**PROJET\_IALGEN\_ALLAL\_BERTRAND\_GOULOUMES**

1. **Note explicative sur les règles du jeu :**

Notre jeu est inspiré du jeu *Space Invaders*. Ce dernier est un jeu d’arcade en 2 dimensions. Nous avons gardé la mécanique de base où le joueur représenté par un vaisseau doit faire face à des vagues ennemies et peut leur tirer dessus en tapant sur la Barre d’espace et se déplacer latéralement/horizontalement (flèches de gauche et de droite) dans les limites de la Box (cadre blanc).

Nous avons ajouté des mécaniques différentes et un thème. En effet le jeu s’appelle **Space Rebellion**, faisant allusion à la série Star Wars (par le design de la page d’accueil, l’allure des vaisseaux ennemis : chasseurs de l’Empire).

Les variantes sont que les chasseurs apparaissent tous par le coin nord-ouest (ou top gauche) de la box mais de manière aléatoire. Et se déplacent horizontalement jusqu’à rencontrer un mur et descendre d’une ligne et de partir dans le sens inverse.

Le jeu se finit lorsqu’un chasseur de l’Empire (appelé *Invader* dans le code) est sur le point de se retrouver sur la même ligne que le vaisseau de la Résistance (appelé *Defendership* dans le code), le rendant vulnérable et annonçant la fin de la Résistance.

Vous pouvez augmenter la difficulté suivant le pilote sélectionné ce qui accélèrera la vitesse des chasseurs ennemis.

Aussi vous ne pouvez plus tirer de balles tant qu’il en reste une dans la Box ou que cette balle n’a pas touché un ennemi.

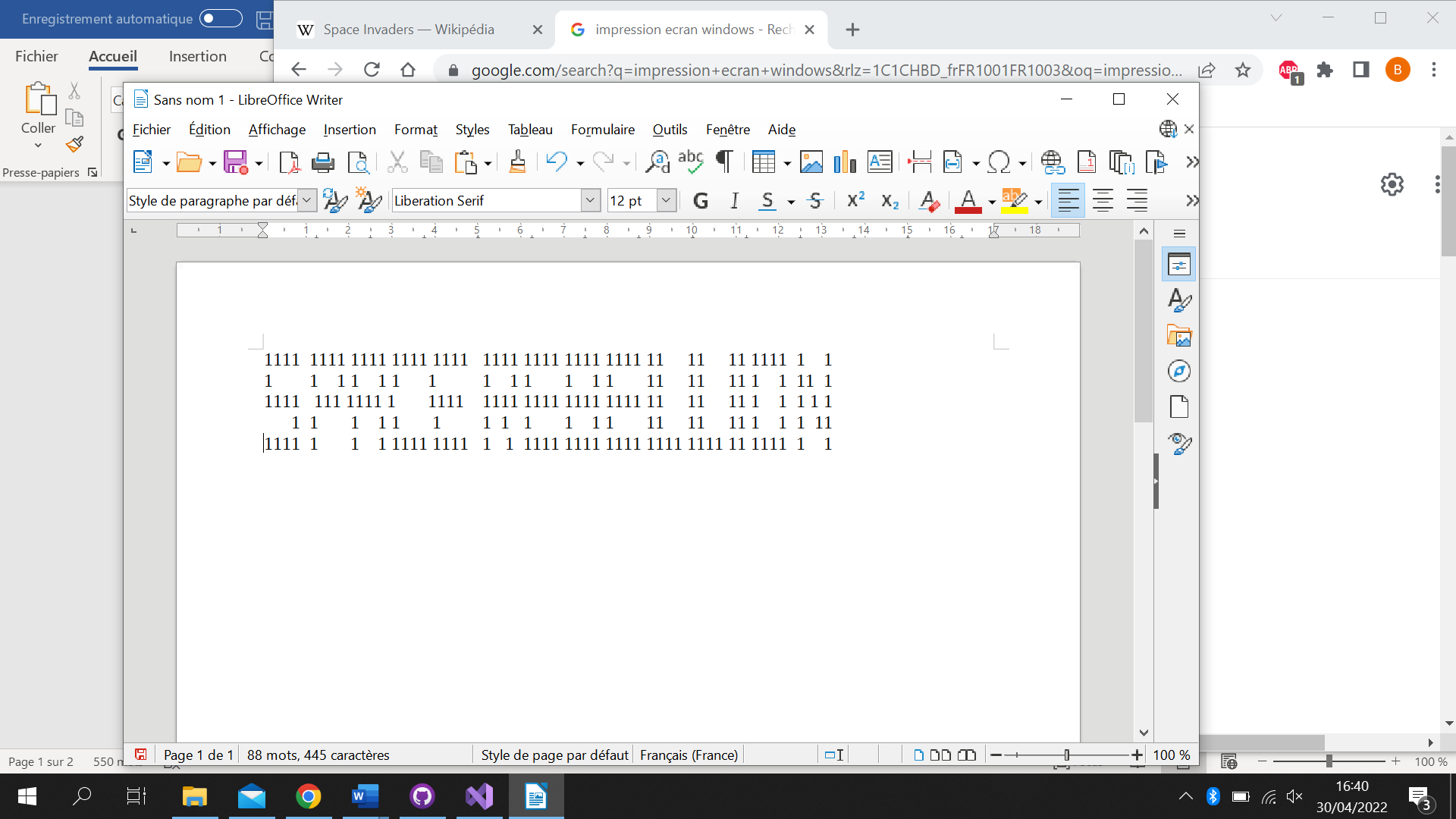
Il est évidemment possible de rejouer, votre score (nombre de vaisseaux abattus et temps de survie) sera affiché.

