

# Présentation de PRE Final

Interaction 2D en réalité virtuelle

Ji YANG

Tuteur: Laurant Grisoni

## **Le planning**

- Présentation du contexte
- Présentation du cahier des charges
- Présentation du travail effectué
- Présentation de la conclusion et la poursuite



SANTA CLAUS IN TROUBLE... AGAIN!

WIN

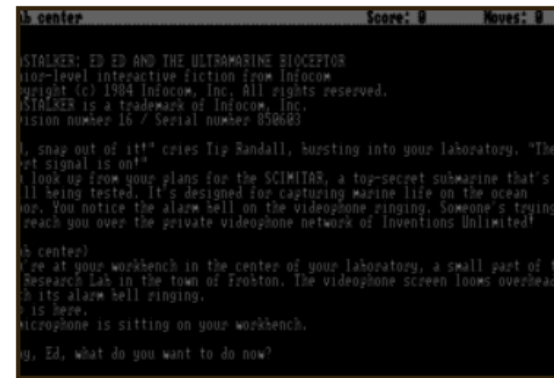
2004



SPACE HARRIER

DOS, C64, MASTER SYSTEM, GAME GEAR,  
AMIGA, AMSTRAD CPC, ATARI ST, PC-88, SEGA  
32X

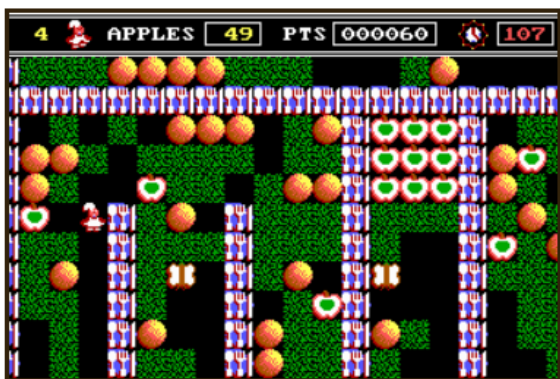
1989



SEASTALKER

DOS, MAC, C64, AMIGA, AMSTRAD CPC, ATARI 8-  
BIT, APPLE II

1984



ROCKFORD: THE ARCADE GAME

DOS, C64, AMIGA, AMSTRAD CPC, ATARI ST

1987



STAR CONTROL II

DOS

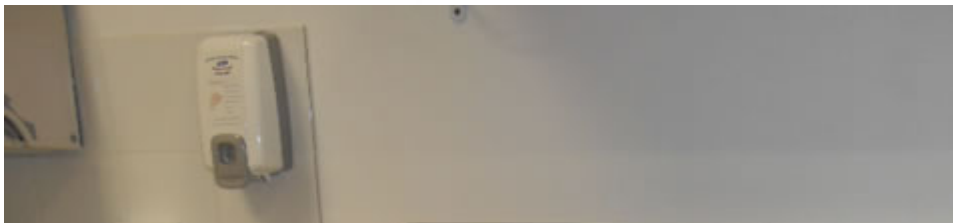
1992



HOOK

DOS, AMIGA, ATARI ST

1992



**Enfant cérébrolese qui fait**  
[http://www.romansferrari.fr/  
ergotherapie\\_centre\\_romans\\_ferrari.html](http://www.romansferrari.fr/ergotherapie_centre_romans_ferrari.html)

