Nouhaud (Pereira) Mattéo - 22008937
Patry Simon - 22004482
Proust Jules - 22002224
Rullier Mateo - 22001944

Théorie des langages et automates

Projet

Rendu Jalon 2

Ce document était dans une archive .zip, accompagné d'un fichier tableur, ce-dernier contient les annexes nécessaires à la compréhension de ce rendu, ainsi qu'un fichier zip à ouvrir avec votre IDE de prédilection.

Ajout – Modification par rapport au Jalon 1

Lexique

Le lexique de notre langage s'est agrandi afin d'ajouter les nouveaux mots nécessaires :
(en vert les nouveaux mots, en rouge les mots supprimés)
Nouveau lexique :

,	EXIT	OFF	SWITCH
;	FROM	ON	TO
(	GHOST	OPEN	TP
)	GOING	OR	U | UP
ALL	GOTO	PLACE	VISIBLE
AND	INVISIBLE	PLAYER	WALL
CLOSE	IS	R | RIGHT	WHEN
CREATE	L | LEFT	REPEAT	X
D | DOWN	LEVEL	SET	Y
DOOR	LOOP	SIZE	num
EXCEPT	NAMED	SPAWN	str

Ancient lexique :

,	EXCEPT	LOOP	SPAWN
;	EXIT	ON	SWITCH
(	FROM	OPEN	TO
)	GHOST	OR	TP
ALL	GOING	PLACE	U | UP
AND	GOTO	PLAYER	WALL
CLOSE	IS_OFF	R | RIGHT	WHEN
CREATE	IS_ON	REPEAT	X
D | DOWN	L | LEFT	SET	Y
DOOR	LEVEL	SIZE	num

L'analyse lexicale corrigées est disponible dans la première feuille du tableur *projet.xlsx* ci-joint.

Motif

Si notre langage s'est agrandi, c'est pour accueillir des fonctionnalités, ici deux :

- La possibilité de nommé le niveau, afin que son nom s'affiche sur le menu de sélection.
- La possibilité de choisir si l'on veut afficher ou non les trappes

Mais aussi à des fins de correction :

- Dans notre première version, nous avions prévu que IS ON et IS OFF ne seraient que dans un seul mot (IS_ON et IS_OFF) de peur que le mot ON ne pose problème de par sa réutilisation. Lors du développement, nous avons réalisé qu'il n'y aurait pas de problème, et avons remis IS ON et IS OFF en deux mots, ce qui à aussi permis de simplifier l'intégration de IS VISIBLE et IS INVISIBLE.

Grammaire et table d'analyse

Les 2e et 3e feuilles du tableur *projet.xlsx*, ci-joint, contiennent les versions modifiées de l'analyse syntaxique et sa table.