1. **Note explicative sur le fonctionnement du code :**

Notre code est notamment composé de 4 objets pour gérer les parties :

* **Defendership** (vaisseau controlé par le joueur)
* **Invader** (vaisseau ennemi)
* **Box** (cadre de jeu et aussi de la page d’accueil)
* **Bullet** (balle envoyée par le Defendership)

La classe **HomePage** gère toute ce qui se passe avant et après une manche de jeu.

Les 2 méthodes importantes sont Launch et Draw. Cette dernière fait appel à Random\_Invaders et Random\_Bullets pour afficher des vaisseaux et des tirs dans la box de la page d’accueil. On se sert au fur et à mesure de working\_on\_row pour déterminer la ligne sur laquelle on affiche. Puis Game\_Title permet afficher le titre. En effet, on ne peut augmenter la taille des caractères dans la console donc l’alternative à été de dessiner le texte sur un fichier word avec des 1 comme on peut le voir sur l’impression écran suivante :

Il suffit ensuite d’automatiquement remplacer les vides par des 0, et Game\_Title peut écrire le titre dans la console. Enfin grâce à 2 booléens (restart et phase\_1) le code sait si on a déjà fait une partie et si déjà eu le premier message (demandant de jouer ou rejouer : Phase\_1 ou EndGame\_Announcement).

Launch gère le déroulé des appels de Homepage.Draw() et récupère les deux réponses du joueur et on les retourne. Il gère aussi si le joueur ne veut plus rejouer.

La classe **Game** gère la partie en question.

Et **Program** correspond au déroulé du jeu, c’est le Main de notre code. Il fait donc appel à Homepage pour gérer l’affichage d’avant partie et récupérer les requêtes du joueur (s’il veut jouer et si oui avec quelle difficulté).

Les 2 boucles *while* imbriquées sont très importantes à comprendre. On reste dans la première tant que le joueur n’a pas dit qu’il ne voulait plus jouer (il peut le dire avant et après une manche en ne cliquant pas sur Y mais sur une autre touche). On est dans la seconde tant que les vaisseaux ennemis n’ont pas gagné.

Dans cette seconde boucle, on récupère la touche sur laquelle le joueur a cliqué (si il clique sur une touche), on met à jour la position des éléments mobiles (Bullet, Defendership et Invaders), on les réaffiche tous (plus la Box).