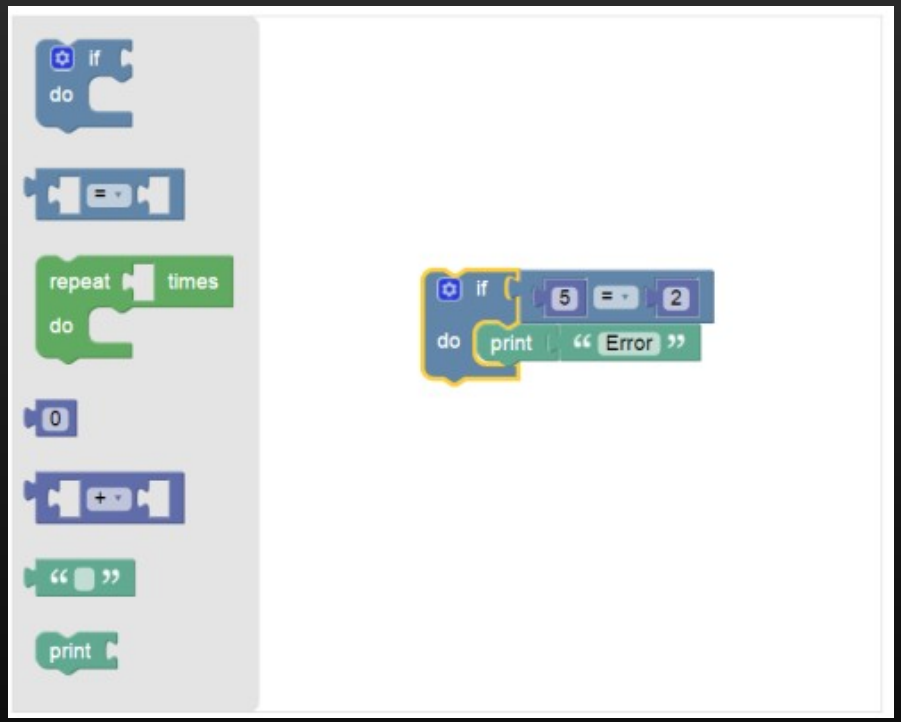
Langage visuel pour un exerciseur

Sommaire

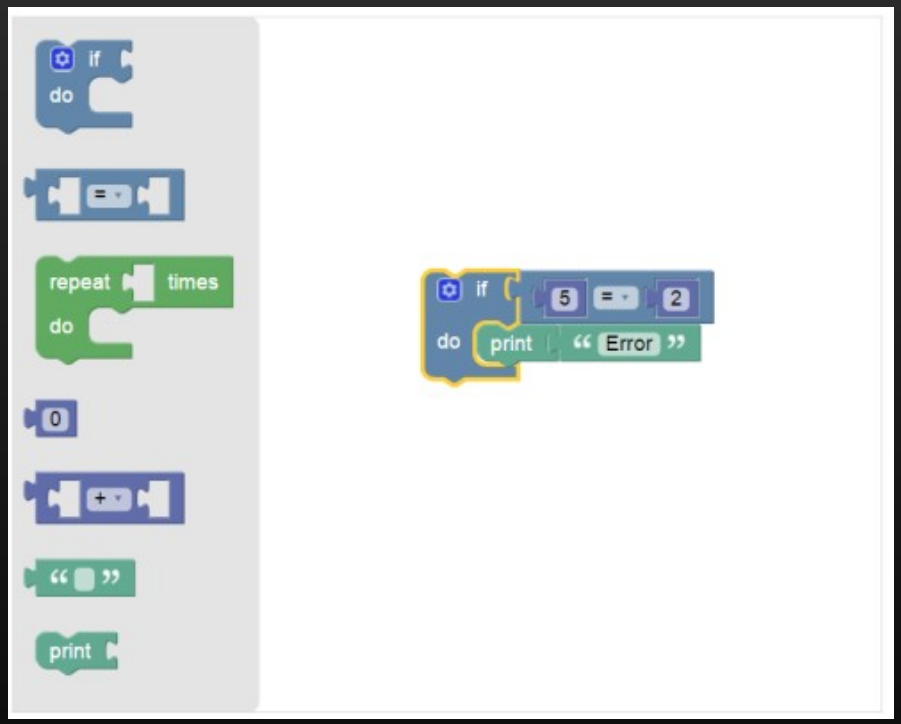
-
- Contexte
-
- Développement
-
- Conclusion

