

Test du programme et règles de qualité

Test de lancement en début de programme :

Est-ce que le programme fonctionne et le jeu se lance ?	Validé : <input checked="" type="checkbox"/>	Pas validé : <input type="checkbox"/>
---	---	--

Test de gameplay :

Ce qu'on teste	Description	Validé	Pas Validé
La voiture avance	La voiture avance bien sur la route par rapport au paysage	<input checked="" type="checkbox"/>	<input type="checkbox"/>
La voiture va à gauche et à droite	La voiture est capable de changer de voie en allant à gauche ou à droite pour esquiver les voitures	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Collision lorsque voiture touche	Lorsque la voiture touche un obstacle, le jeu s'arrête-t-il ?	<input checked="" type="checkbox"/>	<input type="checkbox"/>
La voiture ne sort pas du jeu (en allant tout à droite ou tout à gauche)	La voiture est capable d'aller sur les voies de gauche et de droite sur la route, mais elle ne doit pas sortir de la route	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Bien afficher les résultats	Le programme affiche bien le résultat et le temps réalisé	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Le jeu accélère au bout d'un certain temps	Le jeu accélère après plusieurs secondes afin de rajouter de la difficulté	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Les lignes de sécurité sont bien visibles sur le Micro:Bit	Les lignes de sécurité sont visibles afin de délimiter la route.	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Pas de BUGS lorsque les deux boutons sont appuyés en même temp sur le Micro-Bit	Lorsqu'on appuie sur les deux boutons en même temp sur le Micro-Bit, aucun BUG ne se produit	<input checked="" type="checkbox"/>	<input type="checkbox"/>
L'aléatoire du jeu marche bien	Le jeu fonctionne bien de façon aléatoire et n'est pas identique à chaque partie	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Test des règles de qualité :

Ce qu'on teste	Description	Validé	Pas Validé
<u>Lisibilité du code</u>	////////////////////	////	////
Nommage des variables	Noms explicites et pas trop longs. → Une variable locale commence par une minuscule. → Une variable globale commence par une majuscule.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Nommage des constantes	Noms explicites et pas trop longs. → Les constantes sont écrits tout en majuscules.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Nommage des fonctions	Noms explicites et pas trop longs.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<u>Présentation</u>	////////////////////	////	////
Indentations	Utilisation de la touche TAB pour présenter les instructions selon leur logique. → Permet de repérer facilement les débuts, fins et contenus des instructions.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Séparateurs	Séparer les différentes parties du code par des sauts de lignes (comme des paragraphes). → À l'intérieur des fonctions et entre les fonctions du fichier source.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Fonctions (MODULARITÉ du code)	Organiser le programme en fonctions. → Chaque fonction doit réaliser un seul traitement simple.	<input checked="" type="checkbox"/>	<input type="checkbox"/>

<u>Documentation</u>	////////////////////	////	////
En-tête de fichier	En haut du fichier, un commentaire précise le rôle du fichier, l'auteur et l'année de développement.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
En-tête de fonctions	En haut de chaque fonction, un commentaire explique ce que fait la fonction et le rôle des données échangées.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Commentaire de partie	Les différentes parties du code sont structurées et expliquées par des commentaires.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Commentaire de données	Des commentaires donnent des informations importantes sur les données utilisées.	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Bilan

L'ensemble des tests réalisés montrent que le programme fonctionne correctement et respecte les objectifs fixés. Le jeu se lance sans erreur, le gameplay est fluide et toutes les fonctionnalités principales (déplacements, collisions, affichage des résultats, accélération, aléatoire) sont validées sans bugs. De plus, les règles de qualité du code sont bien respectées : le code est lisible, bien structuré, correctement documenté et modulaire. Aucun dysfonctionnement majeur n'a été constaté, le jeu est donc conforme et opérationnel.

Rôles

Emirhan KARAHAN : RQ - Responsable Qualité

Tests fonctionnels, validation du code, contrôle qualité, tests du programme

Tomás DE ALMEIDA : RD - Chef de projet Développement

Direction du projet, coordination de l'équipe, développement principal.

Furkan AKTAS : RW - Responsable Communication WEB

Développement du site web, communication du projet, documentation en ligne

Hakan OZATA : RC - Responsable Conception/Documentation

Architecture du jeu, conception technique, rédaction de la documentation