

Projet Space Invaders

G2 - G4

BENNACER Chakib

LESBROS Raphaël

MATHIEU--DRIF Clément

PONSARD Nils

TRITZ Hugo

Ajouts graphiques

Utilisation de MinGL :

- création de figures pour les ennemis, le joueur ...
- affichage via vecteurs de positions (un par type d'entité)
- ajout d'un HUD (Head up display)



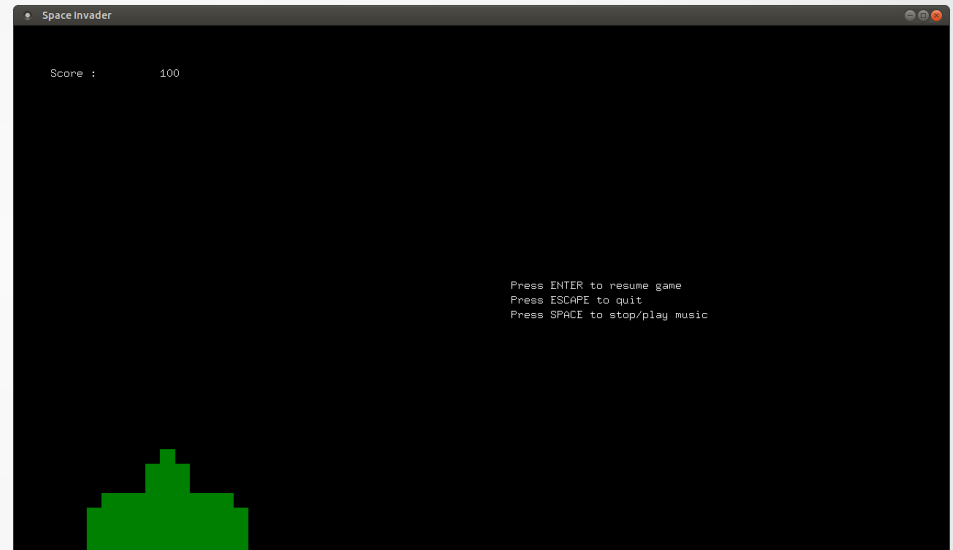
Jeu : vague 1

Ajouts sur le gameplay

- Jeu en temps réel
 - avec une gestion de la fréquence d'images en calculant le temps à attendre pour rester à 30 FPS
- Le jeu devient un jeu d'arcade
 - Ajout du score et d'un classement
 - Augmentation de la difficulté à chaque vague
 - Ajout de points de vies pour le joueur
- Complexification du jeu
 - Bonus (score, vie, vitesse de tir)
 - Accélération de la vitesse des ennemis
 - ennemis bonus régulièrement

Ajouts divers

- Musique et bruitages
- Écran d'accueil
- Pause
- Navigation dans les menus
- Fichier de configuration pour modifier le jeu
- Sauvegarde des meilleurs scores



Pause

Ajouts de fonctionnalités dans minGL

- Ajout d'une fonction pour afficher du texte à l'écran
 - Fonction `displayText(x, y, str)`
 - Utilise la fonction `glutBitmapCharacter()`
 - d'après http://mperriss.free.fr/opengl/Guide_2D/texte.htm
- Modification de la gestion des touches
 - Utilisation d'une map pour connaître les touches pressées
 - Ajout d'un *handler/callback* pour `keyboardUp` et `keyboardSpecialUp`
 - Fonctions `isPressed()` et `resetKey()`
 - Type `keyType`
- Ajout d'une fonction pour changer la couleur de fond
 - Variable privée contenant la couleur actuelle
 - Modifiée avec la fonction `setBgColor(const RGBcolor &col)` ou dans le constructeur

Fonctionnement général

struct spaceInvaders :

- entity, bonusEntity
- Positions
- Vitesses
- délais

Passés en paramètre à
la fonction de gestion
des collisions

Struct entity :

- entityHeight;
- entityWidth;
- figure / vector<figure>

```
pos playerPos;  
vector<pair<pos, unsigned>> invadersPos;  
vector<pos> playerTorpedoPos;  
vector<pair<pos, short>> UpgradePos;
```

Fréquence d'images

```
const chrono::duration<double, milli> frameDuration(33.3); //30fps
```

30 images par seconde



Menu

33.3562
33.3639
33.3677
33.3661
33.3608
33.3741
33.3675
33.3646
33.555
33.363
33.3671
33.3719
33.3759
33.3653
33.3676
33.3684
33.3606
33.3665
33.3587
33.3601
33.4075

En ms

Audio

startMusic()



```
#!/bin/bash
```

```
while [ 1 ];  
    do aplay -q '../ressources/theme.wav'  
done
```

killMusic() : pkill → script + processus aplay correspondant

Gestion des touches

Les pressions de touche sont des callbacks (bind_callback) vers des fonctions

On a 4 fonctions :

GlutKeyboardFunc()

GlutKeyboardUpFunc()

GlutSpecialFunc()

GlutSpecialUpFunc()



Modifie en
fonction

Variable map keyMap associant chaque touche à son état (pressé/relevé)

- Accessible via la fonction isPressed()
- Quand on utilise un bouton dans un menu on utilise resetKey() pour éviter d'interagir avec le menu suivant

keyType - keyMap

```
typedef pair<int, bool> keyType;
```

```
typedef map<keyType, bool> keyMap;
```

```
//http://mperriss.free.fr/opengl/Guide\_2D/claviersouris.htm
```

```
const keyType KEY_UP({101, true});  
const keyType KEY_DOWN({103, true});  
const keyType KEY_LEFT({100, true});  
const keyType KEY_RIGHT({102, true});  
const keyType KEY_ESCAPE({27, false});  
const keyType KEY_SPACE({32, false});  
const keyType KEY_ENTER({13, false});  
const keyType KEY_RETURN({8, false});
```

Lecture du fichier de configuration

```
struct AuthorizedKey
```

```
void createConf(const string & fileName, map<string, unsigned> Map)
```

Créer un fichier de configuration contenant les valeurs par défaut si il n'existe pas

```
map<string, string> loadConfig(const string & fileName)
```

Les valeurs absentes de ce fichier sont remplacées par les valeurs par défaut

Chargement du fichier de score

```
vector<pair<string, unsigned>> loadScores(const string & fileName)
```

Charge au maximum 3 scores (a partir du début du fichier)

```
int saveScores(const vector<pair<string, unsigned>> & scores,  
const string & fileName);
```

```
void scoreSort(vector<pair<string, unsigned>> & scores);
```

```
void insertScore(vector<pair<string, unsigned>> & scores,  
const string & nom, const unsigned & score,  
const bool & popBack);
```

Projet Space Invaders

G2 - G4

BENNACER Chakib

LESBROS Raphaël

MATHIEU--DRIF Clément

PONSARD Nils

TRITZ Hugo