Contexte

- Blockly
-
  - Nombreux outils
  -
  - JavaScript, Python, Dart, PhP
  -
  - Erreurs de syntaxe impossibles

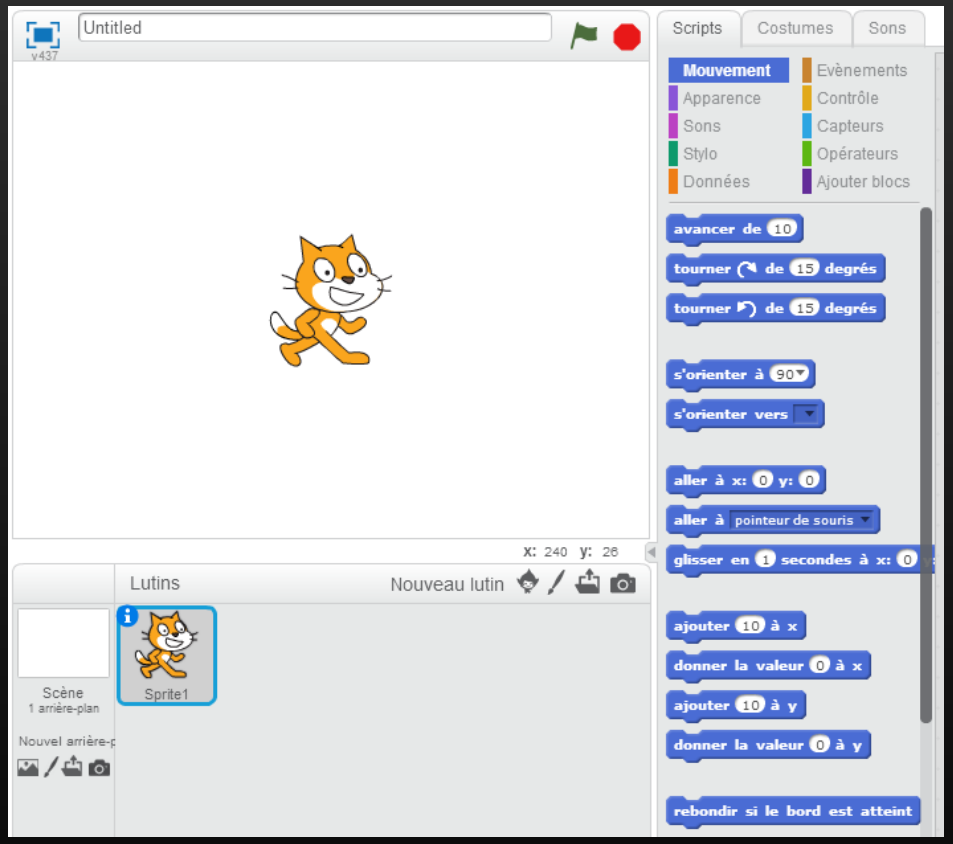


Contexte



- Blockly
-
  - Assemblage de blocks
-
  - Facile d’utilisation
-
  -

Contexte



- Blockly
-
  - Assemblage de blocks
  -
  - Facile d'utilisation
-
  - Scratch
-
  -

