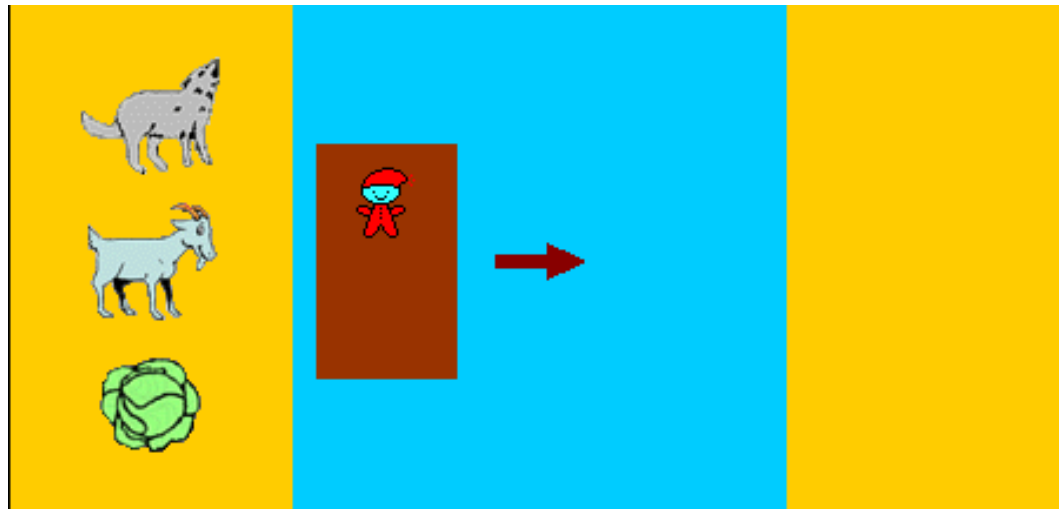
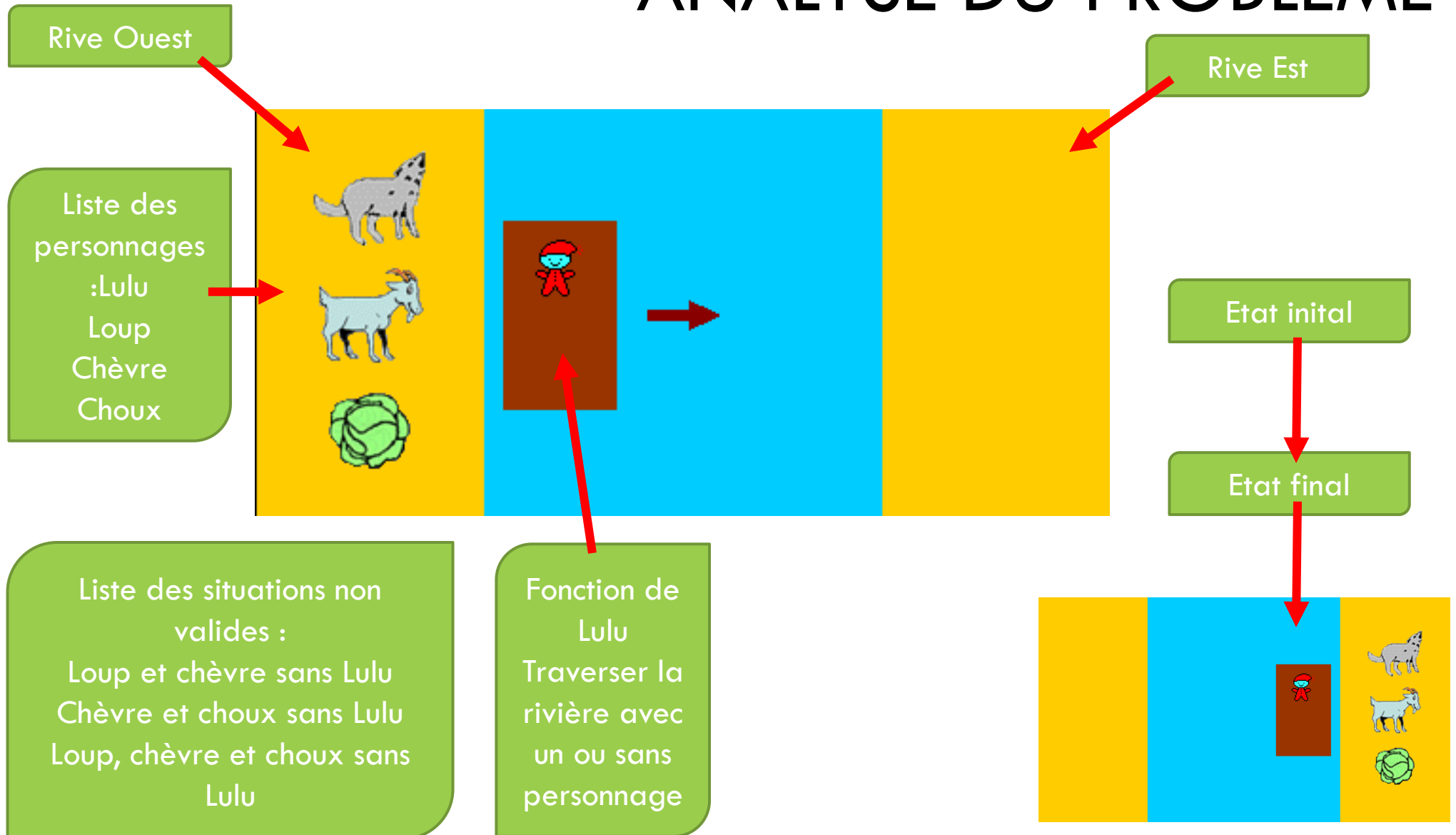


