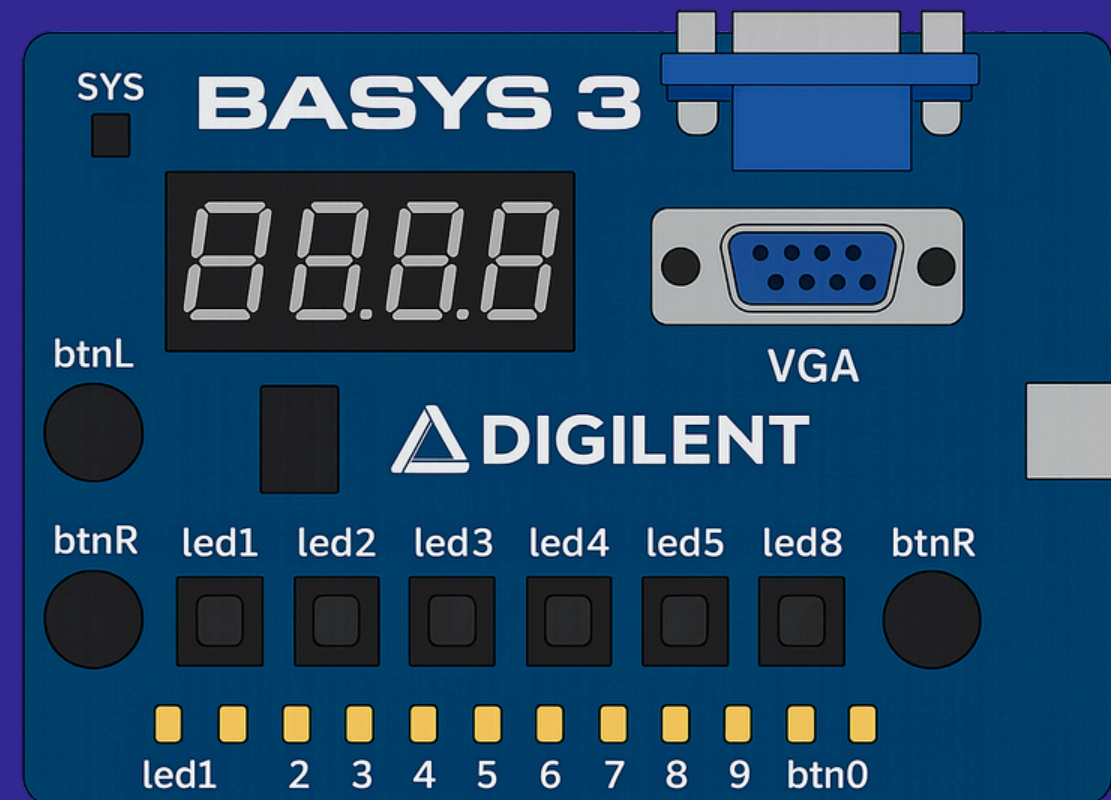


**Clément Veith, Valentin Scias,  
Lucas Ribeiro, Jean-Baptiste Mattei,  
Paul François**

**(projet organisé par le  
meilleur professeur de  
l'ISEN)**

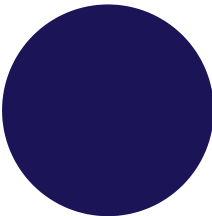
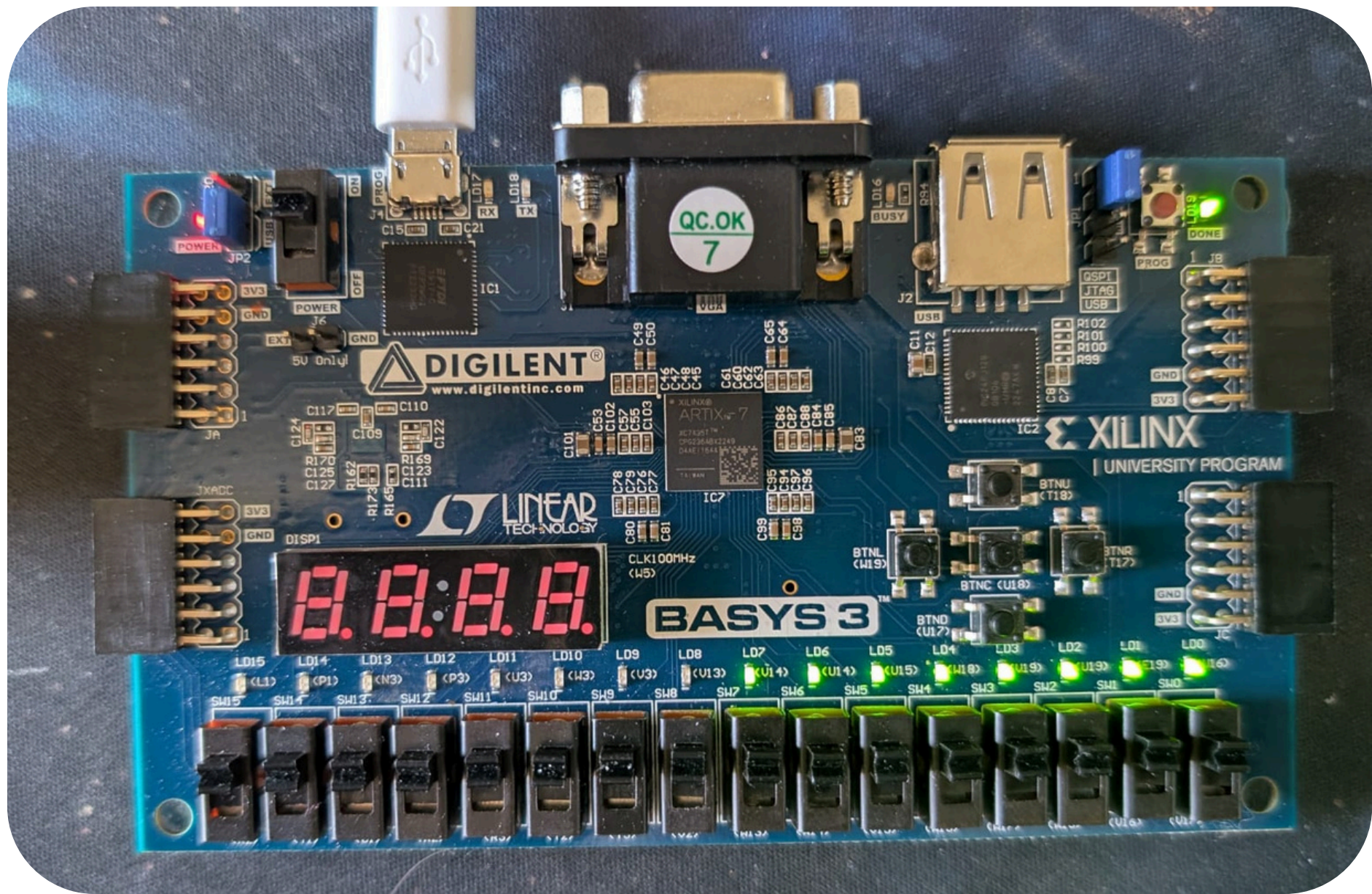


# **VHDL**

**Une présentation ISEN**



**LEDS :**  
**Clignotement, Chenillard, Compteur et décompteur**



# POINTS CLÉS :

- Incrémentation du nombre de LEDs allumées vers la droite (btnL)
- Décrémentation avec le bouton de droite (btnR)
- Affichage direct du nombre d'appuis sur les 16 LEDs de la carte
- Limitation stricte entre 0 et 16 LEDs allumées (aucun dépassement possible)
- Mise en place d'un système d'anti-rebond pour éviter les doubles appuis
- Utilisation d'un diviseur d'horloge pour ralentir la détection des appuis

# DIFFICULTÉS :

✗ Plusieurs LEDs s'allumaient ou s'éteignaient d'un coup →

Résolu par l'ajout d'un système anti-rebond simple avec une temporisation

✗ Les boutons n'étaient pas reconnus →

Résolu en activant leurs broches dans le fichier de contraintes (.xdc)