Contexte

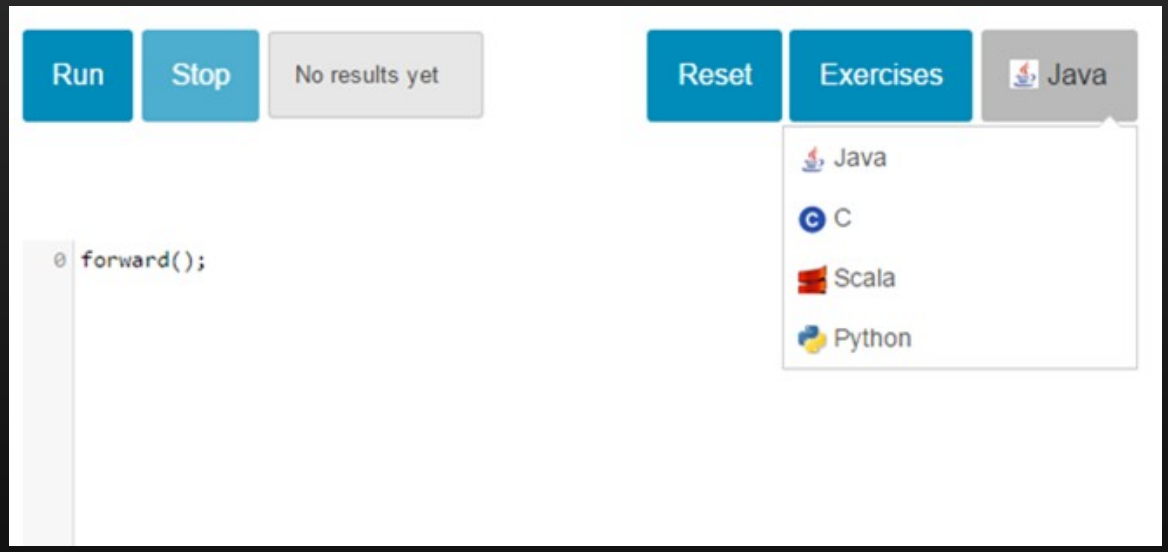
- Outils, technologie et méthodes
-
    - Eclipse / Brackets
    -
    - Java / JavaScript
    -
    - Git via GitHub

développement

- Découverte
-
    - Fonctionnement de l’environnement
    -
    - Exploration du code
    -
    - Remise à niveau FrameWork AngularJS

développement

- Insertion
-
  - Sélection
  -
  - Affichage
  -
  - Génération

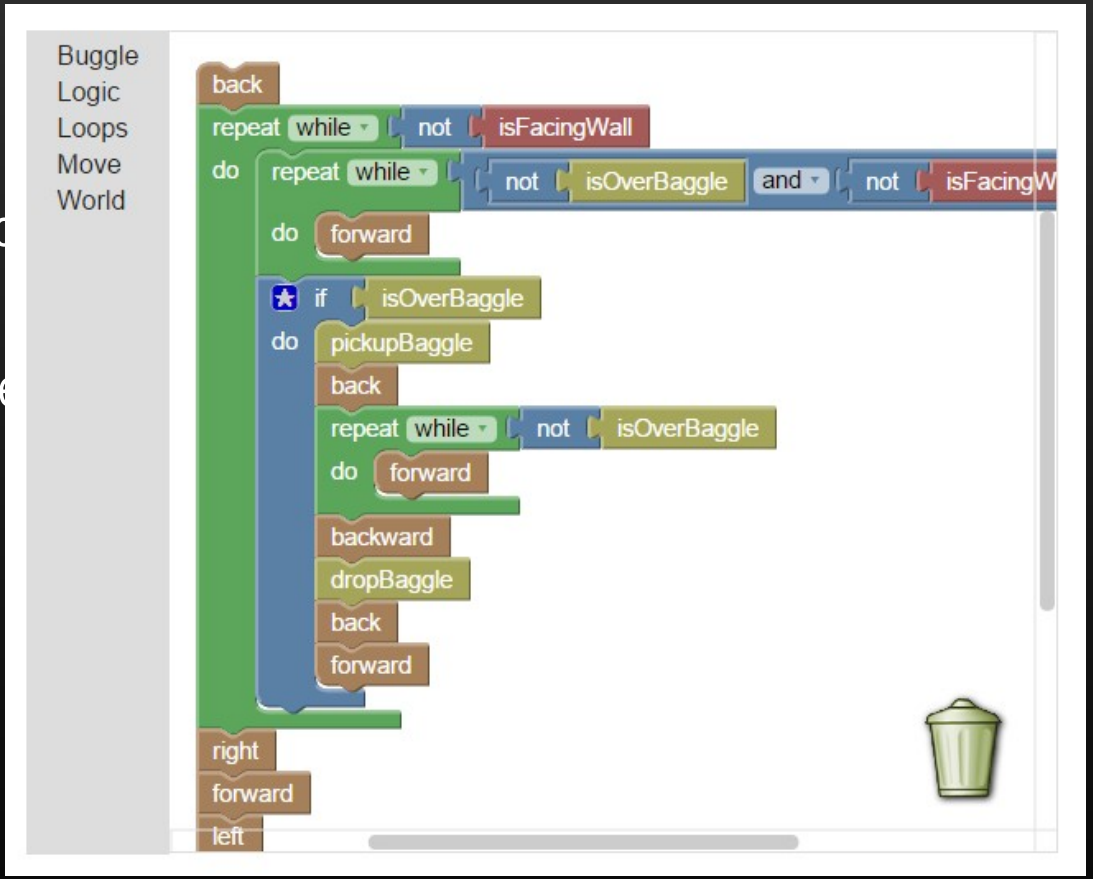


développement

- Sélection
  -
  - Python
  -
  - Deux fichiers
    - LangBlockly,java
    - *Exercise*Entity, blockly
-

