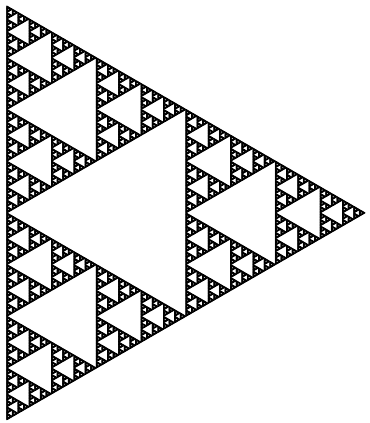
À vous de jouer !



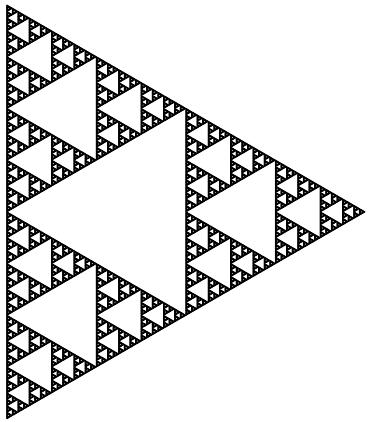


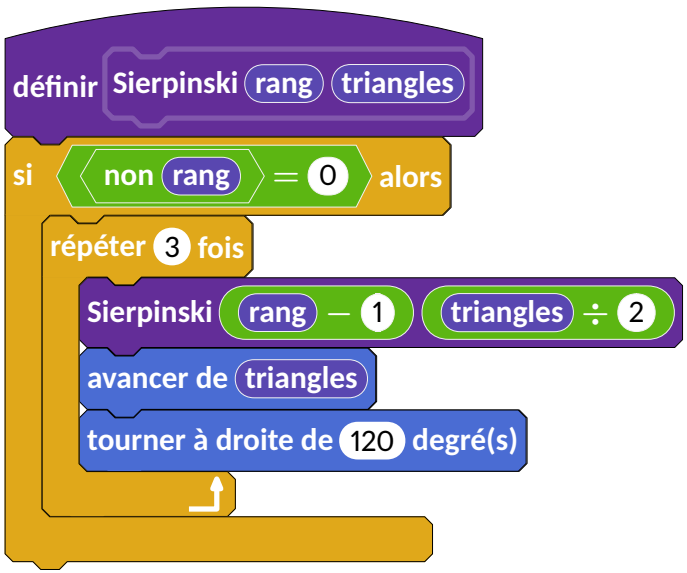

```
draw NouveauBloc("la taille du stylo est aléatoire");
draw MettreTS(OvalOp("nombre aléatoire entre ",OvalNb("3"),"et ",OvalNb("12")));
endfig;
```

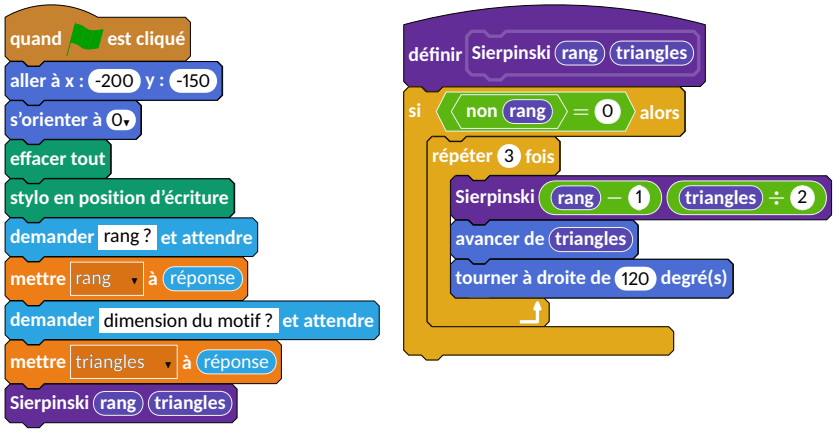
Avec plusieurs éléments ?



Avec plusieurs éléments ?







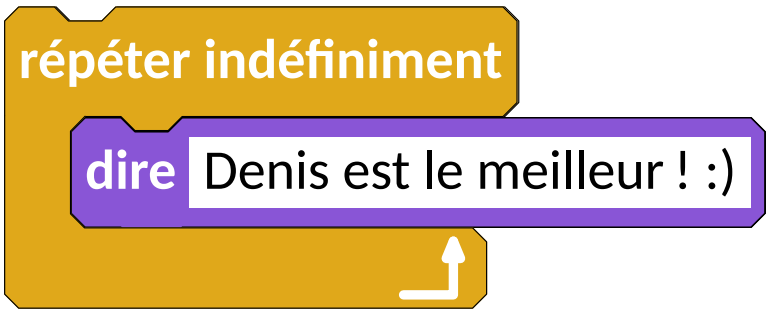

```
draw Drapeau;
draw Aller("-200","-150"); draw Orienter("0");
draw Effacer; draw PoserStylo;
draw Demander("rang?");
draw MettreVar("rang",OvalCap("réponse"));
draw Demander("dimension du motif?");
draw MettreVar("triangles",OvalCap("réponse"));
draw Bloc("Sierpinski",OvalBloc("rang"),OvalBloc("triangles"));
_coinprec:=(8.5cm,0);
draw NouveauBloc("Sierpinski",OvalBloc("rang"),OvalBloc("triangles"));
picture BB[];
BB1=TestOp("non",OvalBloc("rang"));
BB2=TestOp(BB1," = ",OvalNb("0"));
draw Si(BB2);
draw Repeter("3");
BB3=OvalOp(OvalBloc("rang")," - ",OvalNb("1"));
BB4=OvalOp(OvalBloc("triangles")," ÷ ",OvalNb("2"));
draw Bloc("Sierpinski",BB3,BB4);
draw Avancer(OvalBloc("triangles"));
draw Tournerd("120");
draw FinBlocRepeter; draw FinBlocSi;
```

