

Delivery Template

***Asc2* Group Project**

***Development***

|  |
| --- |
| **Conditions d’utilisations :** SUPINFO International University vous permet de partager ce document. Vous êtes libre de :   * Partager — reproduire, distribuer et communiquer ce document * Remixer — modifier ce document   **A condition de respecter les règles suivantes :**  Indication obligatoire de la paternité — Vous devez obligatoirement préciser l’origine « SUPINFO » du document au début de celui-ci de la même manière qu’indiqué par SUPINFO International University – Notamment en laissant obligatoirement la première et la dernière page du document, mais pas d'une manière qui suggérerait que SUPINFO International University vous soutiennent ou approuvent votre utilisation du document, surtout si vous le modifiez. Dans ce dernier cas, il vous faudra obligatoirement supprimer le texte « SUPINFO Official Document » en tête de page et préciser notamment la page indiquant votre identité et les modifications principales apportées.  En dehors de ces dispositions, aucune autre modification de la première et de la dernière page du document n’est autorisée.  **NOTE IMPORTANTE :** Ce document est mis à disposition selon le contrat CC-BY-NC-SA Creative Commons disponible en ligne http://creativecommons.org/licenses ou par courrier postal à Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA modifié en ce sens que la première et la dernière page du document ne peuvent être supprimées en cas de reproduction, distribution, communication ou modification. Vous pouvez donc reproduire, remixer, arranger et adapter ce document à des fins non commerciales tant que vous respectez les règles de paternité et que les nouveaux documents sont protégés selon des termes identiques. Les autorisations au-delà du champ de cette licence peuvent être obtenues à support@supinfo.com.  © SUPINFO International University – EDUCINVEST - Rue Ducale, 29 - 1000 Brussels Belgium . www.supinfo.com |

Table of contents

1 Group summary 4

1.1 Group members 4

2 Project report 5

2.1 Company presentation 6

2.2 Task presentation 6

3 solution manual 7

4 technical documentation 7

# Group summary

## Group members

Campus: Strasbourg

Class: B2

|  |  |  |  |
| --- | --- | --- | --- |
| ID Open Campus | Last Name | First Name | Photo |
| 213969 | Erfani | Sébastien |  |
| 214666 | Biechy | Robin |  |
| 214411 | Baumeyer | Kévin |  |
| 213841 | Jaafar | Lisa |  |
| 213090 | Jolain | Yohann |  |

# Project report

## Company Presentation

Notre groupe de travail, issue de notre startup « **Deviant Squad** », est composé des membres suivants :

• Chef de projet : Robin BIECHY  
 Doté d’un excellent sens relationnel, toujours débordant d’énergie, il a su motiver ses troupes à fournir un travail de qualité. Investit dans le projet également, il s’est assuré que toutes les deadlines soient respectées, et que le rendu final corresponde au cahier des charges initial.

• Développeurs seniors : Sébastien ERFANI ; Lisa JAAFAR  
 Acteurs majeurs dans la construction du projet, nos développeurs seniors se sont occupés de la plus grande partie du code.

• Expert JAVA : Kevin BAUMEYER  
 Chargé de conseiller les développeurs, il s’est assuré de respecter les conventions JAVA. Il a également simplifié certaines parties du code.

• Développeur junior : Yohann JOLAIN  
 Aide principale des développeurs seniors, il s’est occupé des parties moins compliquées du code. Il était également chargé de faire les cafés.

## Task Repartition

# solution manual

Pour utiliser notre jeu, choisissez l’exécutable Java et lancez-le.

Entrez le nombre de commandes de votre choix. Les différentes commandes possibles sont :

|  |  |
| --- | --- |
| **Commandes** | **Effet** |
| AV *pixels* | La tortue bouge de *pixels* vers l’avant |
| RE *pixels* | La tortue bouge de *pixels* vers l’arrière |
| TD *degres* | La tortue tourne de *degres* vers la droite |
| TG *degres* | La tortue tourne de *degres* vers la gauche |
| FCC *couleur* | Change la couleur de la tortue en *couleur* au format RVB (exemple : #FF0000 pour le rouge) |
| LC | Le stylo est levé (rien ne se dessine) |
| BC | Le stylo est baissé (le dessin est active) |
| VE | Efface l'écran et place la tortue au centre, orientée vers le haut |
| CT | Cache la tortue |
| MT | Montre la tortue |
| REPETE *nbr\_de\_fois* [*commande*] | Execute la *commande* un certain *nbr\_de\_fois* |
| VI | Gère la vitesse (1 pour le plus rapide) |

L’application dessinera la tortue de votre choix en fonction des commandes rentrées.

# technical documentation

Reconnaissance de la grammaire:

Pour reconnaitre une commande, nous séparons les mots de la commande (séparés par des espaces) et nous les mettons dans une liste qui nous sert de stack. Si le mot suivant est un nombre nous l’ajoutons à la commande précédente. Cela nous permet d’écrire plusieurs commandes à la suite. Pour savoir si la commande est correcte, nous utilisons la réflexion en Java pour appeler la bonne méthode (qui est préfixée pour éviter les problèmes de sécurité) ; si la commande n’existe pas, Java jettera une exception, que nous attrapons et signalons à l’utilisateur. Ici, Java effectue le rôle d’analyseur syntaxique et lexical.

Gestion de la boucle:

Pour le cas du REPETE, nous comptons les crochets ouvrants et fermants, et quand ce compte devient égal (autant de crochets ouvrants que fermants), le REPETE est terminé. Nous remplaçons ensuite le REPETE, par le nombre de fois la commande demandée, et nous traitons ensuite les commandes normalement, nous les analysons grammaticalement de nouveaux, permettant de gérer les REPETE chainés.