développement

- Affichage
-
  - AngularJS (ng-sho
  -
  - Directive + Service
-



Développement

```
back()
while not (isFacingWall()):
  while not (isOverBaggle()) and not (isFacingWall()):
    forward()
  if isOverBaggle():
    pickupBaggle()
    back()
    while not (isOverBaggle()):
      forward()
    backward()
    dropBaggle()
    back()
    forward()
right()
forward()
left()
forward()
```

- Génération
-
    - Exercise.controller.js
    -
    - Python
    -
    -
-

développement

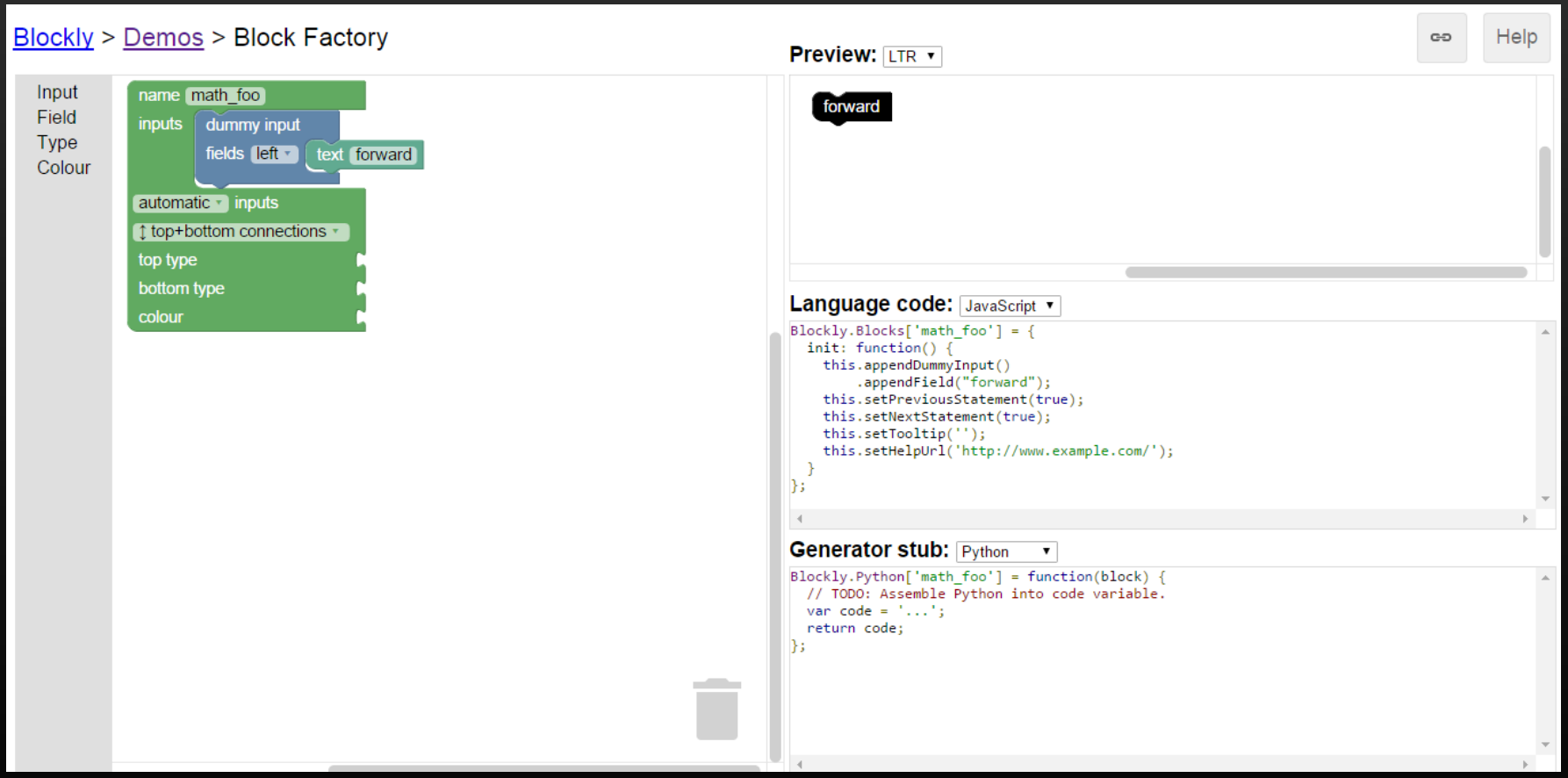
- Toolbox
