



Document ressource « Turing Tumble »

Formation « Informatique, on débranche! »



Document ressource « Turing Tumble »

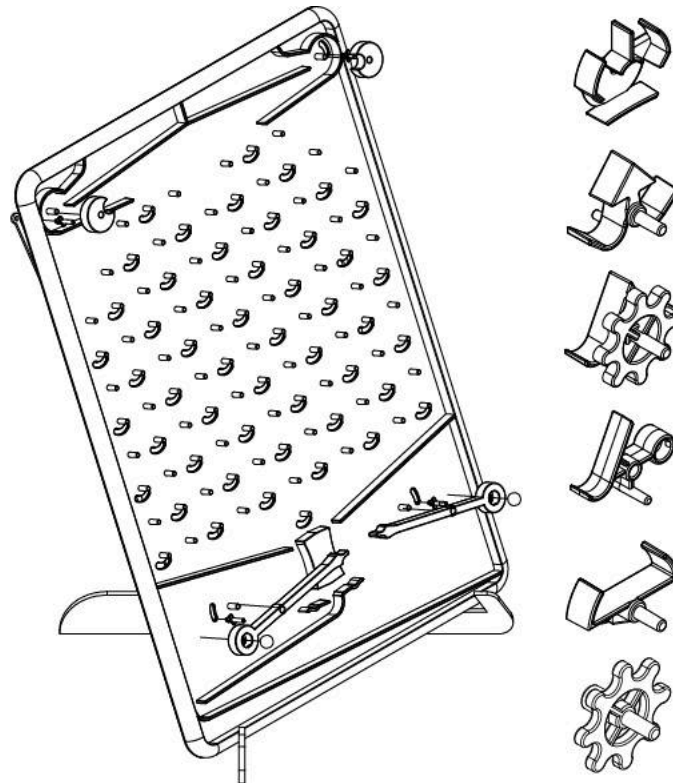
IREM de Lille - Université de Lille

1 Quelques explications (ou rappels)

Turing Tumble est un jeu et une démonstration de portes logiques via ordinateur mécanique.

Turing Tumble comprend les "pièces" suivantes :

- "**Des billes**". La version standard utilise deux rampes qui stockent un nombre donné de billes. Un interrupteur en bas du plateau déclenche la libération de la boule initiale (généralement bleue), depuis le haut à gauche du panneau. La deuxième rampe, à droite, contient des boules rouges.
- "**Rampes**" et "**croisements**". La rampe verte permet aux billes de la descendre dans un sens et de la relâcher uniquement dans cette direction, tandis que le croisement orange permet aux billes de la traverser de chaque côté dans les deux sens, c'est-à-dire de droite à gauche et vice versa .
- "**Intercepteurs**". Cette pièce noire arrête une bille.
- "**Bits**". Il s'agit d'un stockage à un bit : il change de direction lorsqu'une bille roule, de sorte que la bille suivante passe de l'autre côté.
- "**Roue dentée**" et "**Roue dentée binaire**". Les roues dentées binaires sont exactement comme des "bits" ordinaires, mais ils peuvent être connectés à des roues dentées. Les roues dentées permettent de lier les changements d'état, ajoutant ainsi intégralement une puissance supplémentaire.



2 Focus sur les activités

2.1 Activité 1 : Alternance

2.1.1 Objectif

Obtenir le motif bleu, rouge, bleu, rouge, bleu, rouge...



2.1.2 Éclairage

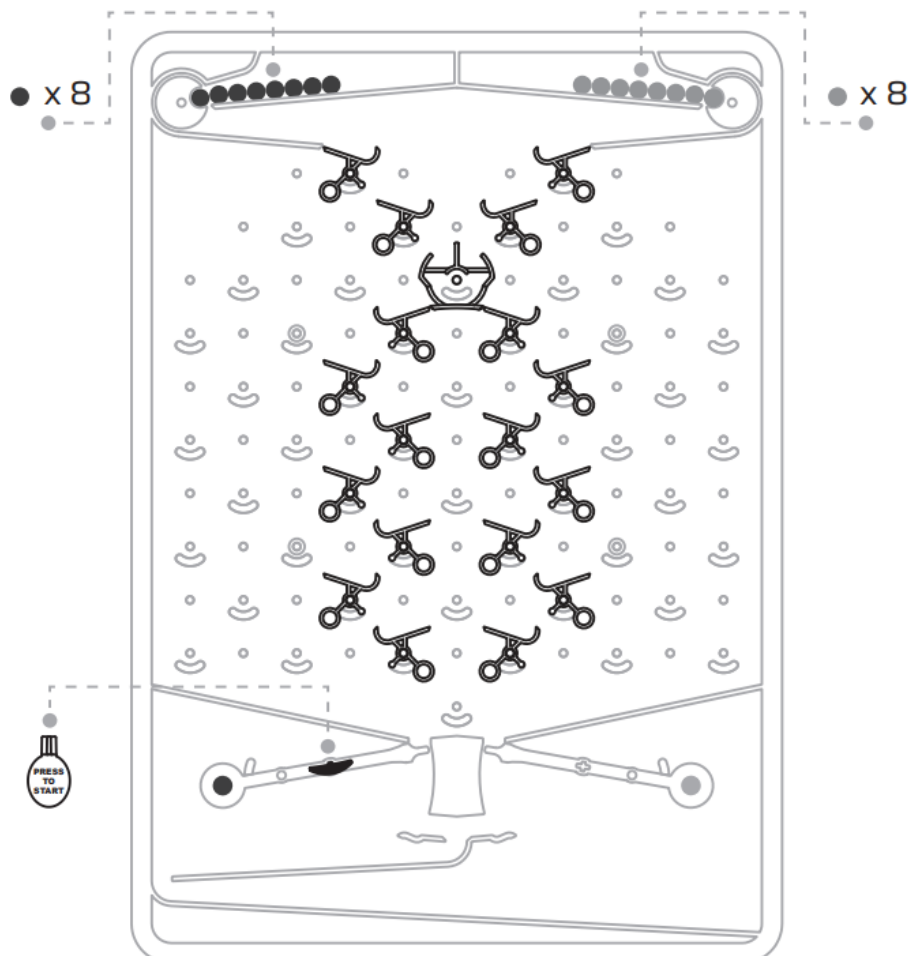
Dans cette activité, l'objet "crossover" est introduite. Il ressemble à ceci :



