

Console de jeux sur FPGA

CHANG Kirsten, LE Hoang Hieu

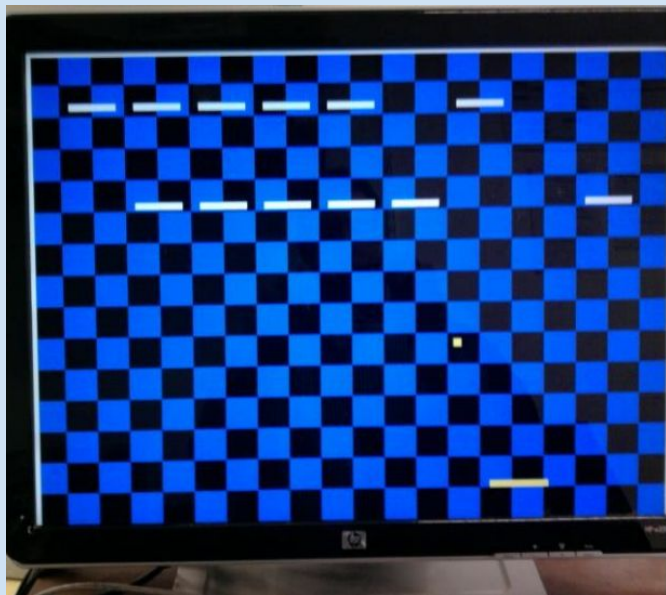
Plan

- ❖ Introduction / Contexte du projet
- ❖ Architecture globale de la console
- ❖ Détail et résultat de la tâche 2
- ❖ Détail et résultat d'une amélioration que nous avons créée
- ❖ Bilan d'avancement
- ❖ Conclusion