LE LOUP, LA CHÈVRE ET LE CHOUX

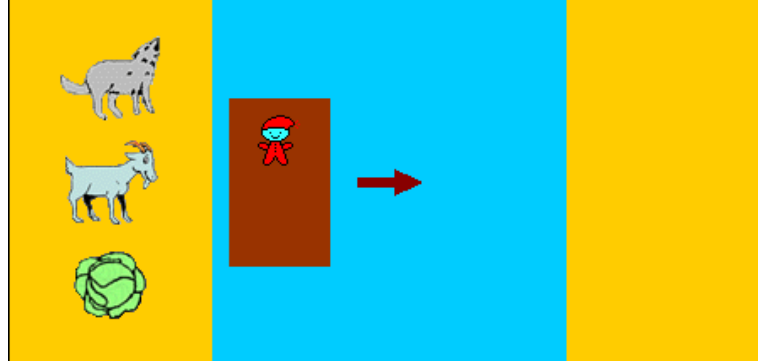


Lulu doit faire passer le chou, la chèvre et le loup de l'autre côté de la rivière.
Mais il n'a qu'une place sur son bateau !
De plus, si la chèvre et le chou sont ensemble sur une rive quand Lulu s'éloigne, la chèvre mange le chou. Et si le loup et la chèvre sont ensemble quand Lulu s'éloigne, le loup mange la chèvre !

ANALYSE DU PROBLÈME



ALGORITHME



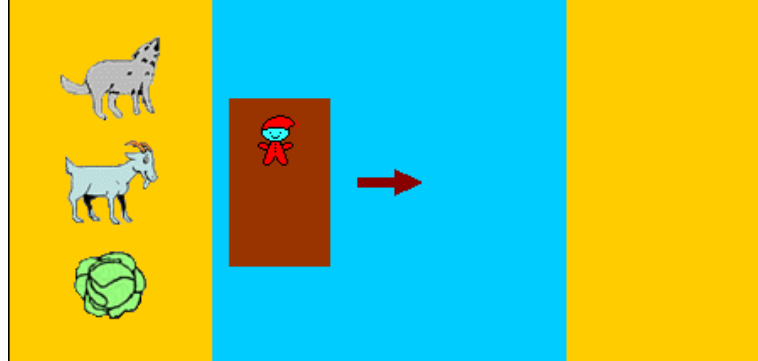
Lulu peut-elle traverser seule ?
-> aucun intérêt et situation non valide

Lulu peut-elle traverser avec le loup ?
-> situation non valide

Lulu peut-elle traverser avec la chèvre ?
-> oui

Lulu traverse avec la chèvre

ALGORITHME



Lulu peut-elle traverser seule ?