# POINTS CLÉS :

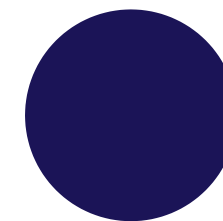
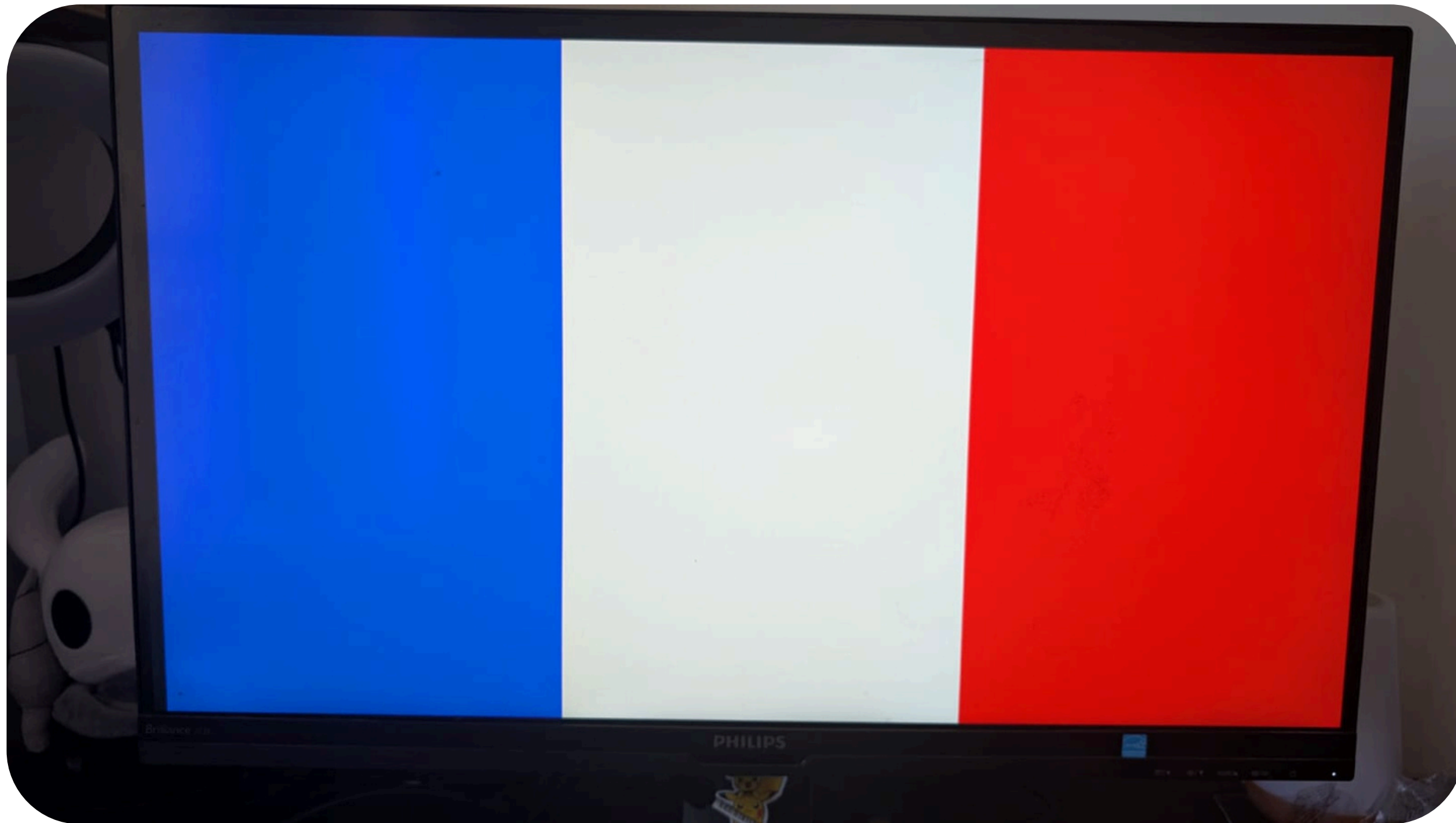
- Conception du fichier `chrono.vhd` en architecture structurée :
- Utilisation de 4 compteurs modulo 10 en cascade pour compter les unités
- Multiplexage de l'affichage à l'aide :
- d'un compteur de rafraîchissement
- d'un multiplexeur 4 vers 1 (`mux4.vhd`) pour sélectionner le chiffre à afficher.
- Activation du bon afficheur via `decodeur_an.vhd`.