Ce que **Scratch** n'a pas…

- Changer la couleur des lignes des blocs

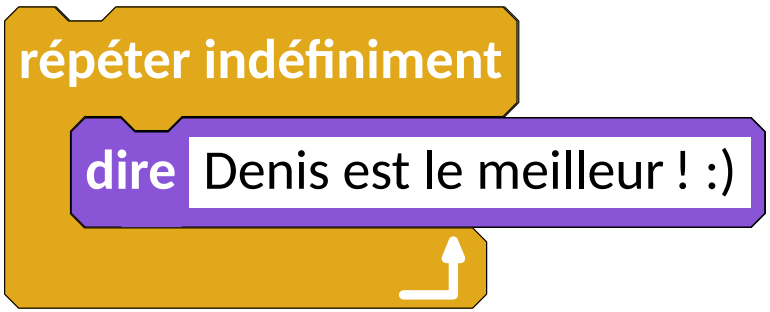
Ce que **Scratch** n'a pas…

- Changer la couleur des lignes des blocs



Ce que **Scratch** n'a pas...

- Changer la couleur des lignes des blocs



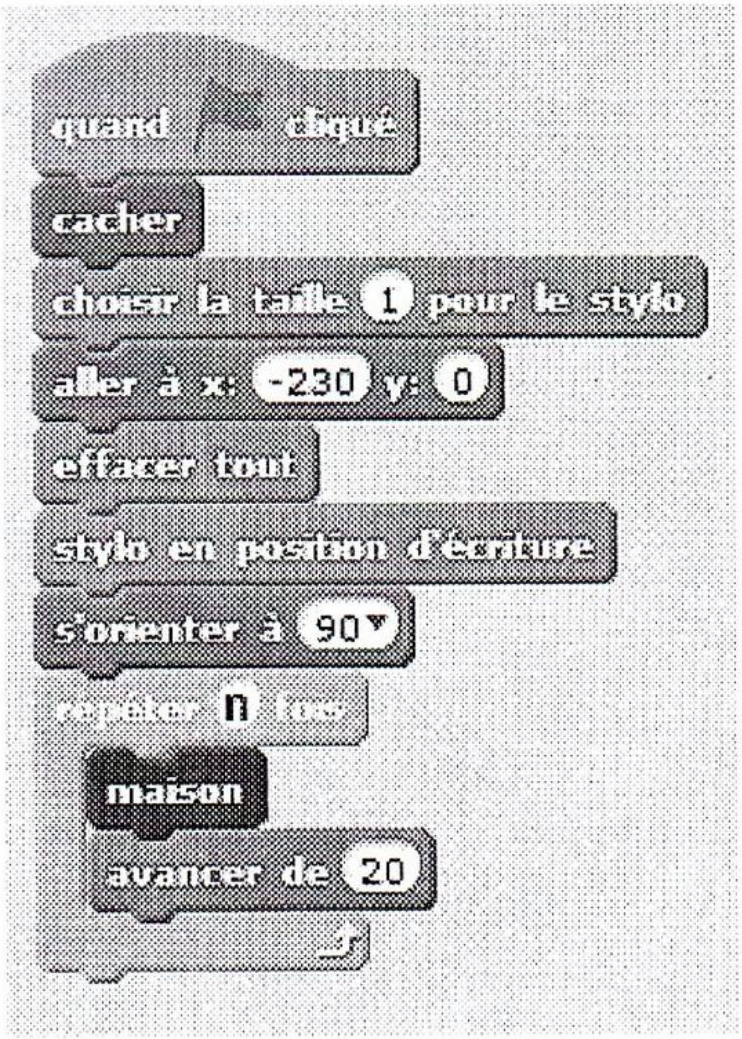

```
beginfig(1);
  draw RepeterI;
  draw Dire("Denis est le meilleur ! :)");
  draw FinBlocRepeterI;
endfig;

CoulLignes:=0.9white;

beginfig(2);
  draw RepeterI;
  draw Dire("Denis est le meilleur ! :)");
  draw FinBlocRepeterI;
endfig;
```