-
  - Bloc
  -
  - Liaison

développement

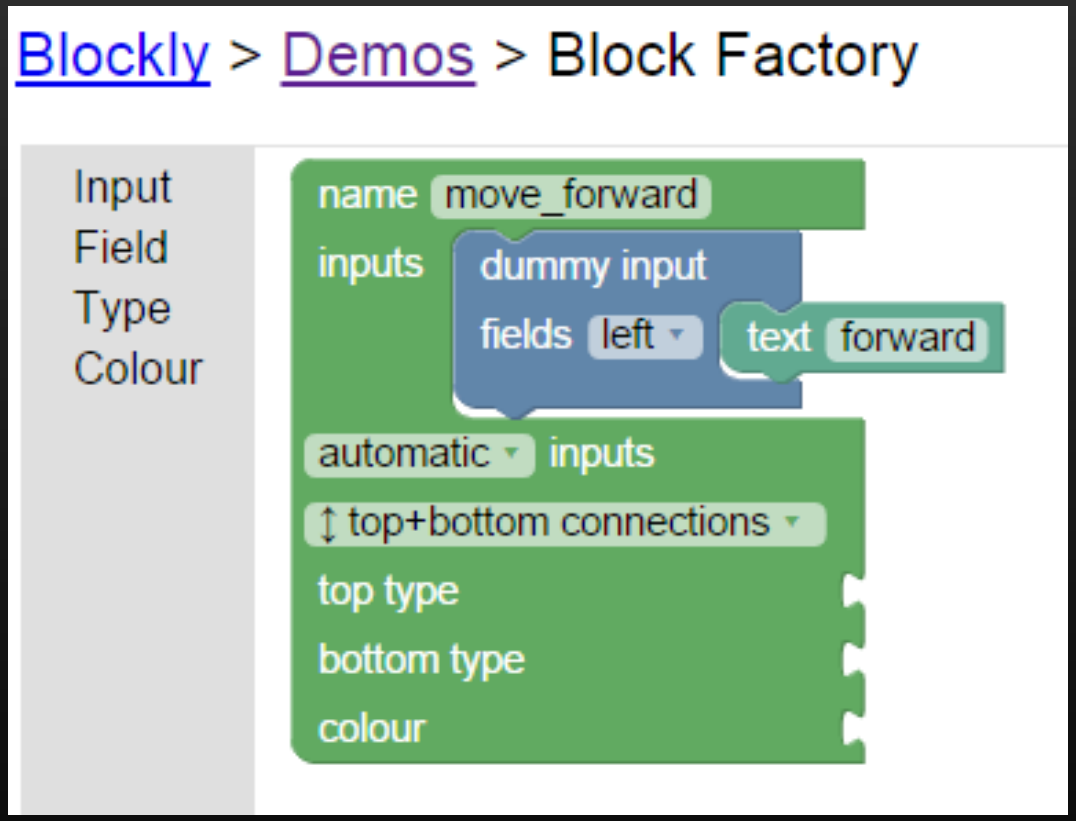
- Bloc
-
  - Nouveaux blocs
  -
  - Visuel et fonctionnel

forward()

développement



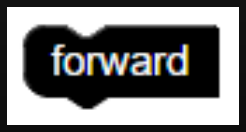
développement



développement

Language code: JavaScript

```
Blockly.Blocks['move_forward'] = {
  init: function() {
    this.appendDummyInput()
        .appendField("forward");
    this.setPreviousStatement(true);
    this.setNextStatement(true);
    this.setTooltip('');
    this.setHelpUrl('http://www.example.com/');
  }
};
```