-> aucun intérêt et situation non valide

Lulu peut-elle traverser avec le loup ?

-> situation non valide

Lulu peut-elle traverser avec la chèvre ?

-> oui

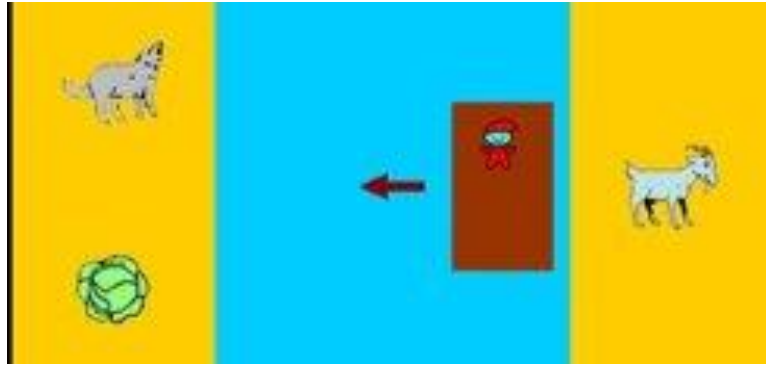
Lulu traverse avec la chèvre

Tant que la situation finale n'est pas réalisée, on continue :

Si c'est la première traversée, Lulu **parcourt la liste des éléments jusqu'à trouver un élément avec lequel elle peut s'éloigner de la rive.**

Une fois l'élément trouvé, elle traverse avec l'élément.

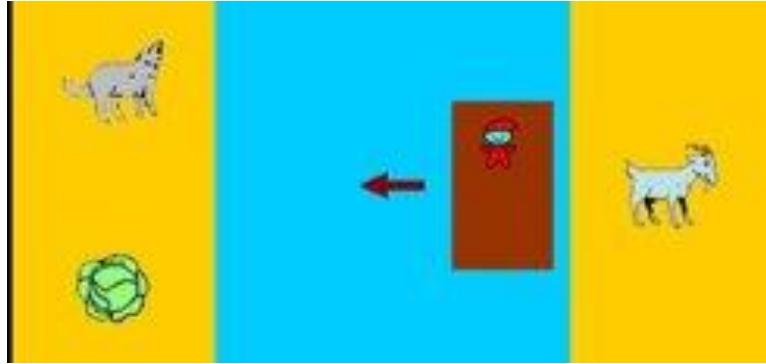
ALGORITHMIQUE



Si Lulu revient avec la chèvre, on revient à la situation précédente.

Lulu ne doit pas traverser deux fois de suite avec le même chargement.

Lulu peut-elle traverser seule ?
-> oui



Si Lulu revient avec la chèvre, on revient à la situation précédente.

Lulu ne doit pas traverser deux fois de suite avec le même chargement.

Lulu peut-elle traverser seule ?
-> oui

ALGORITHMIQUE

Tant que la situation finale n'est pas réalisée, on continue :

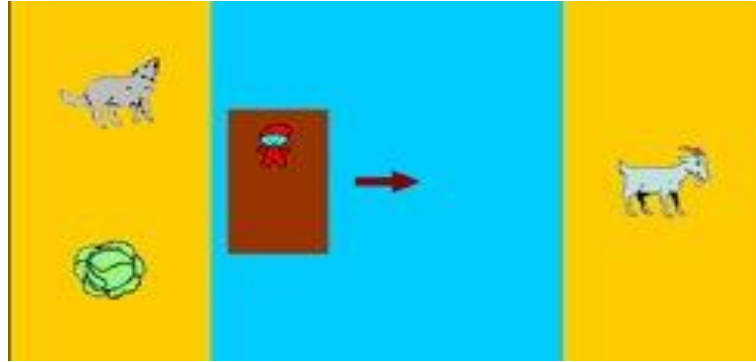
Si c'est la première traversée, Lulu parcourt la liste pour arriver à une situation valide.