Ce que **Scratch** n'a pas…

- La possibilité d'adapter le contenu des blocs

Ce que **Scratch** n'a pas…

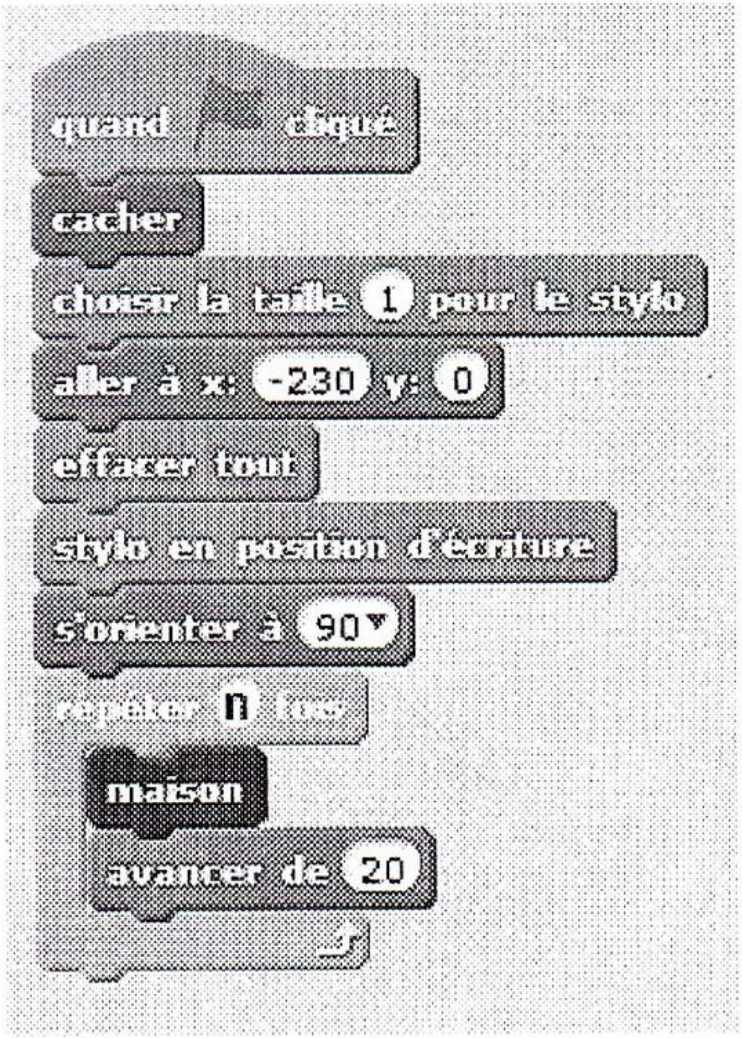
- La possibilité d'adapter le contenu des blocs



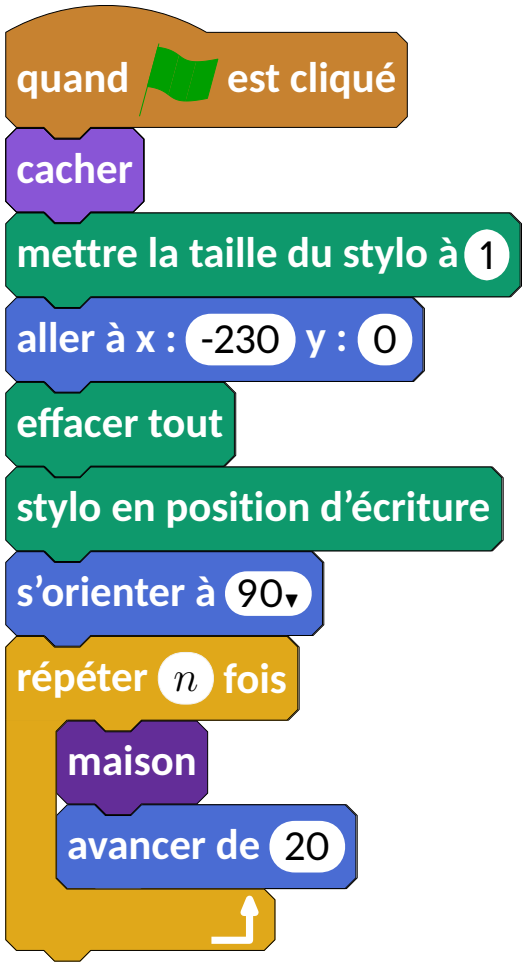
programme principal

Ce que **Scratch** n'a pas…

- La possibilité d'adapter le contenu des blocs



programme principal



Ce que **Scratch** n'a pas…

- Les lignes vides

Ce que **Scratch** n'a pas...

- Les lignes vides


```
beginfig(5);
  draw Si(TestOp(OvalVar("x"),"␣$\bm{=}$␣",OvalNb("20")));
  draw LigneVide;
  draw Sinon;
  draw LigneVide;
  draw FinBlocSi;
endfig;
```

Ce que **Scratch** n'a pas…

- Les blocs vides

Ce que **Scratch** n'a pas...

- Les blocs vides

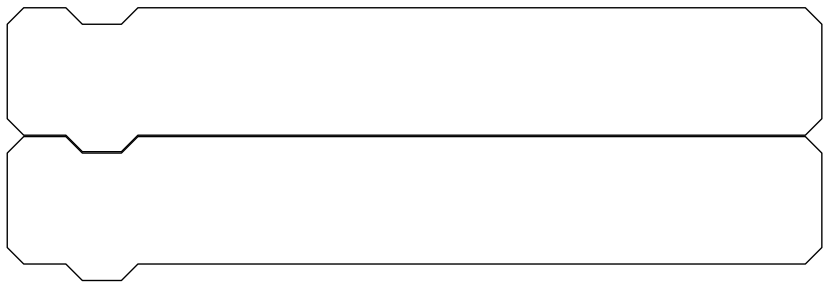

```
draw Drapeau;
draw Repeter("4");
draw Avancer("50");
draw CommandeVide("5");
draw FinBlocRepeter;
```

Ce que **Scratch** n'a pas...