Introduction / Contexte du projet

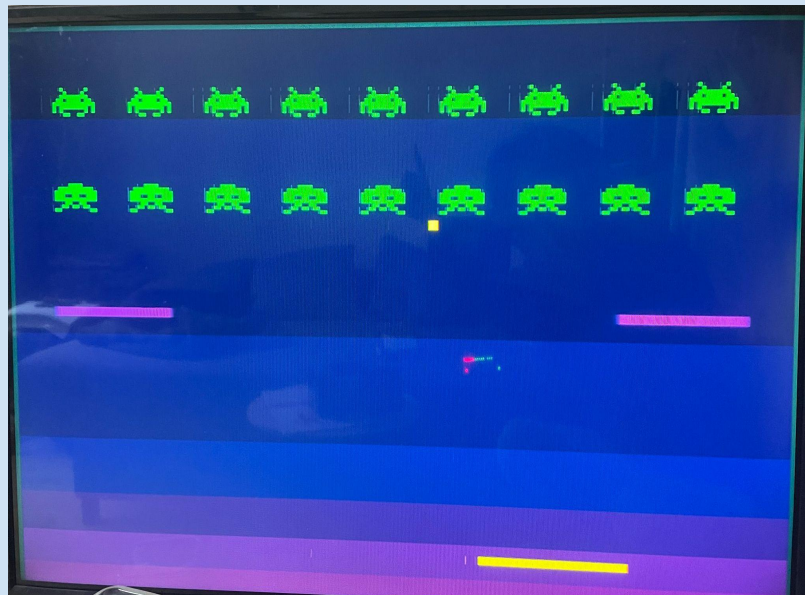
Avant :(

Classique, banal, le jeu ne marche pas

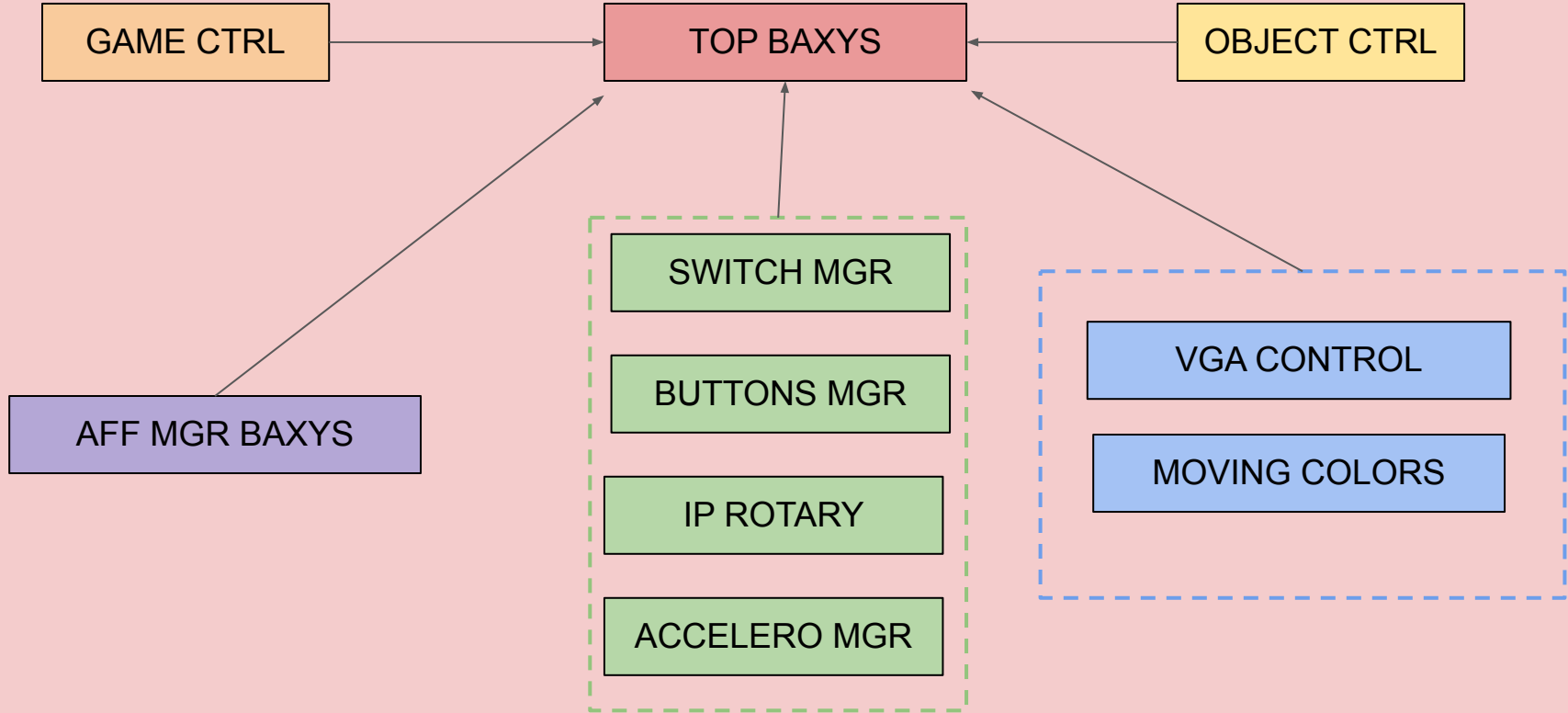


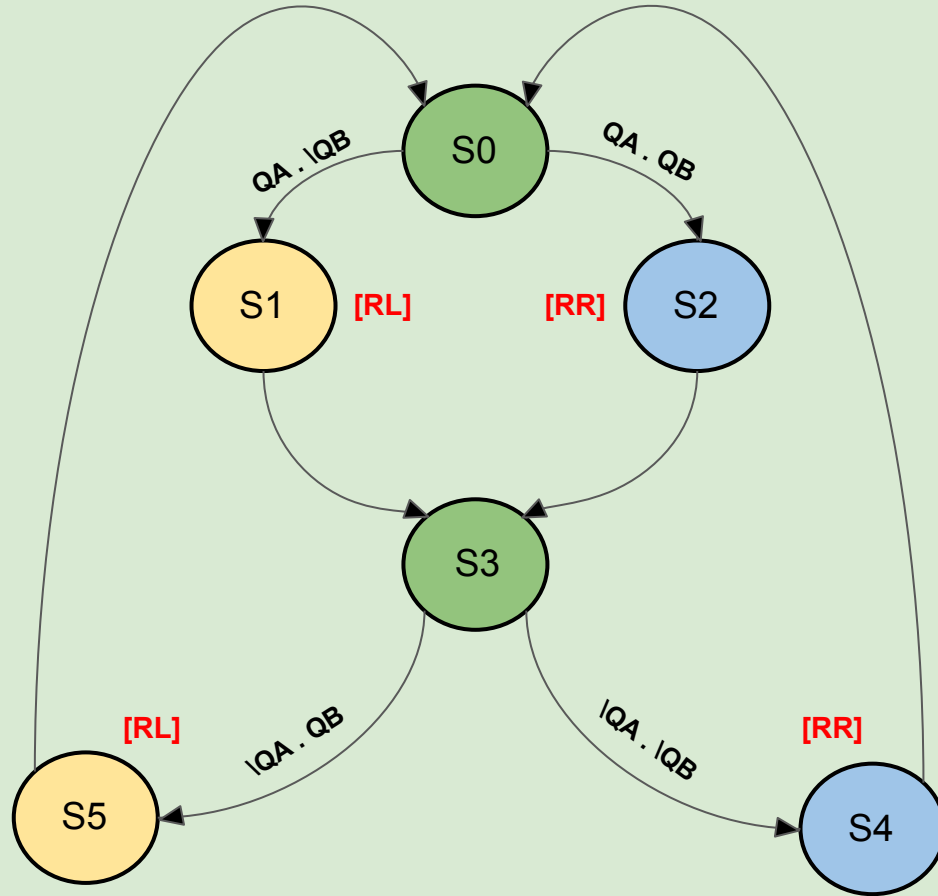
Après :)

WOW! Esthétique, du challenge, le jeu marche!



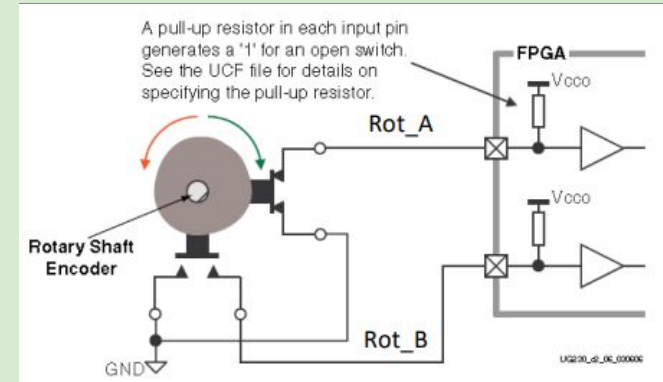
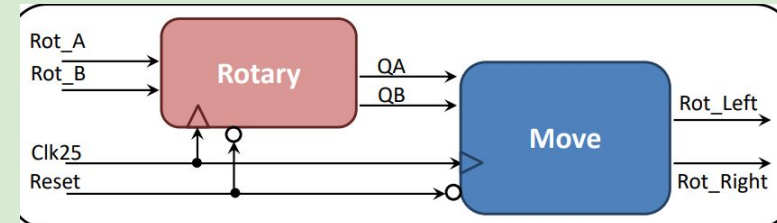
Architecture globale du projet





TÂCHE 2 : Encodeur rotatif

- Dépend de QA :
 - QA → QB : Left
 - QB → QA : Right



P1 : codage des transitions entre les états

```
process(Clk25 , reset)
begin

if reset = '0' then state <= "000";

elsif rising_edge(Clk25) then
    --EL1 = "001"
    if (state = "000") and ( (QA='1') and (QB='0') ) then
        state <= "001" ;

    elsif (state = "000") and ( (QA='1') and (QB='1') ) then
        state <= "010" ;

    elsif (state = "001") or (state = "010") then
        state <= "011" ;

    elsif (state = "011") and ( (QA='0') and (QB='1') ) then
        state <= "100" ;

    elsif (state = "011") and ( (QA='0') and (QB='0') ) then
        state <= "101" ;

    elsif (state = "100") or (state = "101") then
        state <= "000" ;

    else state <= state;

    end if;
end if;

end process;
```