Sinon,
Lulu parcourt la liste des éléments **en excluant l'élément qu'elle vient d'amener.**

Si un élément est trouvé, alors elle traverse avec.

**Sinon,
elle traverse seule.**

ALGORITHMIQUE

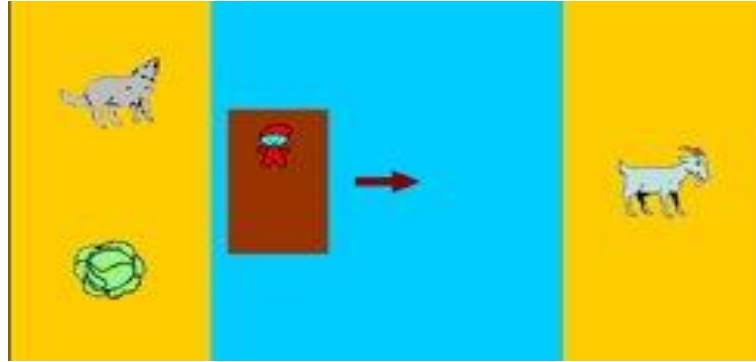


Est-ce que Lulu peut traverser
seule ?

-> Aucun intérêt (situation
précédente)

Est-ce que Lulu peut traverser
avec le loup ?

-> oui



Est-ce que Lulu peut traverser
seule ?

-> Aucun intérêt (situation
précédente)

Est-ce que Lulu peut traverser
avec le loup ?

-> oui

ALGORITHMIQUE

Tant que la situation finale n'est
pas réalisée, on continue :

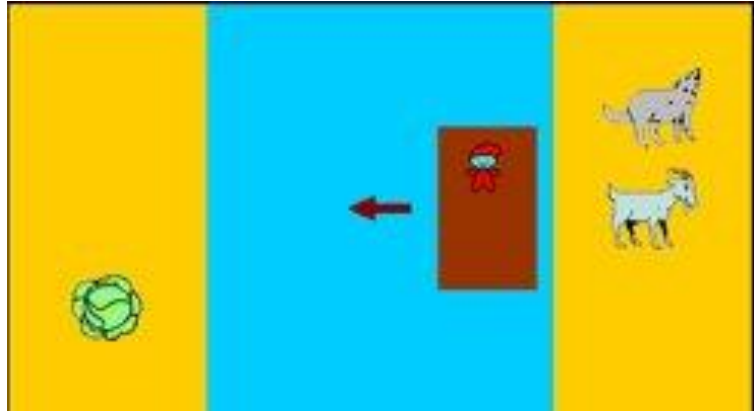
Si c'est la première traversée
Lulu parcourt la liste pour arriver à une
situation valide.

Sinon,
Lulu parcourt la liste des
éléments, en excluant l'élément qu'elle
vient d'amener.

Si un élément valide est trouvé, Lulu
traverse avec.

Sinon,
elle traverse seule, à **condition qu'elle
n'ait pas traversé seule à l'étape
précédente.**

ALGORITHMIQUE



Lulu peut-elle traverser seule ?