Ci-dessous, la version modifiée de la grammaire de notre langage, *les différents symboles non-terminaux sont matérialisés ici en bleu (pour le lien avec le tableur, les nombres font foi), les symboles terminaux en orange, les modifications par rapport au jalon 1 sont indiquées en blanc*:

```
// Base
    000S → CREATE LEVEL 003 ; 001player 002place

    001player → *
    001player → SET PLAYER ON 900position ;

    002place → *
    002place → PLACE 005entity ; 002place

------- Ajout nommage du niveau non-obligatoire avant ou après les dimensions ------
    003 → SIZE 900position 004
    003 → NAMED str SIZE 900position

    004 → *
    004 → NAMED str
--------------------------------------------------------------------------------

    005entity → WALL ON 100positionGroup
    005entity → EXIT ON 100positionGroup
    005entity → TP 030tp
    005entity → GHOST 040ghost
    005entity → SWITCH num 050switch
    005entity → DOOR 060door

    // Différents ordonnancement des parties

- Ajout visibilité de la trappe INVISIBLE par défaut dans tous les ordonnancements -
        030tp → ON 100positionGroup 301
        030tp → FROM 920rightLeftDownUp 302
        030tp → GOTO 900position 303
        030tp → SET IS 915visInvis 304

        301 → FROM 920rightLeftDownUp 305
```

```
301 → GOTO 900position 306
301 → SET IS 915visInvis 307

302 → ON 100positionGroup 305
302 → GOTO 900position 308
302 → SET IS 915visInvis 309

303 → ON 100positionGroup 306
303 → FROM 920rightLeftDownUp 308
303 → SET IS 915visInvis 310

304 → ON 100positionGroup 307
304 → FROM 920rightLeftDownUp 309
304 → GOTO 900position 310

305 → GOTO 900position 916setIsVisInvis
305 → SET IS 915visInvis GOTO 900position

306 → FROM 920rightLeftDownUp 916setIsVisInvis
306 → SET IS 915visInvis FROM 920rightLeftDownUp

307 → GOTO 900position FROM 920rightLeftDownUp
307 → FROM 920rightLeftDownUp GOTO 900position

308 → ON 100positionGroup 916setIsVisInvis
308 → SET IS 915visInvis ON 100positionGroup

309 → GOTO 900position ON 100positionGroup
309 → ON 100positionGroup GOTO 900position

310 → FROM 920rightLeftDownUp ON 100positionGroup
310 → ON 100positionGroup FROM 920rightLeftDownUp
```

```
040ghost → 401 041
040ghost → SPAWN ON 900position 042
040ghost → LOOP 043

041 → SPAWN ON 900position 450
041 → LOOP SPAWN ON 900position

042 → 401 450
042 → LOOP 401

043 → SPAWN ON 900position 401
043 → 401 SPAWN ON 900position
```

```
        050switch → ON 900position 501
        050switch → SET IS 910onOff ON 900position

        060door → ON 100positionGroup 601
        060door → 602 ON 100positionGroup

// Général

    900position → X 901
    900position → Y 902

    901 → num Y num
    901 → Y ( num num )

    902 → num X num
    902 → X ( num num )

--- Modif 910: Remplacement IS_ON IS_OFF par ON OFF, ajout IS devant chaque appel ---
    910onOff → ON
    910onOff → OFF
-------------------------------------------------------------------------------

---------------------------- Ajout VISIBLE | INVISIBLE ----------------------------
    915visInvis → VISIBLE
    915visInvis → INVISIBLE

    916setIsVisInvis → *
    916setIsVisInvis → SET IS 915visInvis
-------------------------------------------------------------------------------

    920rightLeftDownUp → R|RIGHT
    920rightLeftDownUp → L|LEFT
    920rightLeftDownUp → D|DOWN
    920rightLeftDownUp → U|UP

    930orAnd → AND
    930orAnd → OR

// Partie ON ...

--- Ajout de l'ajout ou de la suppression de positions (105) directement dans 100 ---
    100positionGroup → ALL 105moreLessPosition
    100positionGroup → X 101 105moreLessPosition
    100positionGroup → Y 102 105moreLessPosition
-------------------------------------------------------------------------------

    101 → 120 140
```

```
    101 → Y ( num num 131 )

    102 → 120 141
    102 → X ( num num 131 )

    105moreLessPosition → *
    105moreLessPosition → ON 100positionGroup
    105moreLessPosition → EXCEPT 100positionGroup

    120 → num
    120 → ( num 121 )

    121 → *
    121 → , num 121
    121 → TO num 122

    122 → *
    122 → , num 121

    131 → *
    131 → , num num 131

    140 → *
    140 → Y 120

    141 → *
    141 → X 120

// Partie TP

// Partie GHOST

    401 → GOING 402
    401 → GOTO 900position

    402 → 404
    402 → ( 404 403 )

    403 → *
    403 → , 404 403

    404 → 920rightLeftDownUp
    404 → REPEAT 920rightLeftDownUp num

    450 → *
    450 → LOOP
```

```
// Partie SWITCH

    501 → *
    501 → SET IS 910onOff

// Partie DOOR

    601 → *
    601 → 602

    602 → OPEN WHEN 603
    602 → CLOSE WHEN 603

    603 → ( 605 ) IS 910onOff 604
    603 → SWITCH num IS 910onOff 604
    603 → IS 910onOff 607 604

    604 → *
    604 → 930orAnd 603

    605 → SWITCH num 606
    605 → ( 605 ) 606

    606 → *
    606 → 930orAnd 605

    607 → ( 605 )
    607 → SWITCH num
```

La grammaire peut aussi être caractérisée de manière plus lisible, avec en noir les nouvelles fonctionnalités :

```
{ ?1 | ?2 } = ?1 OU ?2
{ ?1 <> ?2 } = ?1 ET ?2 , peu importe l'ordre
[ ? ] = ? pas obligatoire
[? ...] = se répète autant de fois que nécessaire, avec un ? entre chaque occurrence

CREATE LEVEL { SIZE coordSimple <> [ NAMED str ] };
[ SET PLAYER ON coordSimple ; ]
{ PLACE { WALL | EXIT } ON coordCompl ;
| PLACE TP { ON coordCompl <> FROM side <> GOTO coordSimple <> [ SET IS { VISIBLE |
INVISIBLE } ] } ;
| PLACE GHOST { SPAWN ON coordSimple <> { GOING {side|(side [, ...])} | GOTO
coordSimple } <> [LOOP] } ;
| PLACE SWITCH num { ON coordSimple <> [ SET IS { ON | OFF } ] } ;
| PLACE DOOR { ON coordCompl <> [ { OPEN | CLOSE } WHEN cond ] } ;
} [...]
```

```
où coordSimple est :
    { { XY | YX } ( num num ) | { Xnum <> Ynum } }

où coordCompl est :
    { ALL | { XY | YX } ( num num [, ...] ) | { X { num | ( num [, ...] ) } <> Y {
num | ( num [, ...] ) } } | { X | Y } { num |( num [, ...] ) } } } [{ON|EXCEPT} ...]

où side est :
    { R | RIGHT | L | LEFT | D | DOWN | U | UP }

où cond est :
    { {IS ON|IS OFF} <> { SWITCH num | ( SWITCH num [{AND|OR} ...] ) } } [{AND|OR}
...]
    La parenthèse de la condition peut être récursive, peut être accepter :
        IS ON (SWITCH 1 AND (SWITCH 2 OR SWITCH 3))
```

Java

Contacts

Nouhaud (Pereira) Mattéo : matteo.nouhaud@etu.univ-tours.fr
Patry Simon : simon.patry@etu.univ-tours.fr
Proust Jules : jules.proust@etu.univ-tours.fr
Rullier Mateo : mateo.rullier@etu.univ-tours.fr