

Delivery Template

***Class* Group Project**

***Development Project***

Delivery

Group members

Documentation

Version 1.8

Last update: 06/06/2017

Authors: POSTIC Corentin

SALADINI Raphaël

NOSJEAN Aymeric

ALLARD Romain

|  |
| --- |
| **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 4

3 solution manual 6

4 technical documentation 6

# Group summary

## Group members

Campus: Nice

Class: Asc.2

|  |  |  |  |
| --- | --- | --- | --- |
| ID Open Campus | Last Name | First Name | Photo |
| 216338 | POSTIC | Corentin |  |
| 216334 | SALADINI | Raphael |  |
| 213622 | NOSJEAN | Aymeric |  |
| 214551 | ALLARD | Romain |  |

# Project report

Afin d’optimiser au mieux la productivité de l’équipe, nous avons fait le choix d’instaurer une répartition des tâches, en prenant en compte les forces et les faiblesses de chacun.

Nous avons donc assigné la partie développement aux deux personnes les plus à même de réaliser celle-ci : Corentin POSTIC, et Aymeric NOSJEAN.

Concernant l’aspect design, Raphaël SALADINI a été choisi de par ses compétences dans ce domaine.

Enfin pour la partie documentation et présentation, « il s’agit de Romain Allard ».

# solution manual

Nous avons comme vous nous l’avez demandé de créer un nouveau langage de programmation qui se prénommé le logo. Nous voici au moment de vous présenter le langage. Le logo est une langage ludique, rapide à comprendre, et rapide d’exécution. Le logo peut être autant utilisé par des enfants que des personnes âgées. Le langage va vous permettre de faire des dessins facilement et rapidement.

Notre solution ne demande aucune installation, juste besoin d’un navigateur web ce qui préinstaller dans tous les ordinateurs.

Notre équipe vous assure une maintenance exceptionnelle pour qu’en cas de besoin on vous réponde le plus vite possible.

Les différentes possibles en LOGO

Pour programmer en LOGO, juste besoin de connaitre quelques fonctions.

Les fonctions sont intuitives et simples à reproduire.

Voici les commandes que vous pouvez exécuter en LOGO pour ainsi créer un dessin.

|  |  |
| --- | --- |
| AV *pixels* | La tortue avance du nombre de pixels demandé(s) |
| RE *pixels* | La tortue recule du nombre de pixels demandé(s) |
| TD *degrees* | La tortue tourne vers la droite du nombre de degré(s) demandé(s) |
| TG *degrees* | La tortue tourne vers la gauche du nombre de degré(s) demandé(s) |
| FCC color | Change la couleur du tracé de la tortue. La couleur devra être rentrée au format RGB |
| LC | Active le tracé de la tortue (*par défaut*) |
| BC | Désactive le tracé de la tortue |
| CT | Affiche la tortue (*par défaut*) |
| MT | Masque la tortue |
| REPETE *n [commandes]* | Permet de répéter n fois la commande |
| HELP | Ouvre la documentation |

Détails de l’IDE de LOGO

Nous avons choisi de créer un IDE le plus sobre et le plus simple possible pour le langage LOGO.

Notre IDE se présente en trois parties.

*IDE vide sans aucune commande rentrée.*

*IDE avec des commandes entrées.*

Nous pouvons donc voir trois parties différentes. Une (rond bleue) qui est le résultat de la commande exécutée. Une autre partie (rectangles orange) qui est un historique des commandes exécutés. Et enfin une dernière partie (rectangles jaunes) qui est l’éditeur de code ou vous pouvez entrer les commandes que vous voulez exécuter.

# technical documentation

Afin d’obtenir une adaptabilité maximum et un accès disponible sur la majorité des appareils, nous avons fait le choix de développer le jeu en HTML/CSS/JS.

La majorité du jeu est gérée en JavaScript :

* Le moteur de jeu
* La gestion des inputs et leur traitement
* La génération du code HTML

Quant au CSS, celui-ci nous permet d’améliorer l’user interface (police personnalisée, background, mise en forme).

Entrons dans le vif du sujet :

Lors du lancement de la page jeu.html la partie haute de l’écran, dédiée à l’affichage de la tortue,  est composée d’un canvas généré par le JavaScript.

C’est au centre de cette partie que la tortue apparaît avec la tête dirigée vers le haut.

La partie basse de l’écran qui est dédiée à l’entrée des commandes et à l’affichage de l’historique est un simple input en HTML.

Comme demandée dans le sujet, cet input affiche un point d’interrogation en début de ligne afin de signifier à l’utilisateur qu’il est en attente de l’entrée d’une commande.

Lorsque l’utilisateur écrit une commande et tape sur la toucher Entrée, l’event listener convertit tout le texte en majuscule, le divise en mots qu’il ajoute dans un tableau.

Si la longueur de celui-ci est égale à 4, il comprend qu’il s’agit d’une double commande, donne la valeur « true » au booléen « doubleCommand », vérifie que celles-ci sont présentes dans la liste, les exécute et les affiche dans l’historique.

Sinon il vérifie que la commande simple y soit présente, l’exécute et l’affiche dans l’historique.

Les commandes REPETE et FCC présentent une particularité :

La commande REPETE possède une Expression Régulière qui lui permet de récupérer le contenu entre crochets et la commande FCC vérifie que la couleur entrée par l’utilisateur respecte bien le format Hexadécimal à 3 ou 6 caractères.

Afin de faciliter l’accès à la documentation, nous avons fait le choix de rajouter une commande HELP qui ouvre le document PDF explicatif créé par nos soins.