

Alexandre Doukhan

ETUDIANT

BAC et BTS
Systèmes Numériques
Réseaux et Communications

En recherche active
d'un contrat pro en
alternance

CONTACT

E-mail :

alexandre.doukhan@hotmail.com

Tél : 06.25.03.06.18

Adresse : Cannes marina
06210 Mandelieu la Napoule

LOISIRS

Musique
Bricolage
Modélisation 3D
Informatique
Jeux Vidéo

COMPÉTENCES

Programmation :

Web: HTML5 | CSS | PHP | SQL | JS

Langage: C | C# | C++ | VHDL

Anglais: Niveau B2

Logiciels :

Photoshop Cs6 | Vegas Pro 13

Proteus | SolidWorks | MindView

Arduino IDE | Unity 3D | Blender

FORMATION

- **Bachelor Software Development Specialist** Campus ID Sophia Antipolis - 2018
- **BTS Systèmes Numériques** Lycée Guillaume Apollinaire Nice- 2018
Développement d'objets interconnectés, conception d'architecture matérielle/logicielle
Programmation sur microcontrôleurs:
Arduino (ATmega328), Afficheur à pastilles MEGABUS (ATmega8515), Afficheur de gare MAT (ATmega48), Kit basys2 ISE (AT90USB2), HUB Gestion Technique et Energétique WIT.
Matériel de scène : Console DMX CMX24, gradateur dp615
Réseau : CISCO Packet tracer, Création de sites web HTML5/CSS, BDD Mysql/PHP/JavaScript, administration réseau LINUX sur Raspberry. ([voir Projet page suivante](#))
- **BAC Systèmes Électroniques Numériques** Lycée Alfred Hutinel Cannes - 2016
Installation systèmes embarqués, sécurité et alarmes, informatique et réseaux
- **Seconde Création et Innovation Technologique** Lycée Jules Ferry Cannes - 2015
Modélisation 3D création assisté par ordinateur et systèmes domotique
- **BEP Systèmes Électroniques Numériques** Lycée Alfred Hutinel Cannes - 2015
- **Diplôme national du brevet** Mandelieu-la-Napoule - Juillet 2013

EXPERIENCE

Projets personnels :

Développement de jeux vidéos : programmation sur Unity 3D en C# et JS de la logique de gameplay, d'IA de bot, et d'UI, modélisation d'assets sur Blender, design graphique des menus, textures, HUD, et sprites sur Photoshop Cs6.

Bricolage : programmation sur Arduino de systèmes divers avec matériaux de récupération
Programmation divers (Sites web, bot Discord, mods), remise en état d'appareils (ordinateurs, consoles, électroménager),

- **Technicien S.A.V Informatique** Antibes Informatique - Février / Mars 2017
Dépannage d'ordinateurs, montage PC, installation d'OS et admin réseau windows
- **Technicien de maintenance** Station service BP - Juillet / Août 2017
- **Agent de maintenance technique** C.C.I Port de Cannes - Avril / Mai 2016
- **Maintenance matériel de scène** MJC Picaud Cannes - Mars / Avril 2015
- **Technicien S.A.V Multimédia** S.T.E Le Cannet - Février / Mars 2014
- **Technicien S.A.V Informatique** Boulanger Mandelieu - Février / Mars 2013

