# Sophus et Sofus

## Variables et POO

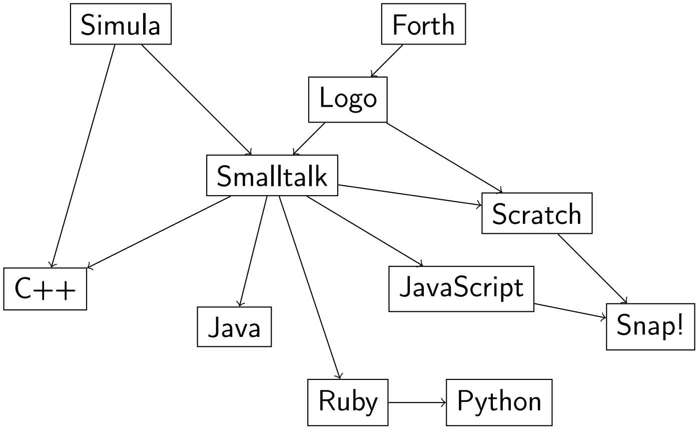
Alain Busser

3 février 2021

# Programme de NSI
## Terminale

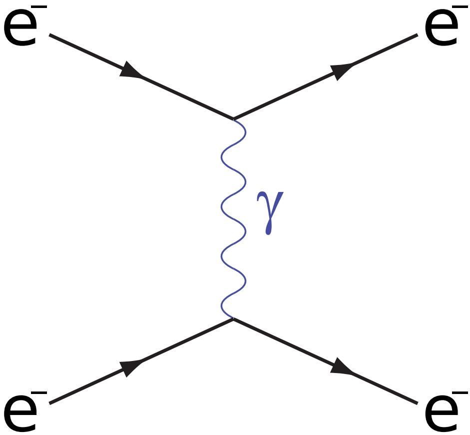
| Contenus | Capacités attendues | Commentaires |
|---|---|---|
| Vocabulaire de la programmation objet : classes, attributs, méthodes, objets. | Écrire la définition d'une classe. Accéder aux attributs et méthodes d'une classe. | On n'aborde pas ici tous les aspects de la programmation objet comme le polymorphisme et l'héritage. |

# Naissance de la POO

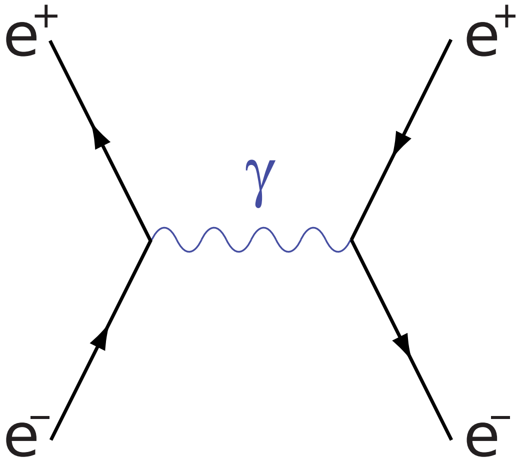


# Les objets de Simula

## Diagramme de Feynman



# Rotation du diagramme
## Diagramme de Feynman



# Alan Kay

## Prix Turing en 2003

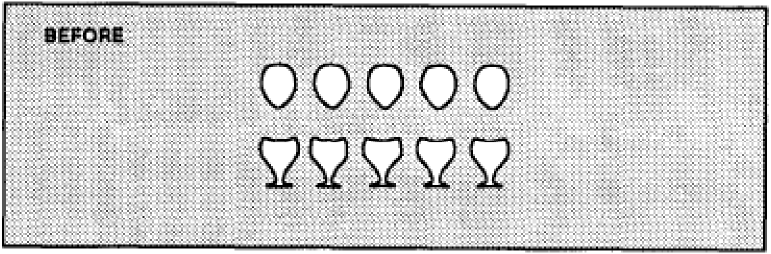
Étudiant de Papert

- Smalltalk (POO)
- fenêtres, bureau...
- ordinateur portable
- Atari, Apple, HP, Walt Disney...
- Mme Kay était l'actrice principale de Tron

Un ensemble d'objets communiquant entre eux est un **micromonde**.

# Piaget selon Minsky

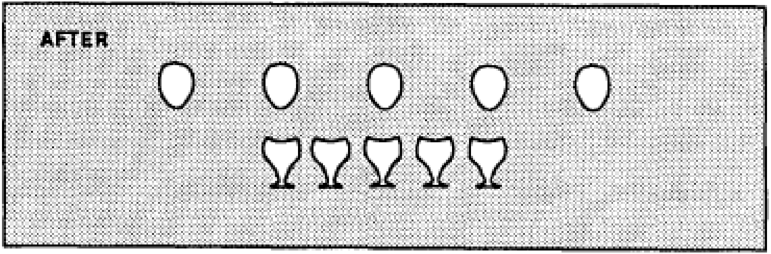
## Y a-t-il plus d'œufs ou de coquetiers ?