Ce que **Scratch** n'a pas…

- Les lignes pointillés

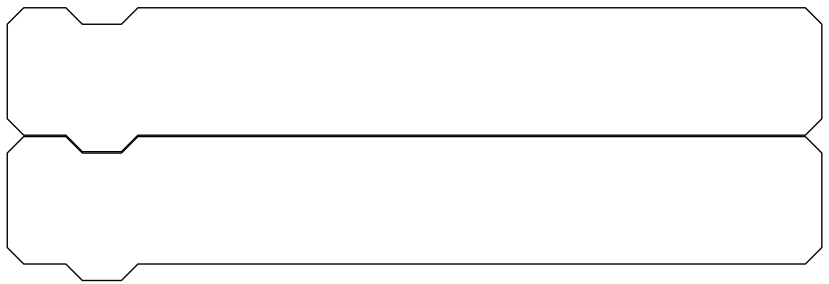
• • •



Ce que **Scratch** n'a pas…

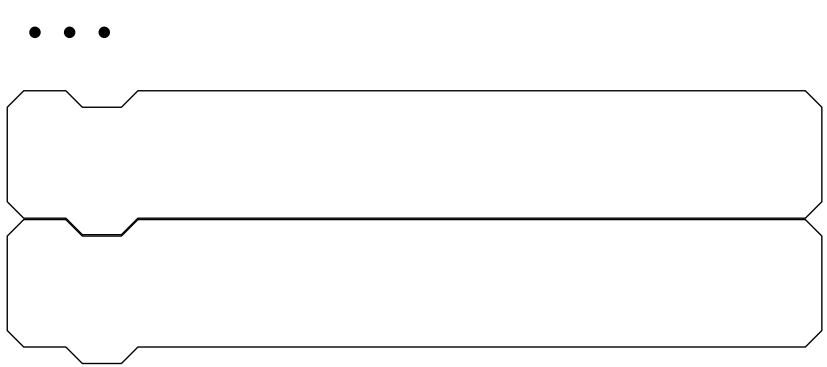
- Les lignes pointillés

• • •



Ce que **Scratch** n'a pas...

- Les lignes pointillés



```
draw Drapeau;
draw LignePointilles;
draw CommandeVide("5");
draw CommandeVide("5");
```

