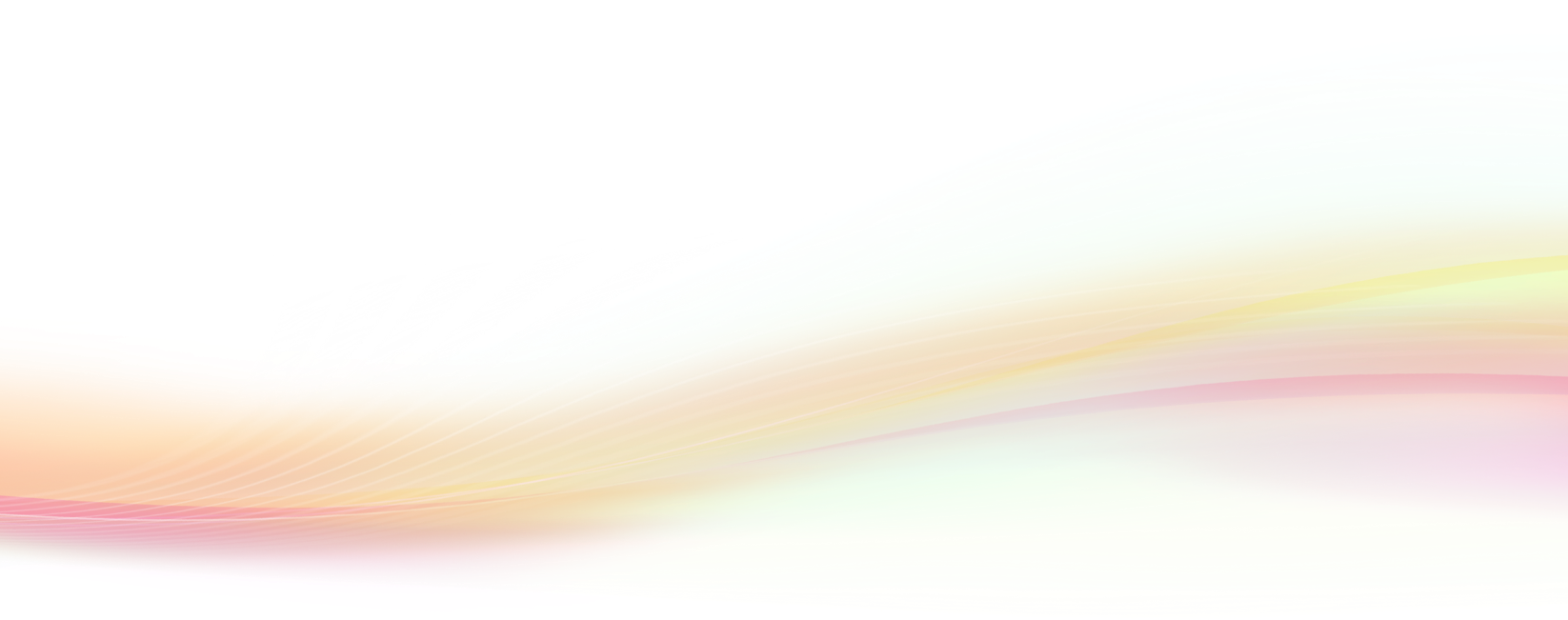
Projet Architecture B1

Rapport

ROUGER Simon



Introduction:

Pour ce projet de fin d'année, il fut demandé de produire un jeu de type casse-brique nommé "Break-Warrior" en Langage Assembleur pour les processeurs du type 8086.

Dans rapport, nous allons commencer en montrant les réponses effectuées au cahier des charges et ensuite parler d'une répartition de travail.

I. Réponse au cahier des charges.

Le travail demandé ne pu malheureusement pas être terminé. Les raisons de ce travail inachevé sont multiples: groupe divisé, gestion du temps imparfaite, répartition du travail manquante, ect... Ainsi, les travaux effectués dont les algorithmes ont été préparés sont les jalons 1, 2, 3, 4, 6 (en partie), 7.

a) Jalon 1

Dans ce jalon, il était simplement demandé d'effectuer un tracé droit oblique partant de l'origine X= 0, Y=0. Mais dans ce premier jalon, on devait choisir quelle méthode utiliser pour l'affichage (utiliser les interruptions systèmes, utiliser la mémoire vidéo, ou utiliser le buffer de la mémoire). Au début, les interruptions ont été choisies du fait de la facilité d'utilisation avec la position du curseur. Cependant, cette méthode ce révèle très lente, consommatrice de charge processeur, et pas optimisée pour un jeu demandant de la vitesse. Donc, l'utilisation directe de la mémoire vidéo fut retenue du fait qu'elle soit plus optimisée. Voici l'algorithme de base utilisé:

FONCTION debut

(emplacement X, Y;

ecrire " ";)

Fonction deplacement

FAIRE

{emplacement X++, Y++;

ecrire " ";}

TANT QUE

(1 =< X =< 80;

1=< Y =< 25;)

b) Jalon 2

Ici, il fallait également effectuer un tracé modélisant le déplacement de la balle, mais il fallait aussi effectuer le rebond de la balle sur les rebords du cade de l'écran. Voici l'algorithme correspondant:

SI ( X== 80 ET emplacement X++, Y++)

ALORS { emplacement X--, Y++;

ecrire " ";}

SI ( X== 1 ET emplacement X--, Y++)

ALORS { emplacement X++, Y--;

ecrire " ";}

SI (X == 80 ET emplacement X++, Y--)

ALORS { emplacement X--, Y--;

ecrire " ";}

SI ( X== 1 ET emplacement X--, Y--)

ALORS { emplacement X++, Y++;

ecrire " ";}

SI ( Y== 25 ET emplacement X++, Y++)