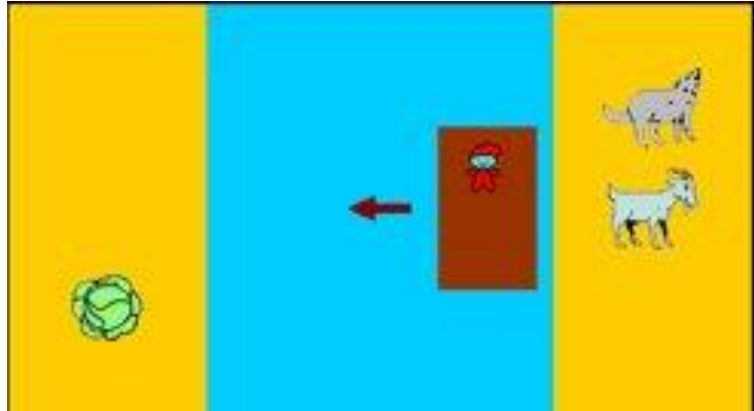
-> non (situation invalide)

Lulu peut-elle traverser avec le
loup ?

-> non (situation précédente)

Lulu peut-elle traverser avec la
chèvre ?

-> oui



Lulu peut-elle traverser seule ?
-> non (situation invalide)

Lulu peut-elle traverser avec le
loup ?
-> non (situation précédente)

Lulu peut-elle traverser avec la
chèvre ?
-> oui

ALGORITHMIQUE

Tant que la situation finale n'est pas réalisée,
on continue :

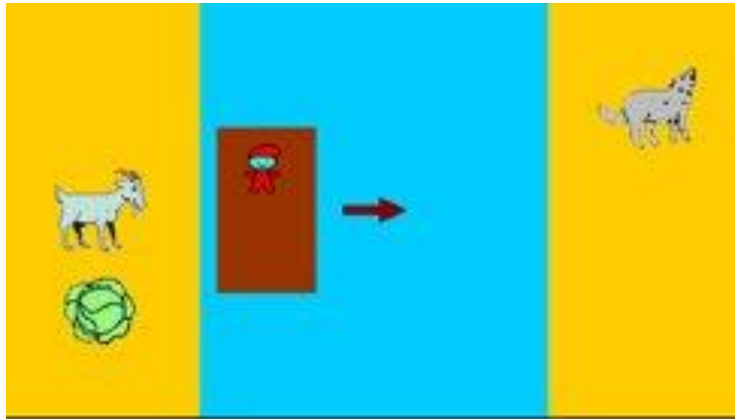
Si c'est la première traversée Lulu parcourt
la liste pour arriver à une situation valide.

Sinon,
Lulu parcourt la liste
des éléments, en excluant l'élément
qu'elle vient d'amener.

Si un élément valide est trouvé, Lulu traverse
avec.

Sinon,
elle traverse seule, à condition qu'elle n'ait
pas traversé seule à l'étape précédente.

ALGORITHMIQUE



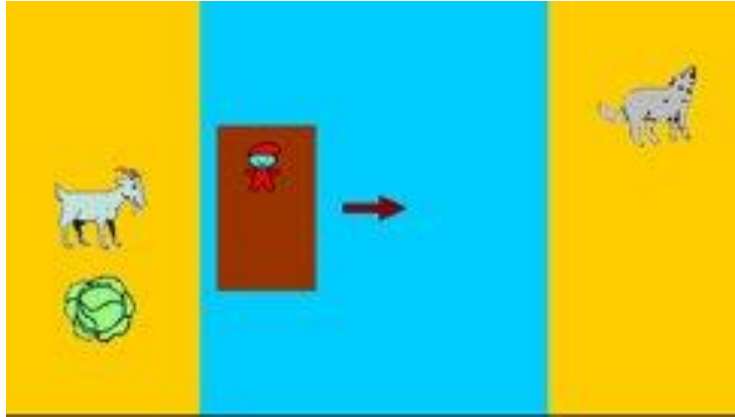
Lulu peut-elle traverser seule ?

-> non (situation non valide)

Lulu peut-elle traverser avec la
chèvre ?