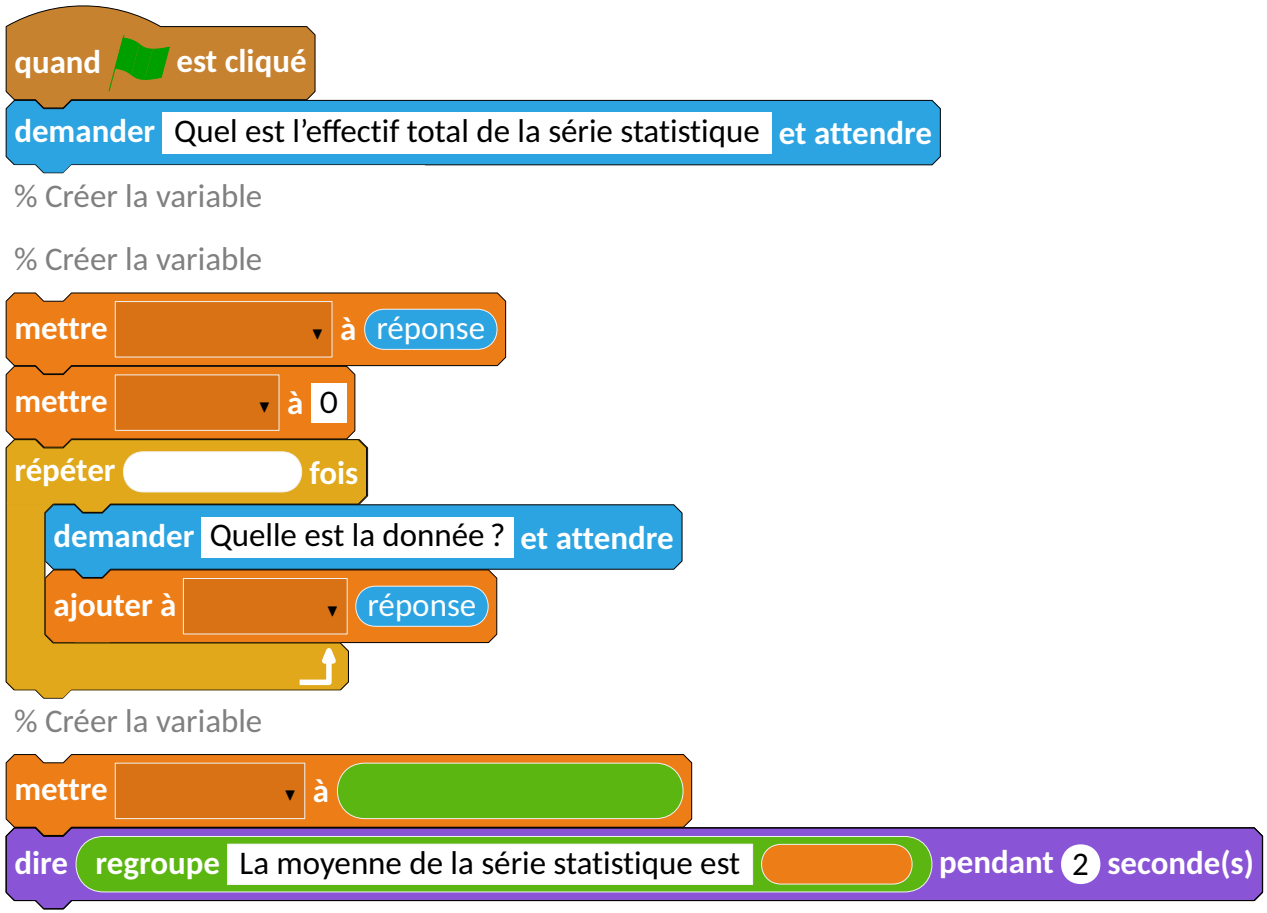
Ce que **Scratch** n'a pas…

Ce que **Scratch** n'a pas…

- Les commentaires « en continu »

Ce que **Scratch** n'a pas…

- Les commentaires « en continu »



Ce que **Scratch** n'a pas...

```
draw Drapeau;
draw Demander("Quel est l'effectif total de la série statistique ?");
draw Commentaires("% Créer la variable");
draw Commentaires("% Créer la variable");
draw MettreVar("\phantom{Effectiftotal}",OvalCap("réponse"));
draw MettreVar("\phantom{Somme}","0");
draw Repeter(OvalNb("\phantom{Effectiftotal}"));
draw Demander("Quelle est la donnée ?");
draw AjouterVar("\phantom{Somme}",OvalCap("réponse"));
draw FinBlocRepeter;
draw Commentaires("% Créer la variable");
draw MettreVar("\phantom{Moyenne}",OvalOp("\phantom{$\mbox{Somme}\bm{\div}\mbox{Effectiftotal}$}
    "));
draw DireT(OvalOp("regroupe",RecText("La moyenne de la série statistique est
    "),OvalVar("\phantom{Moyenne}")),"2");
```

Ce que **Scratch** n'a pas. . .

Ce que **Scratch** n'a pas…

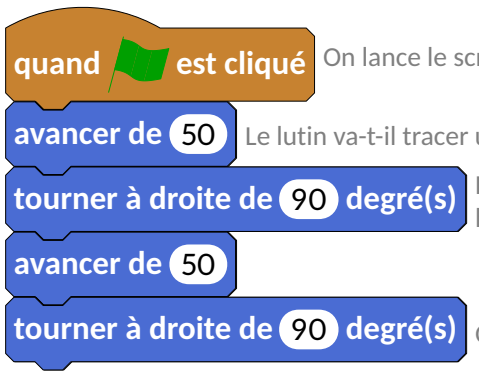
- Les commentaires

Ce que **Scratch** n'a pas…

- Les commentaires en fin de ligne

Ce que **Scratch** n'a pas…

- Les commentaires en fin de ligne



On lance le script par ce bloc

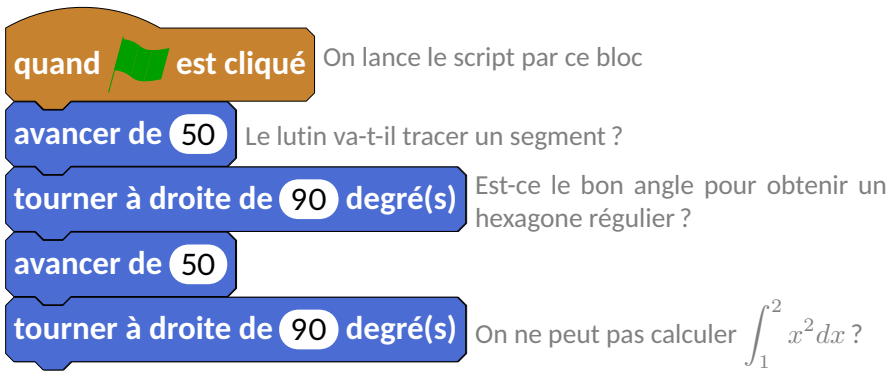
Le lutin va-t-il tracer un segment ?

Est-ce le bon angle pour obtenir un hexagone régulier ?

On ne peut pas calculer $\int_1^2 x^2 dx$?