P2 : codage des sorties pour chaque état

```
36 |
37 | process(state)
38 | begin
39 |     case(state) is
40 |         when "001" =>
41 |             RL <= '1';
42 |             RR <= '0';
43 |
44 |         when "010" =>
45 |             RL <= '0';
46 |             RR <= '1';
47 |
48 |         when "100" =>
49 |             RL <= '1';
50 |             RR <= '0';
51 |
52 |         when "101" =>
53 |             RL <= '0';
54 |             RR <= '1';
55 |
56 |         when others =>
57 |             RL <= '0';
58 |             RR <= '0';
59 |     end case;
60 | end process;
61 |
62 | end archi;
63 |
64 |
```


Amélioration : Space Invaders!



A	B	C	D	E	F	G	H	I	J	K	L	M
[51 - 54 [[41 - 44 [[44 - 47 [[47 - 50 [[50 - 53 [[53 - 56 [[56 - 59 [[59 - 62 [[62 - 65 [[65 - 68 [[68 - 71 [[71 - 74 [[74 - 77 [
[54 - 57 [
[57 - 60 [
[60 - 63 [
[63 - 66 [
[66 - 69 [
[69 - 72 [
[72 - 75 [

```

-- Pour Chaque Brique
for i in 0 to 1 loop
  for j in 0 to 8 loop
    if (
      i = 0 and
      ((ypos > 54 + i*100) and (ypos < 54 + i*100) and (xpos >= 47 + j*64) and (xpos < 50 + j*64))
      or
      ((ypos > 50 + i*100) and (ypos < 54 + i*100) and (xpos >= 68 + j*64) and (xpos < 71 + j*64))
      or
      ((ypos >= 54 + i*100) and (ypos < 57 + i*100) and (xpos >= 50 + j*64) and (xpos < 53 + j*64))
      or
      ((ypos >= 57 + i*100) and (ypos < 60 + i*100) and (xpos >= 47 + j*64) and (xpos < 71 + j*64))
      or
      ((ypos >= 60 + i*100) and (ypos < 63 + i*100) and (xpos >= 44 + j*64) and (xpos < 50 + j*64))
      or
      ((ypos >= 60 + i*100) and (ypos < 63 + i*100) and (xpos >= 53 + j*64) and (xpos < 65 + j*64))
      or
      ((ypos >= 60 + i*100) and (ypos < 63 + i*100) and (xpos >= 68 + j*64) and (xpos < 74 + j*64))
      or
      ((ypos >= 63 + i*100) and (ypos < 66 + i*100) and (xpos >= 41 + j*64) and (xpos < 77 + j*64))
      or
      ((ypos >= 66 + i*100) and (ypos < 69 + i*100) and (xpos >= 41 + j*64) and (xpos < 44 + j*64))
      or
      ((ypos >= 66 + i*100) and (ypos < 69 + i*100) and (xpos >= 47 + j*64) and (xpos < 71 + j*64))
      or
      ((ypos >= 66 + i*100) and (ypos < 69 + i*100) and (xpos >= 74 + j*64) and (xpos < 77 + j*64))
      or
      ((ypos >= 69 + i*100) and (ypos < 72 + i*100) and (xpos >= 41 + j*64) and (xpos < 44 + j*64))
    )
  end if
end for
end for

```


Bilan d'avancement :

4 tâches principales :

- Tâche 1 : Contrôleur VGA ✓
- Tâche 2 : Gestion de la raquette ✓
- Tâche 3 : Gestion des jeux ✓
- Tâche 4 : Obstacle mobile ✓

4 améliorations :

- Ajout d'un timer pour le jeu Casse-briques 🕒
- Changement des briques en aliens 🛸
- Ajout de 2 obstacles mobiles 📶
- Modification des couleurs du fond, obstacle, et mur ✨