# DIFFICULTÉS :

- ✗ Segments mal câblés (affichage illisible) → Correction dans le fichier `.xdc` :  
Mauvais mapping initial de certaines broches (erreurs sur V6, U7, U8...)  
Remplacement progressif de broches jusqu'à obtention d'un affichage correct.
- ✗ Point décimal allumé alors qu'il ne devrait pas → Diagnostic : 8e bit actif à tort → solution : forcer `seg7` à 1
- ✗ Certains chiffres (ex. 9 ou 5) mal affichés → Détection via tests manuels → recalibrage de la table de vérité

**VGA :**

**Image fixe monochrome, Image fixe 3 couleurs différents**





# POINTS CLÉS :

- Reprise du `vga_controller_640_60.vhd` pour gérer l'affichage VGA.
- Création du module `image.vhd` :
- Utilisation des signaux `hc`, `vc` et `blank` pour dessiner les pixels.
- Découpage horizontal de l'écran en 3 bandes égales :
  - Bande gauche : bleue
  - Bande centrale : blanche
  - Bande droite : rouge
- Utilisation des sorties `RED`, `GREEN`, `BLUE` codées sur 4 bits pour les couleurs.

# DIFFICULTÉS :

✗ Drapeau décalé ou mal coloré →

Résolu par calcul précis des limites de `hc` pour chaque bande (0–213, 214–426, 427–639).

✗ Tout l'écran reste noir →

Solution : inversion correcte du signal `blank` (ne pas afficher de couleur si `blank` = '1').

✗ Vitesse d'horloge incorrecte →

Résolu par ajout d'un diviseur d'horloge pour générer le signal `pixel_clk` à 25 MHz.