-> non (situation précédente)

Lulu traverse avec le choux



Lulu peut-elle traverser seule ?
-> non (situation non valide)

Lulu peut-elle traverser avec la
chèvre ?
-> non (situation précédente)

Lulu traverse avec le choux

ALGORITHMIQUE

Tant que la situation finale n'est pas réalisée,
on continue :

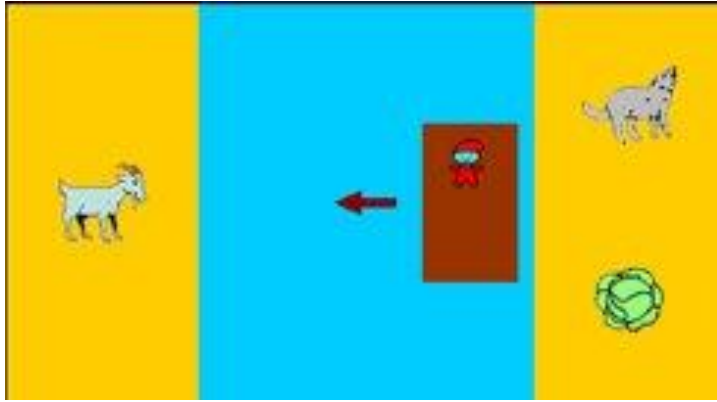
Si c'est la première traversée Lulu parcourt
la liste pour arriver à une situation valide.

Sinon,
Lulu parcourt la liste
des éléments, en excluant l'élément
qu'elle vient d'amener.

Si un élément valide est trouvé, Lulu traverse
avec.

Sinon,
elle traverse seule, à condition qu'elle n'ait
pas traversé seule à l'étape précédente.

ALGORITHMIQUE



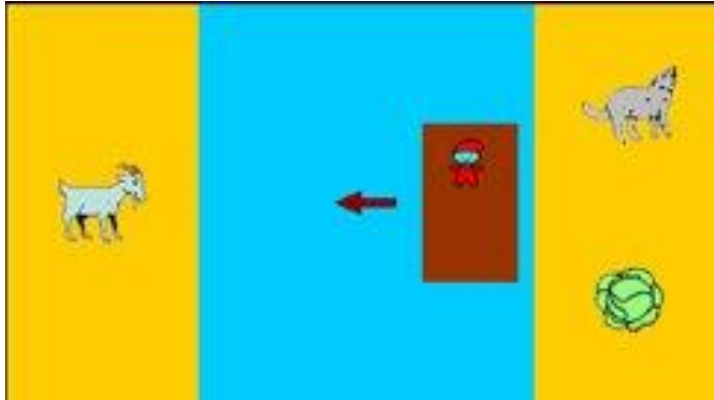
Lulu peut-elle traverser avec le
choux?

-> non (situation précédente)

Lulu peut-elle traverser avec le
loup ?

Elle pourrait, mais problème :
Lulu tourne en boucle.

Il faut vérifier en priorité si Lulu
peut traverser seule.



Lulu peut-elle traverser avec le choux?

-> non (situation précédente)

Lulu peut-elle traverser avec le loup ?

Elle pourrait, mais problème : Lulu tourne en boucle.

Il faut vérifier en priorité si Lulu peut traverser seule.

ALGORITHMIQUE

Tant que la situation finale n'est pas réalisée, on continue :

