

# 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)



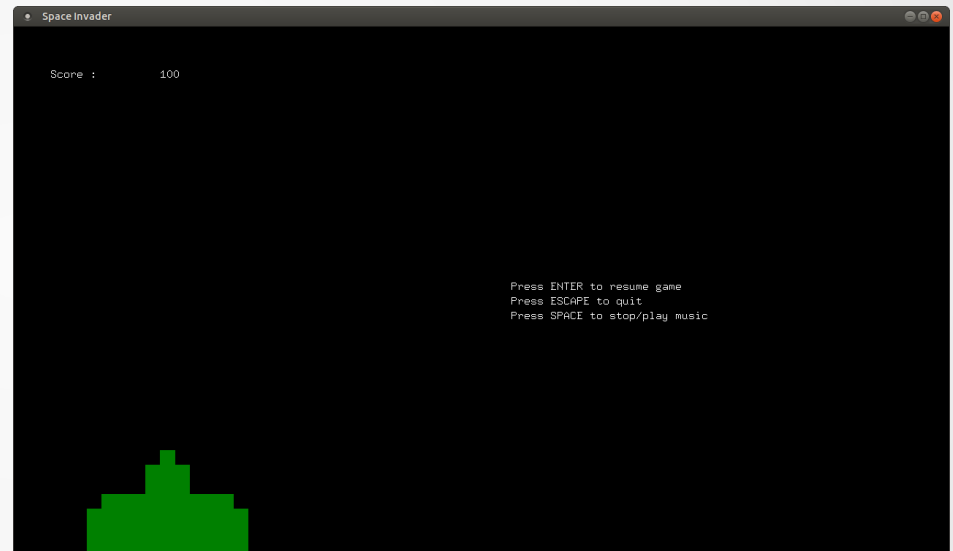
Jeu : vague 1

# 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](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 invaders;
    entity player;
    entity playerTorpedo;
    entity invadersTorpedo;
    entity bonusInvader;

    BonusEntity upgrades;

    ...
};
```

```
struct entity
{
    unsigned entityHeight;
    unsigned entityWidth;

    figure entityFig;
};
```

Passés en paramètre de la  
fonction de gestion des  
collisions

```
struct BonusEntity
{
    unsigned entityHeight;
    unsigned entityWidth;

    std::vector<figure> entityFigs;
};
```

# Gestion des positions

```
struct spaceInvaders
{
    ...

    pos playerPos;
    pos bonusInvaderPos;
    std::vector<std::pair<pos, unsigned>> invadersPos;
    std::vector<pos> playerTorpedoPos;
    std::vector<pos> invadersTorpedoPos;

    std::vector<std::pair<pos, short>> UpgradePos;

    ....
};
```

# Fréquence d'images

```
while (!iLoose)
{
    chrono::time_point<chrono::steady_clock> beg(chrono::steady_clock::now());

    //fonction principale du jeu
    //affichage

    chrono::time_point<chrono::steady_clock> end(chrono::steady_clock::now());
    chrono::duration<double, milli> diff(end - beg);
    if (diff < frameDuration)
        this_thread::sleep_for(frameDuration - diff);
}
```



# Gestion du temps et des vitesses

```
struct spaceInvaders
```

```
{
```

```
...
```

```
std::chrono::duration<int, std::milli> shot;
```

```
std::chrono::time_point<std::chrono::steady_clock> lastShot;
```

```
std::chrono::duration<int, std::milli> invadersShot;
```

```
std::chrono::time_point<std::chrono::steady_clock> invadersLastShot;
```

```
std::chrono::duration<int, std::milli> bonusInvaders;
```

```
std::chrono::time_point<std::chrono::steady_clock> LastBonusInvader;
```

```
unsigned torpedoVelocity;
```

```
unsigned invadersVelocity;
```

```
unsigned invadersMaxVelocity;
```


```
unsigned invadersMinVelocity;
```

```
unsigned upgradeVelocity;
```

```
unsigned bonusInvaderVelocityFactor;
```

```
...
```

```
};
```



En px par image

# Audio

```
system("bash ../ressources/audioSpaceInvaders.bash &");
```



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

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

```
system("aplay -q ../ressources/loseLife.wav" &);
```

# 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 std::pair<int, bool> keyType;
```

```
typedef std::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
{
    /* List of authorized key for the type char in a struct CMyParam*/
    // const std::vector<std::string> VParamChar {};
    /* List of authorized key for the type string in a struct CMyParam*/
    // const std::vector<std::string> VParamString {};
    /* List of authorized key for the type unsigned in a struct CMyParam*/
    const std::vector<std::string> VparamUnsigned{"invadersMaxVelocity".....};
};
```

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

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

# Chargement du fichier de score

```
vector<pair<string, unsigned>> loadScores(const string &fileName)  
//Charge au maximum 3 scores (les premiers du fichier)
```

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

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

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

Dépend du nombre de scores à conserver

# Projet Space Invaders

G2 - G4

BENNACER Chakib

LESBROS Raphaël

MATHIEU--DRIF Clément

PONSARD Nils

TRITZ Hugo