# PONG :





# POINTS CLÉS :

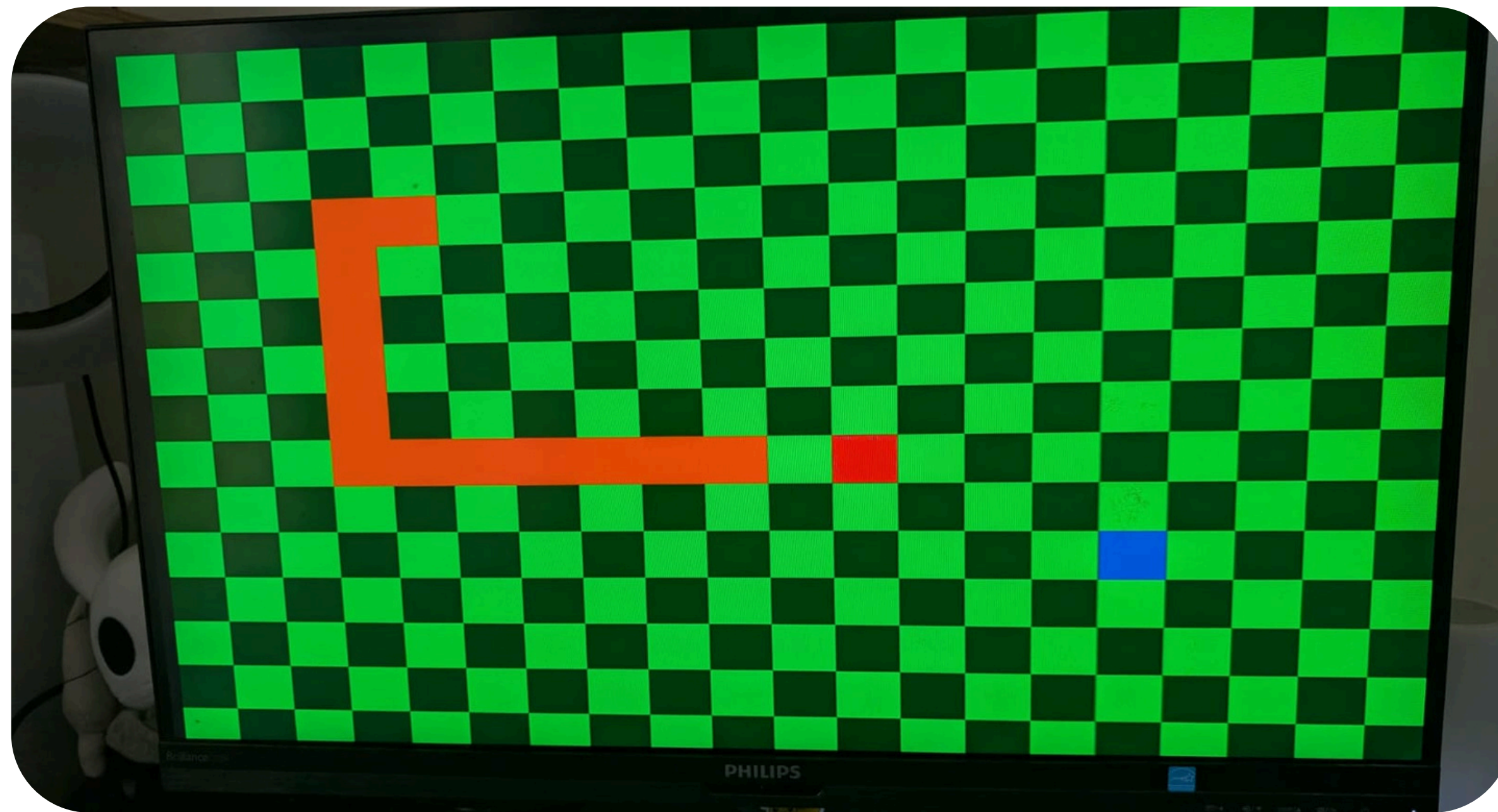
- Affichage de deux raquettes (verte à gauche, bleue à droite).
- Génération et affichage de la balle en mouvement.
- Affichage d'une ligne centrale clignotante (repère visuel).
- Gestion du déplacement des raquettes via les boutons (btnU/btnD et btnL/btnR).
- Implémentation du mouvement automatique de la balle.
- Détection des collisions avec les raquettes et inversion de direction (ball\_dx).
- Ajout de la gestion des rebonds sur les bords haut et bas (ball\_dy).
- Réinitialisation de la balle au centre en cas de sortie par la gauche ou la droite.

# DIFFICULTÉS :

- ✗ Raquettes invisibles → Résolu par correction des coordonnées de dessin.
- ✗ Balle qui sort de l'écran sans revenir → Réinitialisation automatique avec inversion de direction.
- ✗ Pas de rebond sur les bords haut/bas → Implémentation du test sur ball\_y et inversion de ball\_dy.
- ✗ Vitesse de balle trop élevée → Ajout d'un tick pour ralentir les mises à jour.



# SNAKE :



# POINTS CLÉS :

- Grille logique obtenue en divisant l'écran en cellules carrées
- Placement dynamique du serpent (tête + corps) et de la nourriture
- Déplacement contrôlé par boutons directionnels
- Système de croissance du corps à chaque nourriture mangée
- Gestion des collisions (murs, auto-collision)
- Ajout d'un fond damier vert clair / vert foncé pour la lisibilité