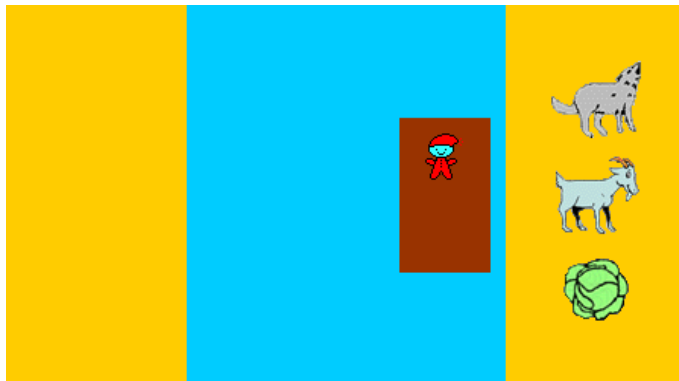
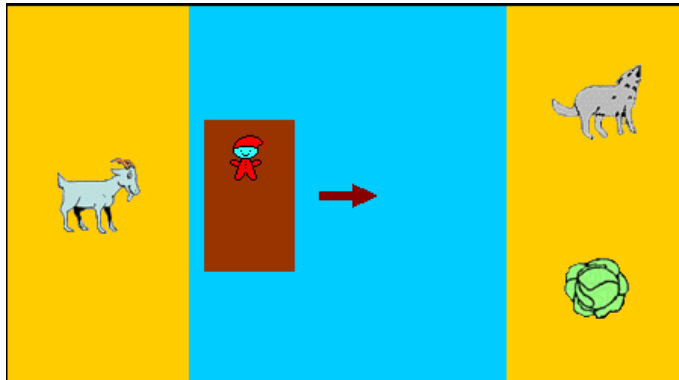
Si c'est la première traversée
Lulu parcourt la liste pour arriver à une situation valide.

**Sinon, Lulu peut-elle traverser seule
(pas de situation non valide ET elle ne vient pas de traverser seule) ?
Auquel cas elle traverse seule.**

Sinon, Lulu parcourt la liste des éléments en excluant l'élément qu'elle vient d'amener

Si un élément valide est trouvé, Lulu traverse avec.

ALGORITHMIQUE



La situation finale n'est pas réalisée, on a terminé

Tant que la situation finale n'est pas réalisée, on continue :

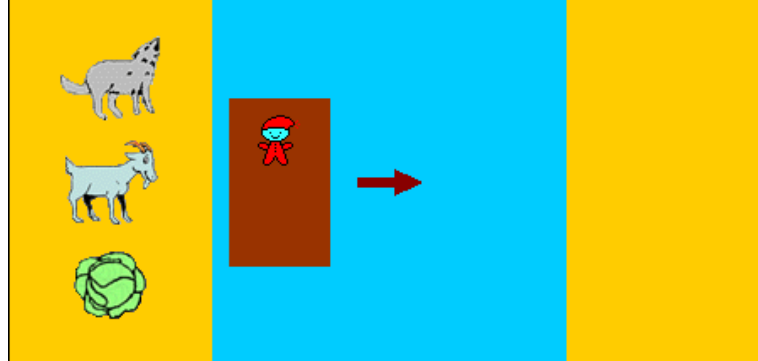
Si c'est la première traversée
Lulu parcourt la liste pour arriver à une situation valide.

Sinon, Lulu peut-elle traverser seule (pas de situation non valide ET elle ne vient pas de traverser seule) ?
Auquel cas elle traverse seule.

Sinon, Lulu parcourt la liste des éléments en excluant l'élément qu'elle vient d'amener

Si un élément valide est trouvé, Lulu traverse avec.

ALGORITHMIQUE



Pour être complet, il faut valider
l'algorithme fini depuis la
situation initiale

Tant que la situation finale n'est
pas réalisée, on continue :

Si c'est la première traversée
Lulu parcourt la liste pour arriver à une
situation valide.

Sinon, Lulu peut-elle traverser seule (pas
de situation non valide) ET elle ne vient
pas de traverser seule?
Auquel cas elle traverse seule.