ALORS { emplacement X++, Y--;

ecrire " ";}

SI ( Y== 25 ET emplacement X--, Y++)

ALORS { emplacement X++, Y--;

ecrire " ";}

SI ( Y== 1 ET emplacement X++, Y--)

ALORS { emplacement X++, Y++;

ecrire " ";}

SI ( Y== 1 ET emplacement X--, Y--)

ALORS { emplacement X--, Y++;

ecrire " ";}

c) Jalon 3

Dans cette partie, il fallait reprendre les travaux des jalons précédents et y effacer les positions anciennes de la balle. Voici l'algorithme:

BOUCLE { mettre registre 1, position;

afficher position

effacer position }

d) Jalon 4

Là, il fallait seulement afficher progressivement un obstacle central sous forme de croix qui part du centre jusqu'aux extrémités. AInsi l'algorithme est:

Position centre ( X== 40, Y==12)

{ecrire "+";}

FAIRE {position X==40;

position Y ==12;

X++;

ecrire "-";

sauvegarde position1 X;}

TANT QUE (X <50)

FAIRE {position X==40;

position Y ==12;

X--;

ecrire "-";

sauvegarde position2 X;}

TANT QUE (X >30)

FAIRE { position Y==12;

position X==40;

Y++;

ecrire "­l";

sauvegarde position3 Y;}

TANT QUE (Y<22)

FAIRE { position Y==12;

position X==40;

Y--;

ecrire "­l";

sauvegarde position3 Y;}

TANT QUE (Y>2)

e) Jalon 5

Ce jalon fut une grande difficulté pour le développer en Assembleur. Il n'a pu être fait malheureusement. Il aurait certainement été judicieux d'utiliser un tableau à 2 dimensions pour s'y référencer à propos des obstacles et de pouvoir les supprimer du tableau des qu'ils sont supprimés par la balle. Voici tout de même l'algorithme produit:

TANT QUE (X=="-" ET emplacement X++, Y++)

FAIRE {emplacement X--, Y++;

ecrire " "}

TANT QUE (X=="-" ET emplacement X--, Y++)

FAIRE {emplacement X++, Y++;

ecrire " "}

TANT QUE (X=="-" ET emplacement X++, Y--)

FAIRE {emplacement X--, Y--;

ecrire " "}

TANT QUE (X=="-" ET emplacement X--, Y--)

FAIRE { emplacement X++, Y--;

ecrire " ";}

TANT QUE (X=="l" ET emplacement X++, Y++)

FAIRE { emplacement X++, Y--;

ecrire " ";}

TANT QUE (X=="l" ET emplacement X++, Y--)

FAIRE { emplacement X++, Y++;

ecrire " ";}

TANT QUE (X=="l" ET emplacement X--, Y++)

FAIRE { emplacement X--, Y--;

ecrire " ";}

TANT QUE (X=="l" ET emplacement X--, Y--)

FAIRE { emplacement X--, Y++;

ecrire " ";}

f) Jalon 6

Dans ce jalon, il était demandé de rajouter les procédures cadre, raquette et score. Malheureusement, la procédure score n'a pas pue être afficher.

g) Jalon 7

Ici, il fallait simplement afficher un message si on avait perdu et un autre si on gagnait la partie. Ainsi, si Y= 25, on affiche le message perdu, si le score correspond à une certaine somme correspondant à tous les obstacles supprimés on affiche le message gagné.

SI (Y==25)

ALORS {afficher message perdu;}

SI (Score ==41)

ALORS {afficher message gagné}

h) Jalon 8

Pour ceci, il fallait introduire la prise en compte de l'entrée clavier. Les interruptions étaient possibles sauf que peut être lentes comme pour la sortie écran. Voici tout de même une partie en algorithmique:

TANT QUE (interruption clavier W)

FAIRE { position raquette X-- ;}

TANT QUE (interruption clavier X)

FAIRE { position raquette X++; }

TANT QUE (interruption clavier Q)

FAIRE { Quitter programme;}

i) Option réseau

Optionnellement, il fallait réfléchir à une implémentation pour du jeu en réseau. Il serai possible de monter un serveur pour récupérer un "high score" des joueurs et afficher ainsi les meilleurs scores au jeu. Ainsi, aux différentes couches OSI, l'Application serai utilisée par un protocole http, la présentation par un fichier .txt, session par une session unique, le transport par des paquets TCP pour assurer l'envoi correcte du fichier, le réseau par de l'encapsulation paquet IP, la liaison de données par de l'encapsulation de trames MAC, et pour finir la liaison physique par des données binaires.

II. Répartition du travail.

La répartition du travail fut malheureusement très difficile du fait que ça soit notre premier projet de ce genre effectué par nous même et surtout le fait que nous soyons divisés à cause des histoires sur notre campus (campus de Bordeaux). Ainsi, nous étions à deux endroits différents pour faire les cours et avions beaucoup de mal pour nous retrouver et nos semaines de projet n'étaient pas en même temps. Donc, pour ce projet, une seule personne fut dessus (Simon ROUGER), ce qui a peut être engendré le fait que le projet soit rendu inachevé.

Conclusion: Ce premier projet fut d'une difficulté plutôt élevé du fait qu'il soit effectué par une seule personne n'ayant pas beaucoup d'aide et aussi d'une gestion du temps difficile dans le contexte des histoires au sein même du campus de Bordeaux. Malgré tout, ce projet montre bien que le travail de groupe est très important pour ce type de projet et permet une amélioration significative des membres du projet dans une matière donnée (l'assembleur, l'architecture des ordinateurs). Ce projet fut également très instructif et montre bien que le choix de bonnes personnes et primordial pour effectuer à bien ce projet ainsi qu'une bonne gestion du temps.