Le but du croisement est de laisser les chemins de billes se croiser. Une bille entrant du côté gauche sort du côté droit et inversement.

2.1.3 Solution expliquée

Il fallait créer votre propre chemin pour les billes rouges pour passer au levier gauche ! Vous devez utiliser le croisement où ton chemin croise le chemin des billes bleues.



2.2 Activité 2 : Boucle "if"

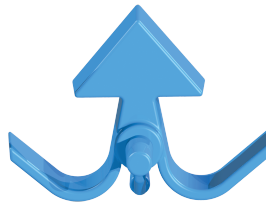
2.2.1 Objectif

Si la machine démarre avec le bit A pointant vers la gauche, intercepter une boule bleue. Sinon, interceptez une boule rouge.



2.2.2 Éclairage

Dans cette activité, l'objet "bit" est introduite. Il ressemble à ceci :



Ces bits sont des versions mécaniques des interrupteurs électroniques à l'intérieur des puces informatiques. Les interrupteurs électroniques vous permettent de choisir dans quel sens l'électricité ira en fonction de la façon dont vous les définissez. Ces interrupteurs mécaniques vous permettent de choisir la direction dans laquelle la bille roulera en fonction de direction dans laquelle ils sont pointés.

Lorsque les bits sont utilisés pour stocker des informations, ils sont comme des variables dans un langage de programmation. Bien sûr, un bit ne stocke pas beaucoup d'informations, juste un 1 ou un 0. Mais lorsque vous combinez plusieurs bits en une seule variable, ils peuvent représenter des chiffres, des lettres ou... n'importe quoi d'autre.

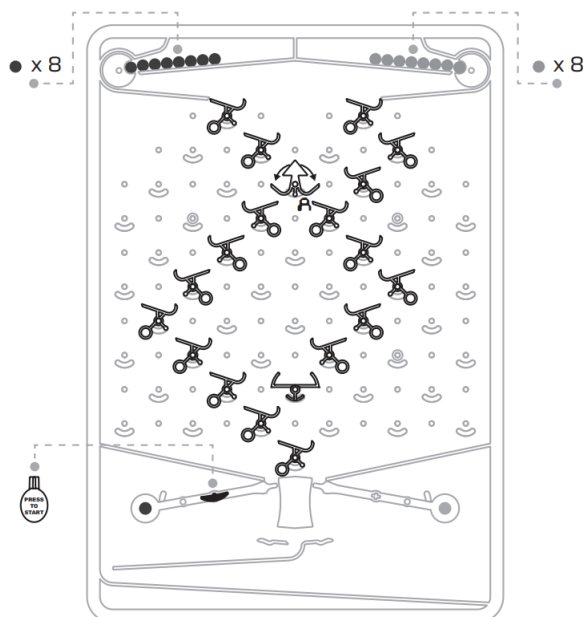
Les bits fonctionnent également comme la commande de programmation la plus fondamentale de toutes : l'instruction "si". On peut se le représenter de la sorte :

```
If (bit.direction = right) Then
    ball.send_left()
Else
    ball.send_right()
```

2.2.3 Solution expliquée

Analysons les deux situations de départ séparément :

- Le bit A commence à gauche : la première bille bleue tout simplement tombe dans l'intercepteur.
- Le bit A commence à droite : la première bille bleue tombe sur le levier du côté droit de la machine, libérant une bille rouge qui atterrit dans l'intercepteur.