- 5 ans : autant de chaque
- 7 ans : autant de chaque

# Piaget selon Minsky

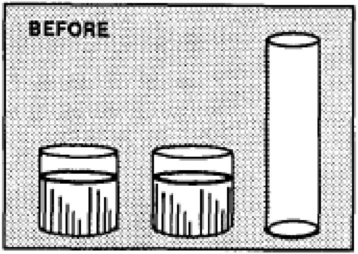
## Y a-t-il plus d'œufs ou de coquetiers ?

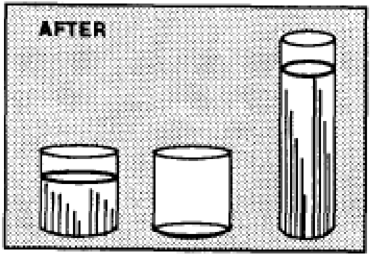



- 5 ans : plus d'œufs que de coquetiers
- 7 ans : autant de chaque

# Piaget selon Minsky

## Principe d'invariance (Papert)




Il y en a plus après, qu'avant.

# Logo

## Seymour Papert

Deux définitions en Logo

- `TO procedure`
- `CALL variable`

Exemple

```
CALL 3.14159265 "pi"
CALL 2.25 "R"
```

Alors

```
2 * :pi * :R
```

est plus simple que

```
2 * 3.14159265 * 2.25
```

# Logo

## Seymour Papert

Trois applications de Logo à l'enseignement de l'analyse :

- calcul intégral
- équations différentielles
- invariants topologiques

```
CALL :TOTAL + FIELD "TOTAL"
```

# Programme de mathématiques
## Première technologique

Le programme vise la consolidation des notions de variable[..]

**Capacités attendues**

*Variables :*

- utiliser un générateur de nombres aléatoires entre 0 et 1 pour simuler une loi de Bernoulli de paramètre $p$ ;
- utiliser la notion de **compteur** ;
- utiliser le principe d'**accumulateur** pour calculer une somme, un produit.

# Attribut

| tortue | variable |
|---|---|
| • abscisse `tortue.x`<br>• ordonnée `tortue.y`<br>• direction `tortue.heading` | valeur de la variable `variable.valeur` |

Lorsqu'une variable se réfère à sa propre valeur on écrit `self.valeur`

# Méthode

## Accesseur

| tortue | variable |
| --- | --- |
| • abscisse `tortue.getposition()`<br>• ordonnée `tortue.getposition()`<br>• direction `tortue.angle()` | valeur de la variable `variable.repr()` |

```
def __repr__(self):
    return str(self.valeur)
```

permet de faire `print(variable)`

# Méthode

## Transformateur (ou mutateur)

| tortue | variable |
|---|---|
| • avancer `tortue.forward()` | • augmenter `variable.augmentetoide()` |
| • reculer `tortue.backward()` | • diminuer `variable.diminuetoide()` |
| • tourner `tortue.left()` | • affecter `variable.prendlavaleur()` |

## Augmentation

```
def augmente_toi_de(self,d=1,mode=None):
    if mode == pourcents:
        self.valeur *= 1+d/100
    else:
        self.valeur += d
```

# Diminution

```
def diminue_toi_de(self,d=1,mode=None):
    if mode == pourcents:
        self.valeur *= 1-d/100
    else:
        self.valeur -= d
```

# Autres méthodes

```
def multiplie_toi_par(self,d=2,mode=None):
    if mode == pourcents:
        self.valeur *= d/100
    else:
        self.valeur *= d
def divise_toi_par(self,d=2,mode=None):
    if mode == pourcents:
        self.valeur /= d/100
    else:
        self.valeur /= d
```

## Autres méthodes

```
def eleve_toi(self,p=au_carre):
        self.valeur **= p
def inverse_toi(self):
        self.valeur = 1/self.valeur
def double_toi(self):
        self.valeur *= 2
def triple_toi(self):
        self.valeur *= 3
def decuple_toi(self):
        self.valeur *= 10
```

# Objet et Classe

```
class Variable():
    def __init__(self,v):
        self.valeur = float(v)
    def __repr__(self):
        return str(self.valeur)
```