PROJET : Développement de jeux vidéo

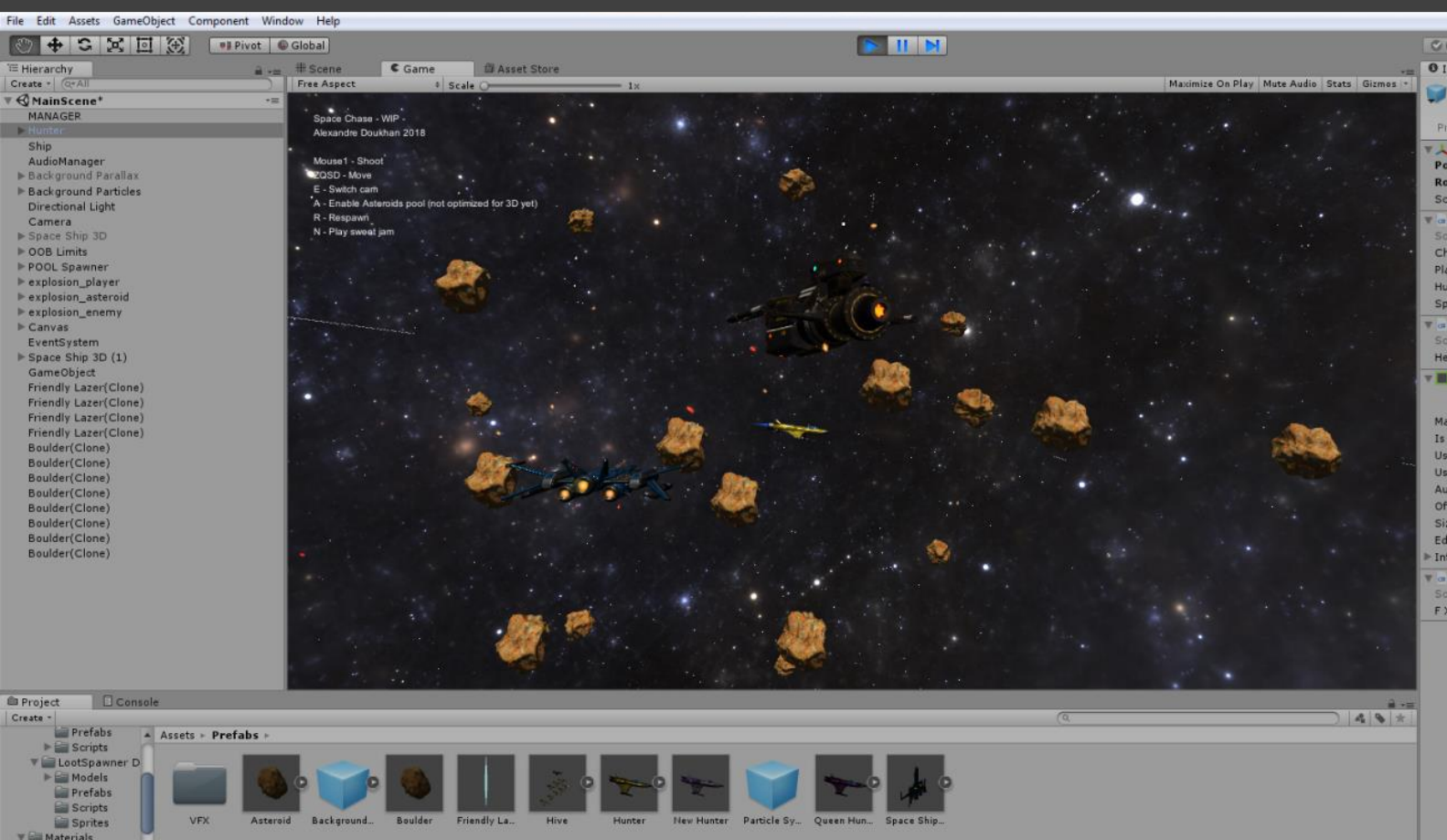
Modélisation Blender :

- Modélisation de personnages réaliste à l'aide de l'outil de sculptage de Blender
- Textures et Maps de déplacement créées sous Photoshop et Substance Painter



Programmation sur Unity C#

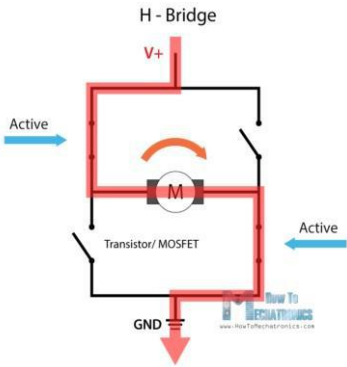
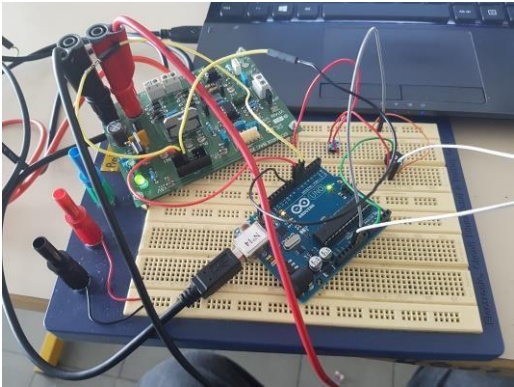
Programmation d'outils et de jeux vidéo dans l'environnement d'Unity 3D



PROJET: Réalisation d'une ruche autonome:
Programmation sur Arduino:

- Control Manuel/Automatique de deux moteurs (Azimute et élévation) pour l'orientation du panneau solaire de la ruche afin de suivre en permanence le soleil.

```
switch (lcd_key)
{
    // selon le bouton
    case btnUP:
    {
        digitalWrite($n1, LOW);
        digitalWrite($n2, HIGH);
        break;
    }
    case btnDOWN:
    {
        digitalWrite($n1, HIGH);
        digitalWrite($n2, LOW);
        break;
    }
    case btnLEFT:
    {
        digitalWrite($n3, LOW);
        digitalWrite($n4, HIGH);
        break;
    }
    case btnRIGHT:
    {
        digitalWrite($n3, HIGH);
        digitalWrite($n4, LOW);
        break;
    }
    case btnSELECT:
    {
        menu=menu-2;
        lcd.clear();
        break;
    }
    case btnNONE:
    {
        digitalWrite($n1, LOW);
        digitalWrite($n2, LOW);
        digitalWrite($n3, LOW);
        digitalWrite($n4, LOW);
        break;
    }
}
```

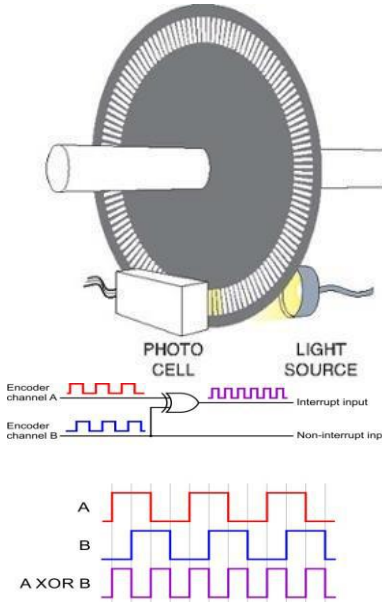


- Programmation de deux codeurs incrémentaux destinés à compter les tours des moteurs afin d'avoir l'azimute et l'élévation du panneau solaire en degrés

```
void doEncoder()
{
    if (digitalRead(encoder0PinA) == digitalRead(encoder0PinB))
    {
        encoder0Pos++;
    }
    else
    {
        encoder0Pos--;
    }
    degreeSp = map (encoder0Pos,0,79300,0,360);
}

void doEncoder1()
{
    if (digitalRead(encoder0PinC) == digitalRead(encoder0PinD))
    {
        encoder0Pos1++;
    }
    else
    {
        encoder0Pos1--;
    }
    degreeSp1 = map (encoder0Pos1,0,79300,0,360);
}
```

```
if (menu==2)
{
    lcd.setCursor(0,1);
    lcd_key = read_LCD_buttons(); // lire
    lcd.write(byte(3));
    lcd.write(byte(2));
    //lcd.print("=");
    lcd.setCursor(3,1);
    if(degreeSp<10) lcd.print(" ");
    if(degreeSp>=10) lcd.print(" ");
    if(degreeSp>=100) lcd.setCursor(3,1);
    lcd.print(degreeSp);
    lcd.setCursor(8,1);
    lcd.write(byte(1));
    lcd.write(byte(0));
    if(degreeSp1<10) lcd.print(" ");
    if(degreeSp1>=10) lcd.print(" ");
    if(degreeSp1>=100) lcd.setCursor(13,1);
    lcd.print(degreeSp1);
}
```



- Interface de control sur un écran lcd 16x2 avec boutons de navigations afin de permettre à l'utilisateur de visualiser les consignes d'élévation et azimute, et d'ajuster manuellement le panneau solaire à l'aide des boutons ou de passer le system en mode d'ajustement automatique.

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);

// if (menu==3)
{
    lcd.print("Quit.Ajust.Auto?");
    lcd.setCursor(3,1);
    lcd.print("OUI NON");
    lcd.setCursor(0,1);
    lcd.write(byte(3));
    lcd.setCursor(15,1);
    lcd.write(byte(2));
    lcd_key = read_LCD_buttons(); // lire la valeur du switch (lcd_key)
    {
        case btnLEFT:
        {
            menu=menu-3;
            lcd.clear();
            break;
        }
        case btnRIGHT:
        {
            menu=menu-2;
            lcd.clear();
            break;
        }
    }
}

// ===== VARIABLES =====

#define btnRIGHT 0
#define btnUP 1
#define btnDOWN 2
#define btnLEFT 3
#define btnSELECT 4
#define btnNONE 5
int menu = 0;
int adc_key_in = 0;
int lcd_key = 0;

int read_LCD_buttons()
{
    adc_key_in = analogRead(0);
    if (adc_key_in > 1000) return btnNONE;
    if (adc_key_in < 50) return btnRIGHT;
    if (adc_key_in < 250) return btnUP;
    if (adc_key_in < 450) return btnDOWN;
    if (adc_key_in < 650) return btnLEFT;
    if (adc_key_in < 850) return btnSELECT;
    return btnNONE;
}
```