Ce que **Scratch** n'a pas…

- Les commentaires en fin de ligne




```
draw CommentairesLigne("\footnotesize On lance le script par ce bloc");
draw Avancer("50");
draw CommentairesLigne("\footnotesize Le lutin va-t-il tracer un segment ?");
draw Tournerd("90");
draw CommentairesLigne("\footnotesize \begin{minipage}{150 pt}Est-ce le bon angle pour obtenir un hexagone
    régulier ?\end{minipage}");
draw Avancer("50");
draw Tournerd("90");
draw CommentairesLigne("\footnotesize On ne peut pas calculer $\displaystyle \int_1^2x^2dx$ ?");
```

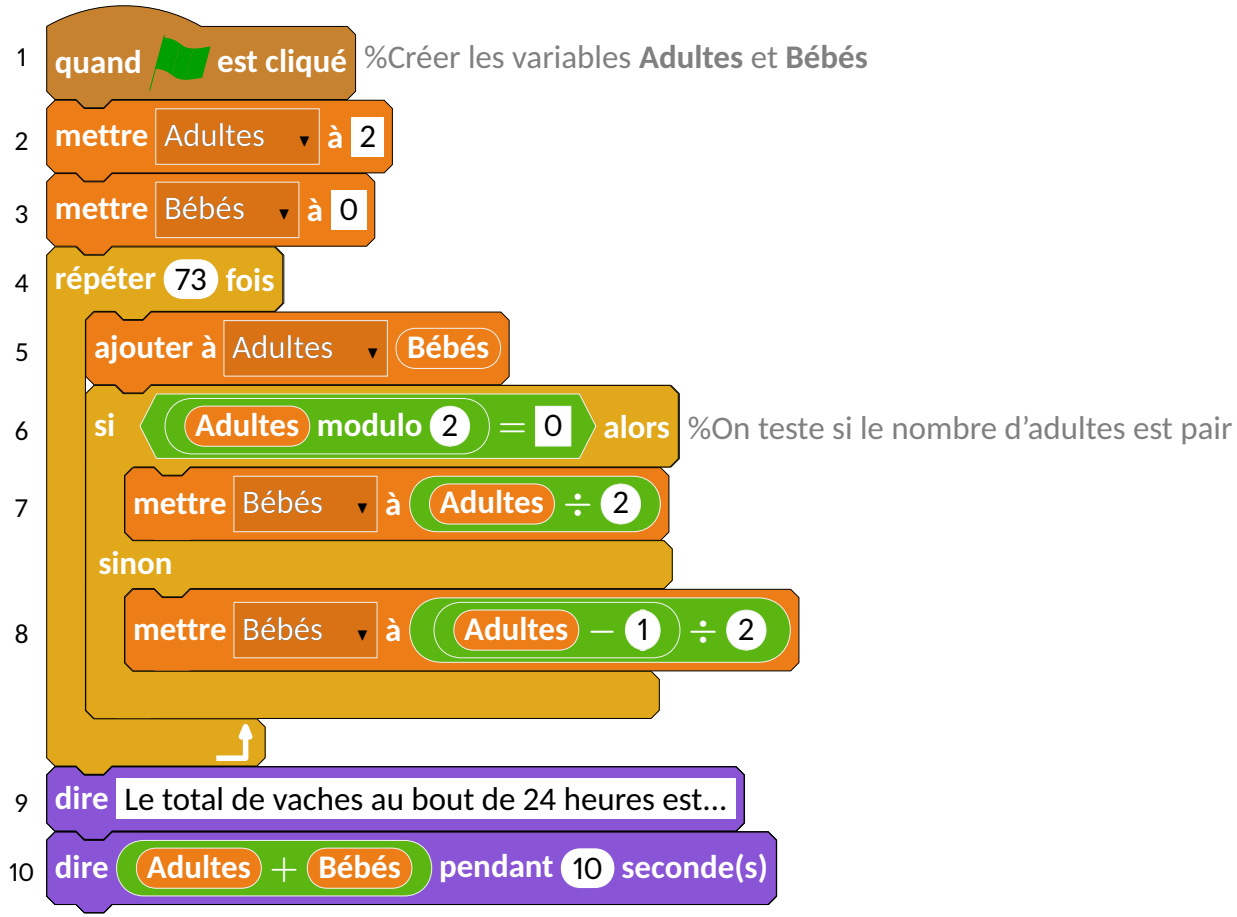
Ce que **Scratch** n'a pas...

Ce que **Scratch** n'a pas…

- La numérotation des lignes

Ce que **Scratch** n'a pas…

- La numérotation des lignes



Ce que **Scratch** n'a pas...

```
draw Drapeau;
draw CommentairesLigne("%Créer les variables \textbf{Adultes} et \textbf{Bébés}");
draw MettreVar("Adultes","2");
draw MettreVar("Bébés","0");
draw Repeter("73");
picture BB[];
BB1=OvalVar("Bébés");
BB2=OvalOp(OvalVar("Adultes"),"modulo",OvalNb("2"));
BB3=TestOp(BB2,"$\bm{=}$",RecText("0"));
BB4=OvalOp(OvalVar("Adultes"),"$\bm{\div}$",OvalNb("2"));
BB5=OvalOp(OvalVar("Adultes"),"$\bm{-}$",OvalNb("1"));
BB6=OvalOp(BB5,"$\bm{\div}$",OvalNb("2"));
BB7=OvalOp(OvalVar("Adultes"),"$\bm{+}$",OvalVar("Bébés"));
draw AjouterVar("Adultes",BB1);
draw Si(BB3);
draw CommentairesLigne("%On teste si le nombre d'adultes est pair");
draw MettreVar("Bébés",BB4);
draw Sinon;
draw MettreVar("Bébés",BB6);
draw FinBlocSi;
draw FinBlocRepeter;
draw Dire("Le total de vaches au bout de 24 heures est...");
draw DireT(BB7,"10");
```

Ce que **Scratch** n'a pas...