## Collatz

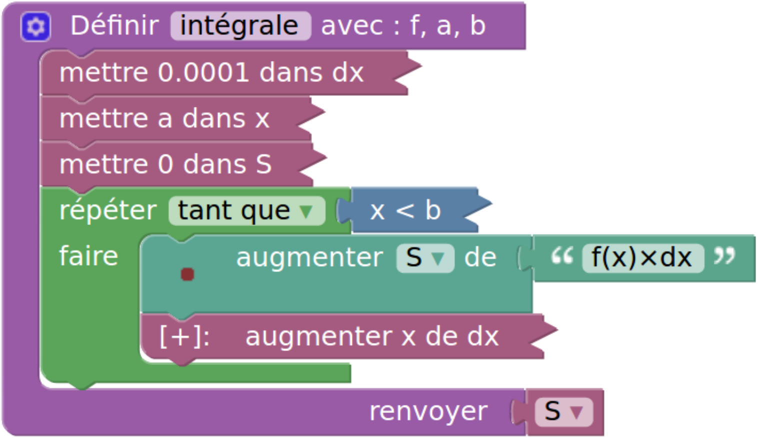
```
u = Variable(13)
while u.valeur != 1:
    if u.est_impair():
        u.triple_toi()
        u.augmente_toi_de(1)
    u.divise_toi_par(2)
```

# Intégrale

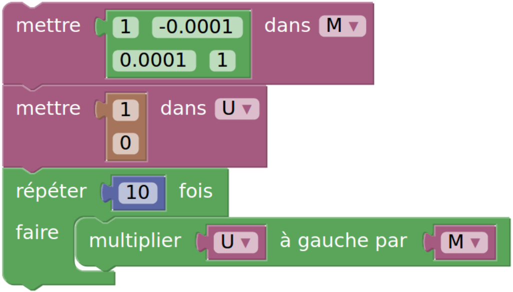
## méthode des rectangles

```
from sofus import *
def integrale(f,a,b):
    dx = 0.0001
    x = Variable(a)
    S = Variable(0)
    while x.valeur < b:
        S.augmente_toi_de(f(x.valeur)*dx)
        x.augmente_toi_de(dx)
    return S.valeur
```

# Intégrale
## méthode des rectangles

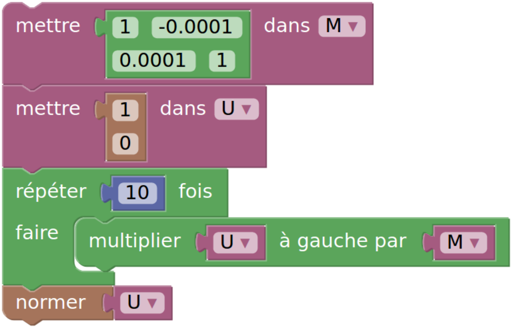


# Matrices
## Similitude itérée




$$\vec{U}\begin{pmatrix} 0{,}99999955 \\ 0{,}00099999988 \end{pmatrix}$$

# Matrices
## Rotation




$$\vec{U}\begin{pmatrix} 0{,}99999955 \\ 0{,}00099999988 \end{pmatrix} \quad \frac{\vec{U}}{\|\vec{U}\|}\begin{pmatrix} 0{,}9999995 \\ 0{,}00099999983 \end{pmatrix}$$

# Algorithme CORDIC

## COordinate Rotation DIgital Computer

(Jack Volder 1959) Pour appliquer à $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ une rotation de 0,25 radian, on peut lui appliquer

- 2 rotations d'un déciradian
- 5 rotations d'un centiradian

$$\begin{pmatrix} \cos(0{,}25) & -\sin(0{,}25) \\ \sin(0{,}25) & \cos(0{,}25) \end{pmatrix} =$$

$$\begin{pmatrix} \cos(0{,}1) & -\sin(0{,}1) \\ \sin(0{,}1) & \cos(0{,}1) \end{pmatrix}^2 \times \begin{pmatrix} \cos(0{,}01) & -\sin(0{,}01) \\ \sin(0{,}01) & \cos(0{,}01) \end{pmatrix}^5$$

# Groupe de Lie

L'ensemble des matrices $\begin{pmatrix} \cos(t) & -\sin(t) \\ \sin(t) & \cos(t) \end{pmatrix}$ est le groupe $\mathcal{S}\mathbb{O}_2(\mathbb{R})$.

Un groupe de Lie est un groupe tel que

- Pour tout $a$, $x \mapsto a \times x$ est différentiable
- Pour tout $b$, $x \mapsto x \times b$ est différentiable
- $x \mapsto x^{-1}$ est différentiable

# Algorithme CORDIC
## COordinate Rotation DIgital Computer

Pour appliquer à $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ une similitude de 0,25 radian, on peut lui appliquer

- 2 similitudes d'un déciradian
- 5 similitudes d'un centiradian

$$\begin{pmatrix} 1 & \text{-tan(0,25)} \\ \text{tan(0,25)} & 1 \end{pmatrix} \propto \begin{pmatrix} 1 & \text{-tan(0,1)} \\ \text{tan(0,1)} & 1 \end{pmatrix}^2 \times \begin{pmatrix} 1 & \text{-tan(0,01)} \\ \text{tan(0,01)} & 1 \end{pmatrix}^5$$