# DIFFICULTÉS :

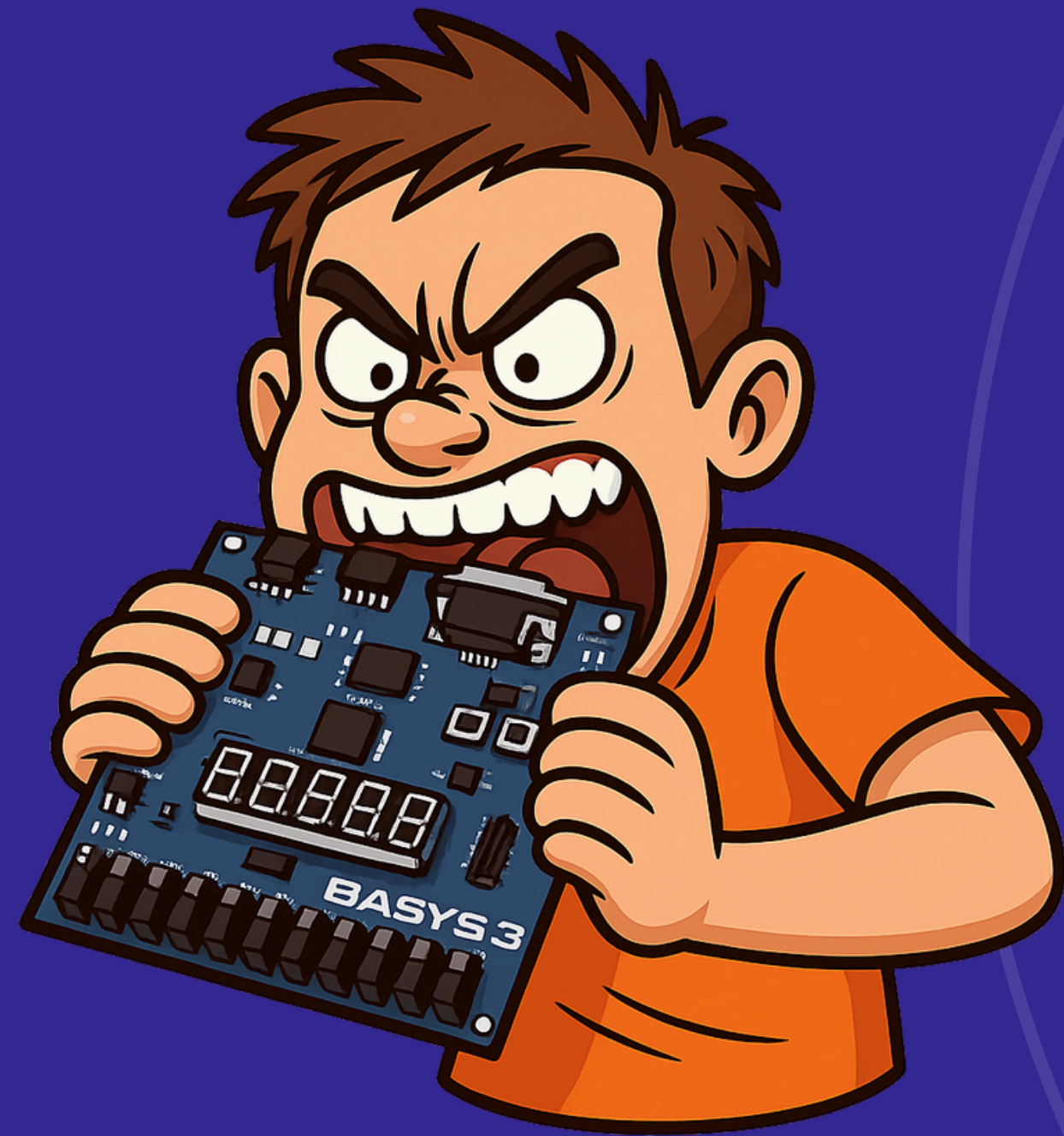
✗ Grille trop grande → Résolu en augmentant la taille des cellules (ex. 32 px → 20\*15)

✗ Espace visuel entre tête et corps → Résolu en mettant à jour la tête avant le corps, puis en stockant l'ancienne position

✗ Déplacement trop rapide / bruité → Résolu par ajout d'un système de tempo simple (anti-rebond)

✗ Valeurs hors limites → Résolu en fixant les bornes de position de manière explicite





**MERCI**

**( AVERAGE VHDL MOMENT )**