Ce que **Scratch** n'a pas…

- L'impression

Ce que **Scratch** n'a pas…

- L'impression

Ce que **Scratch** n'a pas…

- L'impression


```
coefprint:=0.9;
print:=true;

NumeroteLignes:=true;

draw Drapeau;
draw Effacer;
draw Aller("-200","-100");

print:=false;

draw Orienter("90");
draw MettreVar("côté","100");

print:=true;

draw Repeter("5");
draw Bloc("triangle");
draw Avancer(OvalBloc("côté"));
draw AjouterVar("côté","-20");
draw FinBlocRepeter;
```

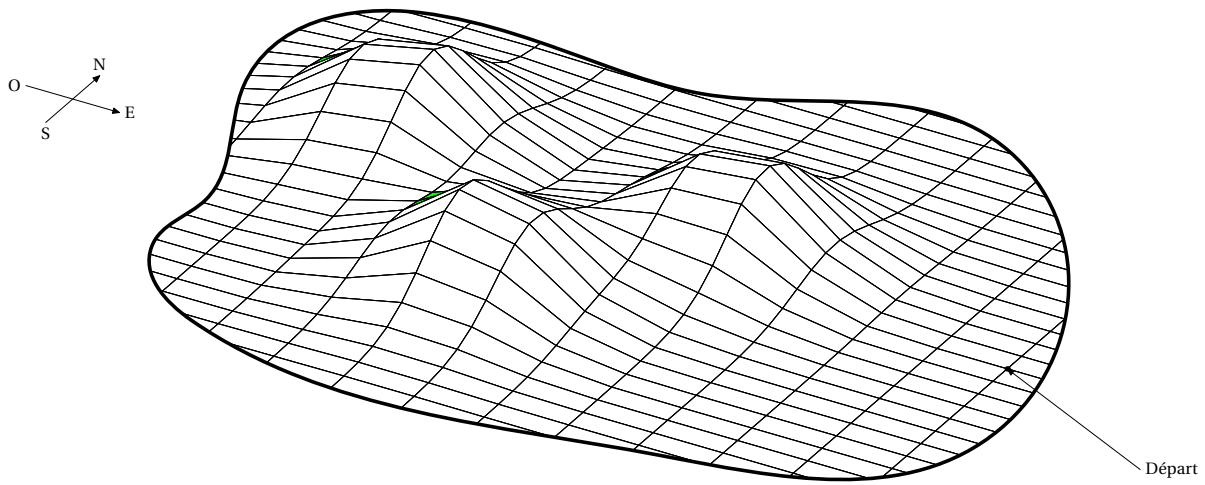
Ce que **Scratch** n'a pas...