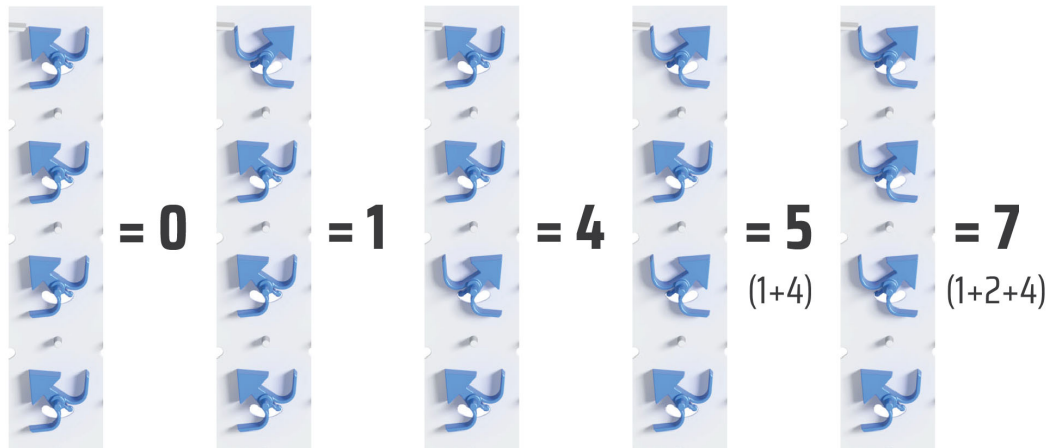
2.3 Activité 3 : Comptage

2.3.1 Objectif

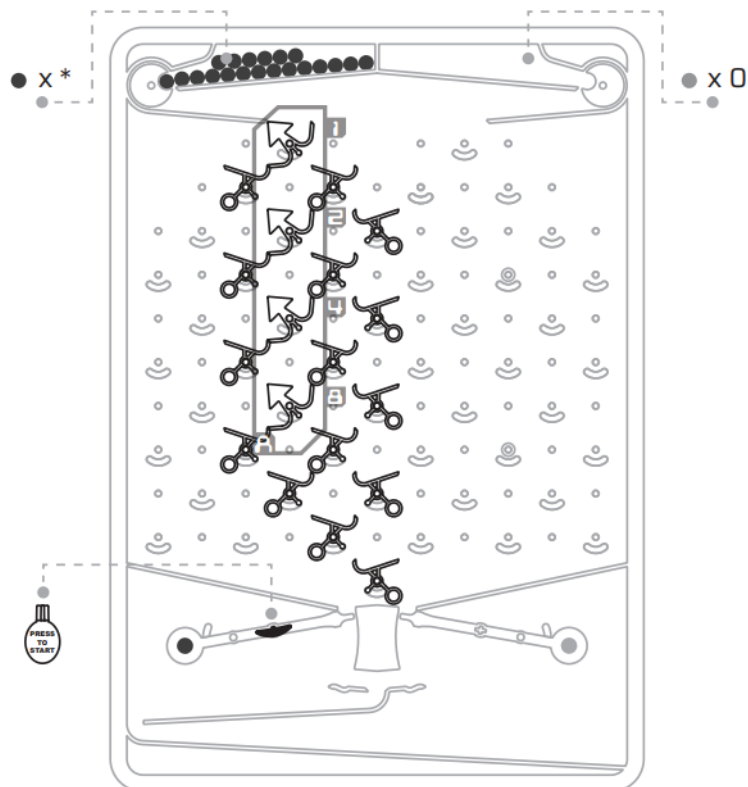
Réaliser un compteur du nombre de boules bleues tombées dans le bas du tableau.

2.3.2 Éclairage

À l'aide de quatre bits, il est possible d'exprimer différents nombres, comme dans l'image présentée ici. Pour comprendre le nombre que représente la série de bits, il faut simplement retenir que le premier bit vaut 1, le deuxième vaut 2, le troisième vaut 4, et le dernier vaut 8. Il suffit ensuite d'additionner la valeur des bits qui pointent vers la droite. Par exemple :



2.3.3 Solution expliquée



3 Et après ? Qu'est-ce qu'on peut faire d'autre ?

Il faut comprendre que la seule limite est la taille du tableau ! Si l'on avait un tableau un peu plus grand ¹, Turing Tumble serait aussi puissant qu'un smartphone.

Avec *Turing Tumble*, on peut :

- Compter
- Faire des maths
- Comparer
- Créer des séquences
- Illustrer une logique
- et bien plus encore !

4 Références et ressources en ligne

- github.com/jessecrossen/ttsim Il s'agit d'une simulation logicielle du Turing Tumble. Il s'exécute entièrement dans votre navigateur. Si vous avez utilisé un Turing Tumble réel, il devrait être simple de commencer, mais voici une documentation (jessecrossen.github.io/ttsim/usage) sur la façon de l'utiliser.
- Article de Natacha Portier « *À la découverte du jeu Turing Tumble* », interstices.info.
- « *Constructing Reversible Logic Elements on Turing Tumble Model* ». Proceedings of Automata 2018 : 25–32. 20–22 June 2018. Retrieved 2019-12-10. (NB. A longer version was published in 2019.)
- « *Universal logic elements constructed on the Turing Tumble* ». Natural Computing (Springer-Verlag) 19 (9). 2019-09-03. doi :10.1007/s11047-019-09760-8. ISSN 1567-7818. Retrieved 2020-07-27. (NB. A short version of this paper was presented at AUTOMATA 2018.)

1. de la taille de La France...