Une fois $T = tan(t)$ calculé, on en déduit $cos(t) = \sqrt{\frac{1}{1+T^2}}$ et $sin(T) = T \times cos(t)$

# Espace tangent

L'ensemble des $\begin{pmatrix} 1 \\ \tan(t) \end{pmatrix}$ est une droite : l'espace tangent au groupe de Lie, en $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$.

$$\begin{pmatrix} 1 & -\tan(t) \\ \tan(t) & 1 \end{pmatrix} = I + tan(t) \times \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

L'ensemble des $\begin{pmatrix} 0 & -z \\ z & 0 \end{pmatrix}$ est un espace vectoriel : l'algèbre de Lie $\mathfrak{so}_2$ du groupe de Lie $\mathcal{S}\mathbb{O}_2\,(\mathbb{R})$.

$$\begin{pmatrix} 0 & -a \\ a & 0 \end{pmatrix} \begin{pmatrix} 0 & -b \\ b & 0 \end{pmatrix} = \begin{pmatrix} -ab & 0 \\ 0 & -ab \end{pmatrix}$$

# Algèbre de Lie

$[A, B] = AB - BA$
$B \mapsto [A, B]$ est l'adjointe de $A$ notée $ad_A$
$(A, B) \mapsto tr(ad_A \circ ad_B)$ est une forme bilinéaire : la forme de Killing
Si $\exists B \in \mathfrak{g}, \forall A \in \mathfrak{g}, [A, B] = \alpha(A) \times B$, la forme linéaire $\alpha$ s'appelle une **racine** de $\mathfrak{g}$
Le groupe des symétries des racines s'appelle le groupe de Weyl de l'algèbre de Lie.

# Système de racines

## Définition

Un ensemble de vecteurs $\alpha$ d'un e.v. euclidien $E$ est un **système de racines** si

- L'e.v. engendré par les $\alpha$ est $E$
- Les seuls vecteurs du système qui sont colinéaires à $\alpha$ sont $\alpha$ et $-\alpha$
- le système de racines est symétrique par rapport à l'hyperplan perpendiculaire à $\alpha$
- le produit scalaire de deux racines $\alpha$ et $\beta$ est le multiple du carré scalaire de $\alpha$ par un demi-entier.

L'angle entre deux racines ne peut être que $\frac{\pi}{2}$, $\frac{\pi}{3}$, $\frac{\pi}{4}$, $\frac{\pi}{6}$ ou leur supplémentaire.

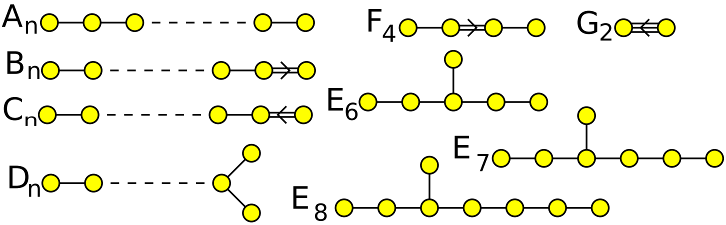
# Diagramme de Dynkin

Eugene Dynkin associe à chaque système de racines, un multigraphe, tel que

- chaque sommet du graphe représente une racine
- si l'angle entre deux racines est $\frac{\pi}{2}$, il n'y a pas d'arête entre les deux sommets
- si l'angle entre deux racines est $2\frac{\pi}{3}$, il y a une arête entre les sommets
- si l'angle entre deux racines est $3\frac{\pi}{4}$, il y a une double arête entre les sommets
- si l'angle entre deux racines est $5\frac{\pi}{6}$, il y a une triple arête entre les sommets

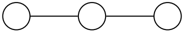
# Classification des groupes de Lie

## Il n'existe pas d'autres diagrammes de Dynkin que

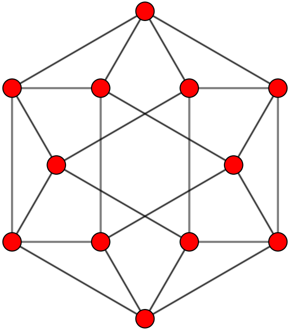


## $D_3 = A_3$

### algèbre de Lie $\mathfrak{so}(6)$ groupe de Lie $\mathbb{SO}(6)$

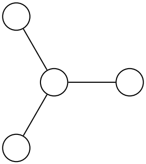


Le système de racines est un cuboctaèdre :



## $D_4$

algèbre de Lie $\mathfrak{so}(8)$ groupe de Lie $\mathbb{SO}(8)$



Le système de racines est un icositétrachore :