Ce que **Scratch** n'a pas…

- Les blocs personnalisables

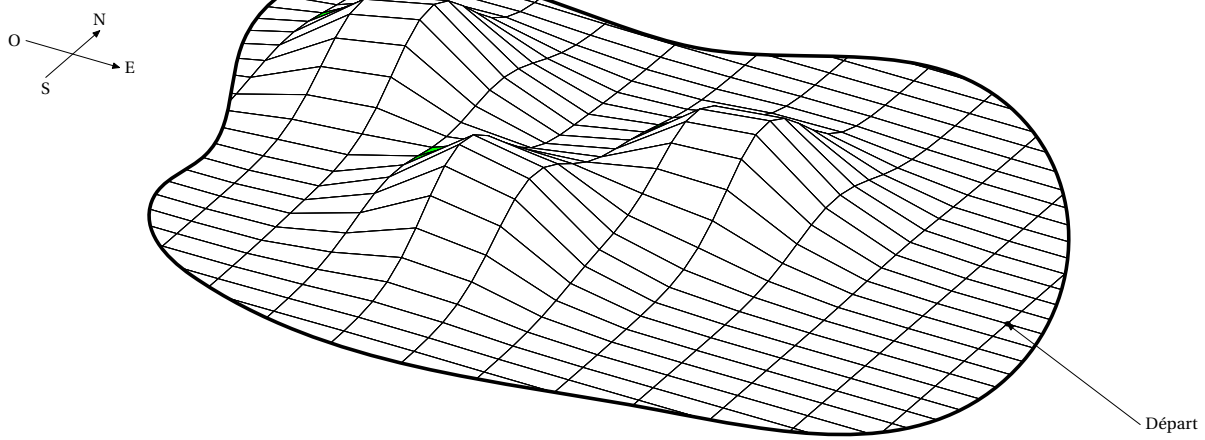
Ce que **Scratch** n'a pas…

- Les blocs personnalisables



Ce que **Scratch** n'a pas…

- Les blocs personnalisables



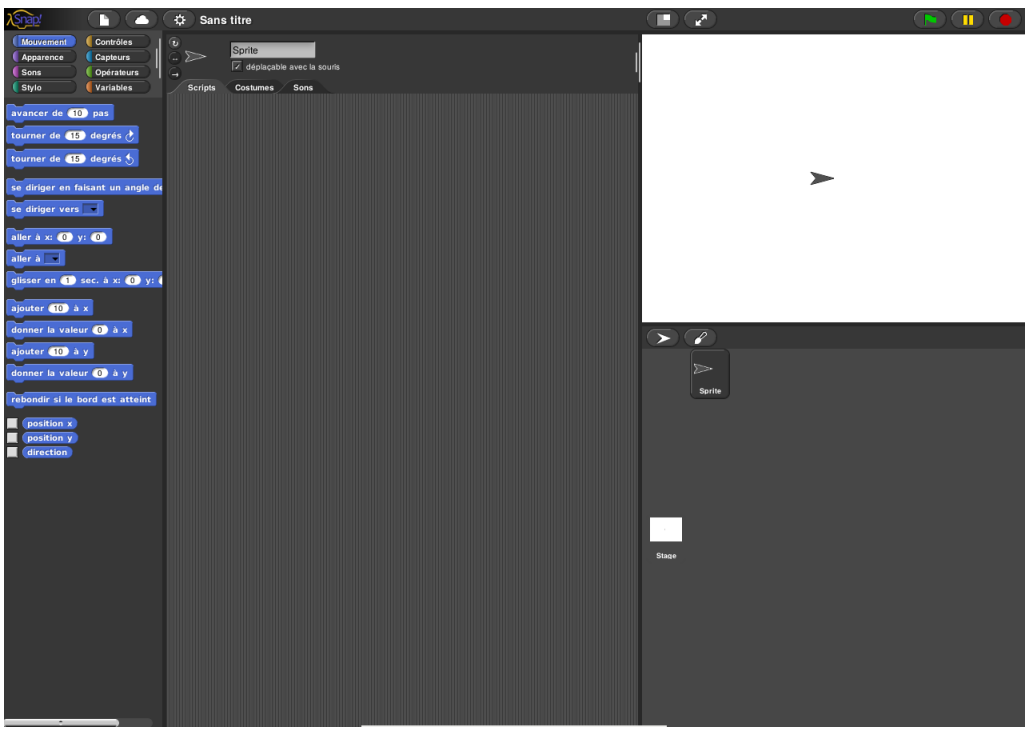

```
draw BlocUser((44/256,120/256,195/256))("Avancer de
 ","OvalNb("3"),"vers l'ouest");
```

Perspectives ?

- Poursuivre le développement :)

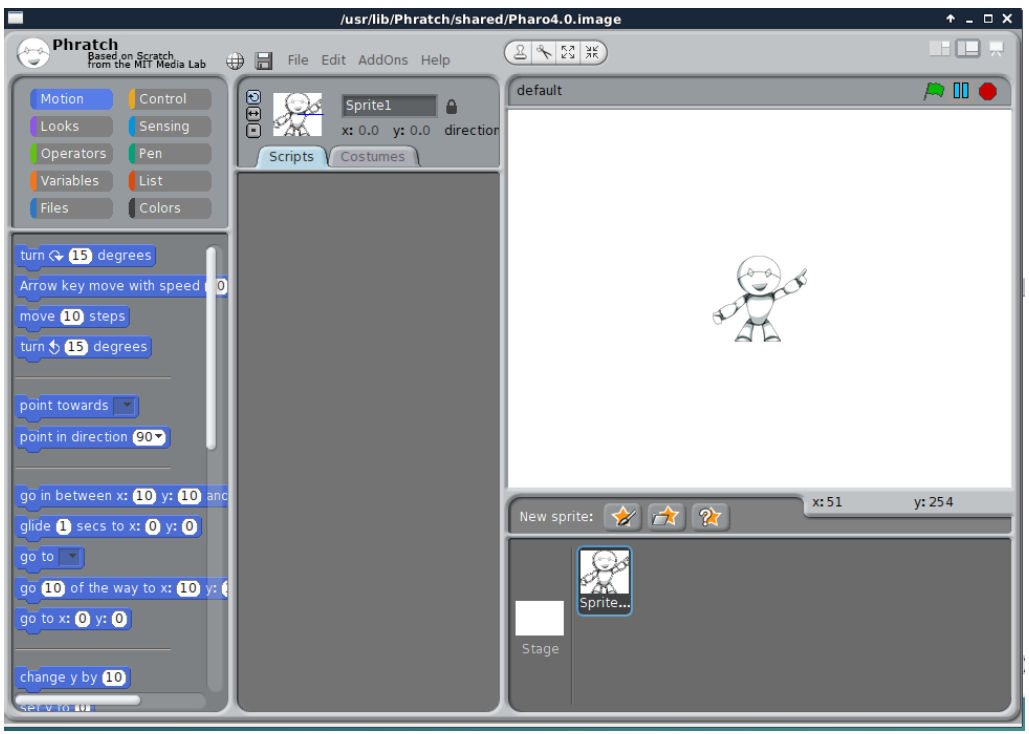
Perspectives ?

- Poursuivre le développement :)
- Snap ? `http://snap.berkeley.edu/`



Perspectives ?

- Poursuivre le développement :)
- Snap ? `http://snap.berkeley.edu/`
- Phratch ? `http://www.phratch.com/`



Perspectives ?

- Poursuivre le développement :)
- Snap ? `http://snap.berkeley.edu/`
- Phratch ? `http://www.phratch.com/`
- Scratch 3 ? `https://llk.github.io/scratch-gui/custom-procedures/`