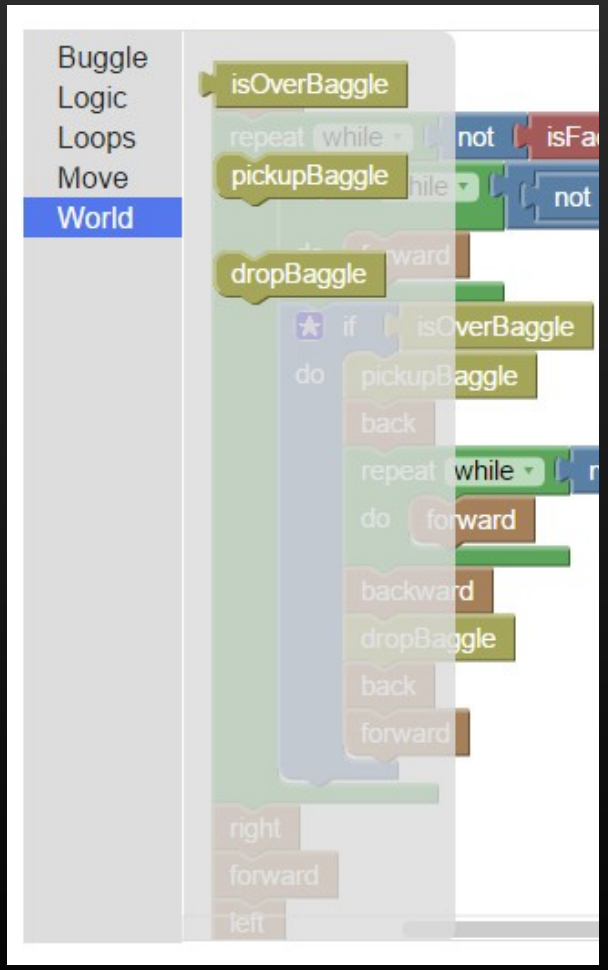


Generator stub: Python

```
Blockly.Python['move_forward'] = function(block) {
  // TODO: Assemble Python into code variable.
  var code = '...';
  return code;
};
```

développement

- Liaison
-
  - Boite d’outils / exercice
  -
  - Non obligatoire
  -
  - JSON simple



développement

- Liaison
-
    - Boite d’outils / exercice
    -
    - Non obligatoire
    -
    - JSON simple

```
[{
    "name": "Buggle",
    "blocks": [{
        "type": "buggle_facingWall"
                }]
            }, {
    "name": "Logic",
    "blocks": [{
        "type": "controls_if"
                }, {
        "type": "newlogic_operation"
                }, {
        "type": "logic_negate"
                }]
            }, {
    "name": "Loops",
    "blocks": [{
        "type": "controls_whileUntil"
                }]
            }, {
    "name": "Move",
    "blocks": [{
        "type": "move_forward"
                }, {
        "type": "move_backward"
                }, {
        "type": "turn_right"
                }, {
        "type": "turn_left"
                }, {
        "type": "turn_back"
                }]
            }, {
    "name": "World",
    "blocks": [{
        "type": "world_baggle_ground"
                }, {
        "type": "world_baggle_pickup"
                }, {
        "type": "world_baggle_drop"
                }]
}]
```

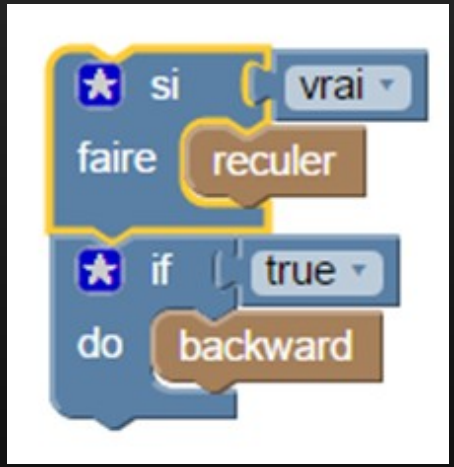
développement

- Exemple pour les blocs forward puis backward
-
- <xml xmlns=« http://www.w3.org/1999/xhtml"><block type="move_forward" id="6" x="25"><next><block type="move_backward" id="12"></block></next><block></xml>

- Stockage
-
  - Reprise du travail
  -
  - Suivi de l’utilisateur
  -
  - XML
-

développement

- Traduction
-
  - 3 systèmes
  -
  - Blockly: riche / rechargement
  -
  - Service



Conclusion

-
- Blockly fonctionnel
-
- Piste de poursuite de développement
-
- Complément d’expérience vis-à-vis de l’IUT

Merci de votre attention

Des questions ?