Grammaire :

<script> : := ‘SCRIPT’ { <instruction> } ‘FIN’

<instruction> : := <action> | <control>

<action> : := ‘AVANT’ <entier>
| ‘DROITE’ <entier>
| ‘GAUCHE’ <entier>
| ‘ALLERA’ <entier> <entier>
| ‘POSER’
| ‘LEVER’
| ‘COULEUR’ <couleur>
| ‘EPAISSEUR’ <entier>

<couleur> : := ‘NOIR’
| ‘BLANC
| ‘GRIS’
| ‘BLEU’
| ‘VERT’
| ‘ROUGE’
| ‘JAUNE’
| ‘ROSE’
| ‘ORANGE’
| ‘VIOLET’
| ‘MARRON’

<entier> : := (‘0’ | … | ‘9’) { ‘0’ | … | ‘9’ }

<control> : := ‘REPETER’ <expression-arithm> <script>
| ‘SI’ <condition> ‘ALORS’ <script> | SINON <script>
| ‘TANTQUE’ <condition> <script>

<condition> : := <condition-simple>
| ‘(‘ <expression-booleenne> ‘)’

<expression-booleenne> : := <condition>

<condition-simple> : := ‘ESTLEVE’
| <requete-position> <op-compare> <expression-arithm>
| <requete-graphique>

<requete-position> : := ‘POSX’ | ‘POSY’

<expression-arithm> : := <entier> | <operator>

<op-compare> : := ‘==’
| ‘<’
| ‘<=’
| ‘>’
| ‘>=’

<requete-graphique> : := ‘POSCOLEUR’ <couleur>

<operator> : := ‘+’
| ‘*’
| ‘-’
| ‘/’

<variables> : := ‘VAR’ <entier> | <boolean>
| ‘VAR =’ <entier> | <boolean>

<boolean> : := (‘true’ | ‘false’) | { ‘true’ | ‘false’ }

<appel-variable> : := ‘$VAR’

<function> : := ‘FONCTION NAME()’ { <instruction> } ‘FINFONCTION’

<run-function> : := ‘RUN NAME()’

<bezier> : := ‘BEZIER’ <entier> <entier> <entier> <entier> <entier> <entier>

En rouge : Implanté dans le projet mais non fonctionnel.

Description du projet :

Nous avons souhaité rendre notre projet le plus accessible possible en nous mettant à la place d’un utilisateur qui débuterait avec notre logiciel.

Nous avons donc ajouté un système d’aide expliquant la plupart des commandes. Des boutons simples et explicites sont également présents pour permettre une meilleure compréhension des commandes.

D’un point de vue code, nous avons implémenté le modèle MVC pour une meilleure compréhension du code que nous avions. De plus, l’utilisation des design pattern singleton pour le crayon et visiteur pour le script nous paraissait logique.

Nous avons également ajouté la possibilité d’enregistrer les dessins et même d’en charger. Nous avons également choisi d’ajouter la possibilité de déclarer des variables et même de lancer des fonctions. Un manque de compréhension du parsing de token nous a empêché de terminer à temps les conditions, bien qu’elles soient déjà implantées et possèdent un visiteur, etc … Les expressions arithmétiques contenant les opérateurs ‘+’ et ‘/’ fonctionnent.

Nous avons en définitive essayé d’être le plus cohérent dans la définition de notre grammaire en utilisant des mots-clefs simples que tous peuvent comprendre. En outre, nous avons essayé d’étendre la grammaire au maximum pour incorporer le plus de fonctionnalités malgré nos difficultés de compréhension de certains aspects du développement, notamment le parsing du texte.