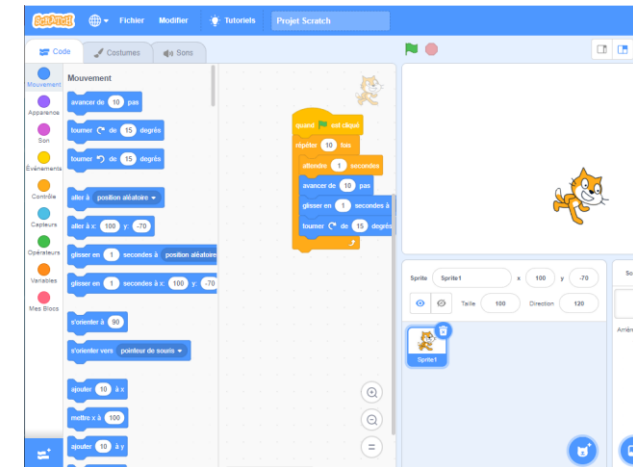
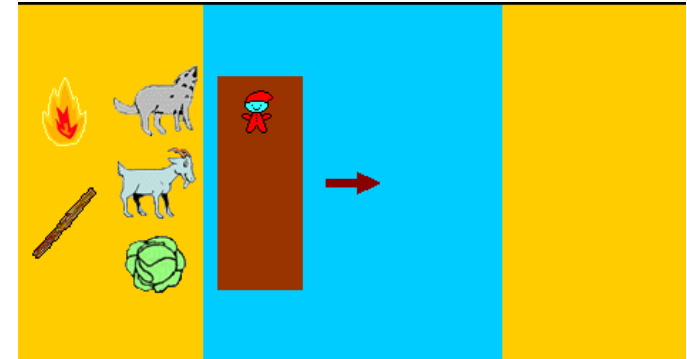
Sinon, Lulu parcourt la liste des éléments
en excluant l'élément qu'elle vient
d'amener
Si un élément valide est trouvé, Lulu
traverse avec.

ALGORITHMIQUE

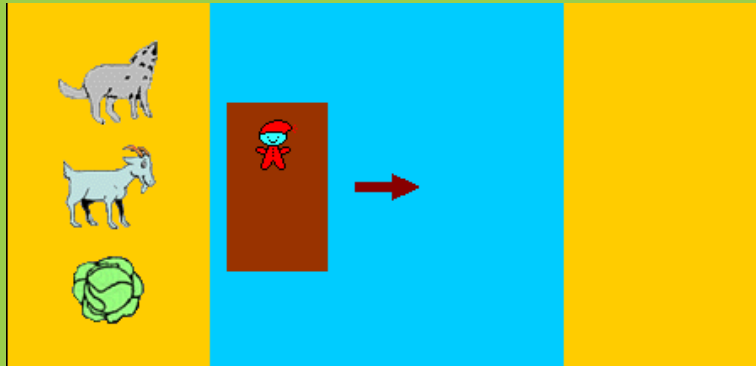
POUR ALLER PLUS LOIN

Dans le même style, un peu plus compliqué : Le feu brûle le baton, le baton bat le loup, le loup mange la chèvre qui mange le choux si Lulu s'éloigne. Il y a deux place possible sur le bateau.

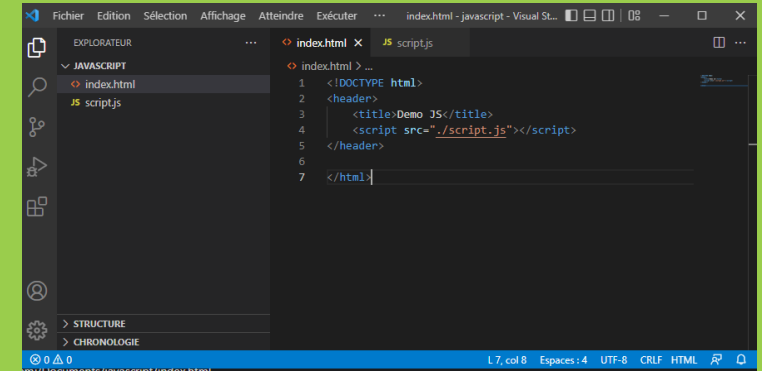
Logiciel Scratch



EN RÉSUMÉ



- 1-Nommage des entités essentielles
- 2-Rassembler dans des listes les éléments similaires utilisés
- 3-Définir la situation finale et la comparer à la situation en cours
- 4-Définir les situations non valides.
- 5-Définir les fonctions de certains éléments
- 6-Reboucler tant que la situation finale n'est pas atteinte.



- 1- Les variables
- 2-Type de données (tableaux, objets)
- 3-Expression
- 4-Conditions
- 5-Les fonctions
- 6-Boucles