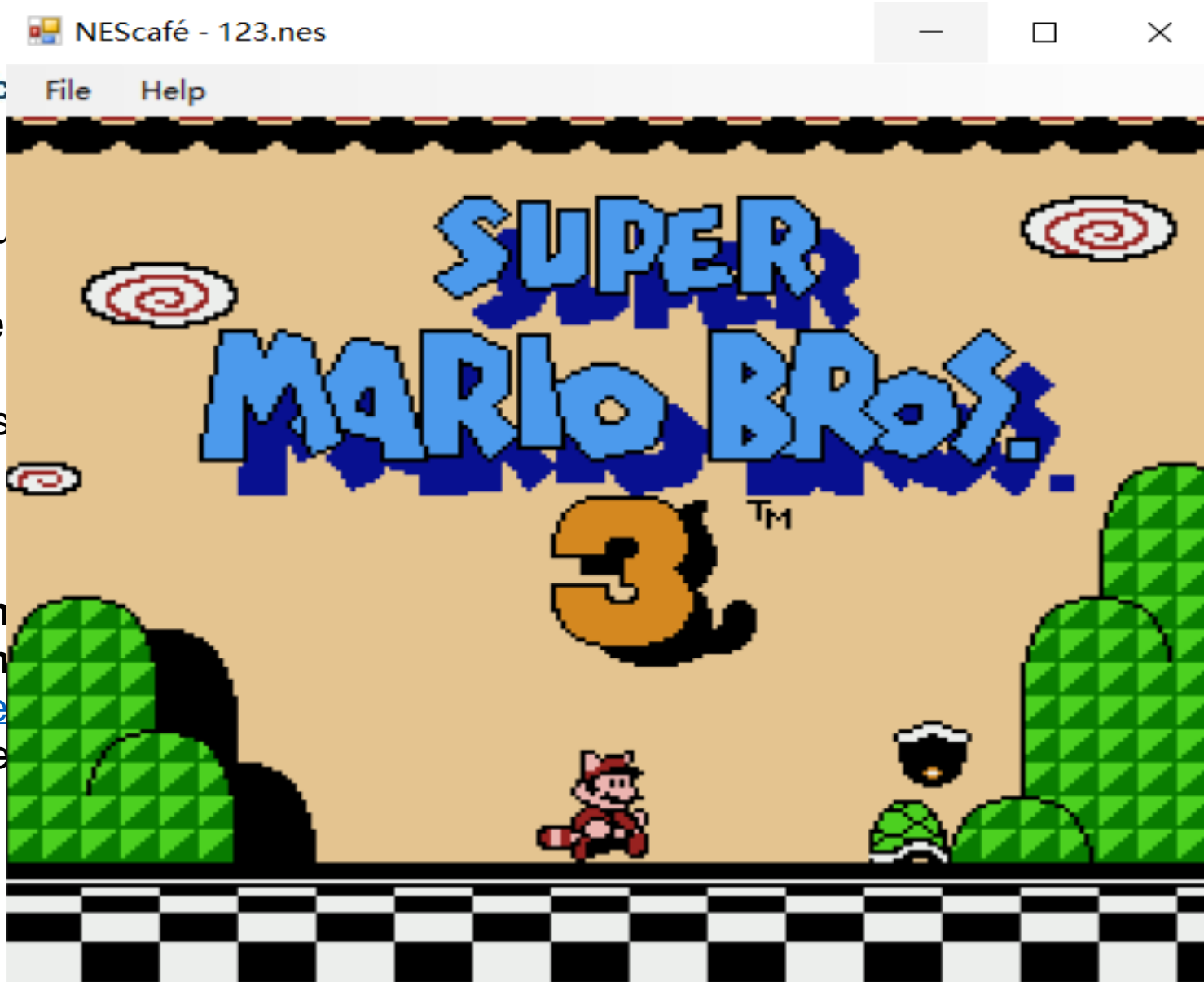
## Présentation du contexte

### 3. La limite du jeux vidéo sur l'ordinateur

La taille d'écran de l'ordinateur est fixée. En générale, elle est trop petite pour les enfant cérébrolésé.  
Par contre, avec la réalité virtuelle, on peut changer facilement la taille d'écran virtuel.

## Présentation du cahier des charges

1. On peut afficher l'image dans la réalité virtuelle avec un casque virtuel.
2. On peut changer la taille d'écran virtuel.
3. On peut faire l'interaction, avec clavier ou souris, avec des jeux vidéo abandonnés par un émulateur de jeux vidéo.
4. On peut faire le 'touch' à partir des mouvement des mains.
5. C'est meilleur qu'on peut explorer les modalité d'interaction.



Présentation du

1. Trouvé et te

Opens

NES:

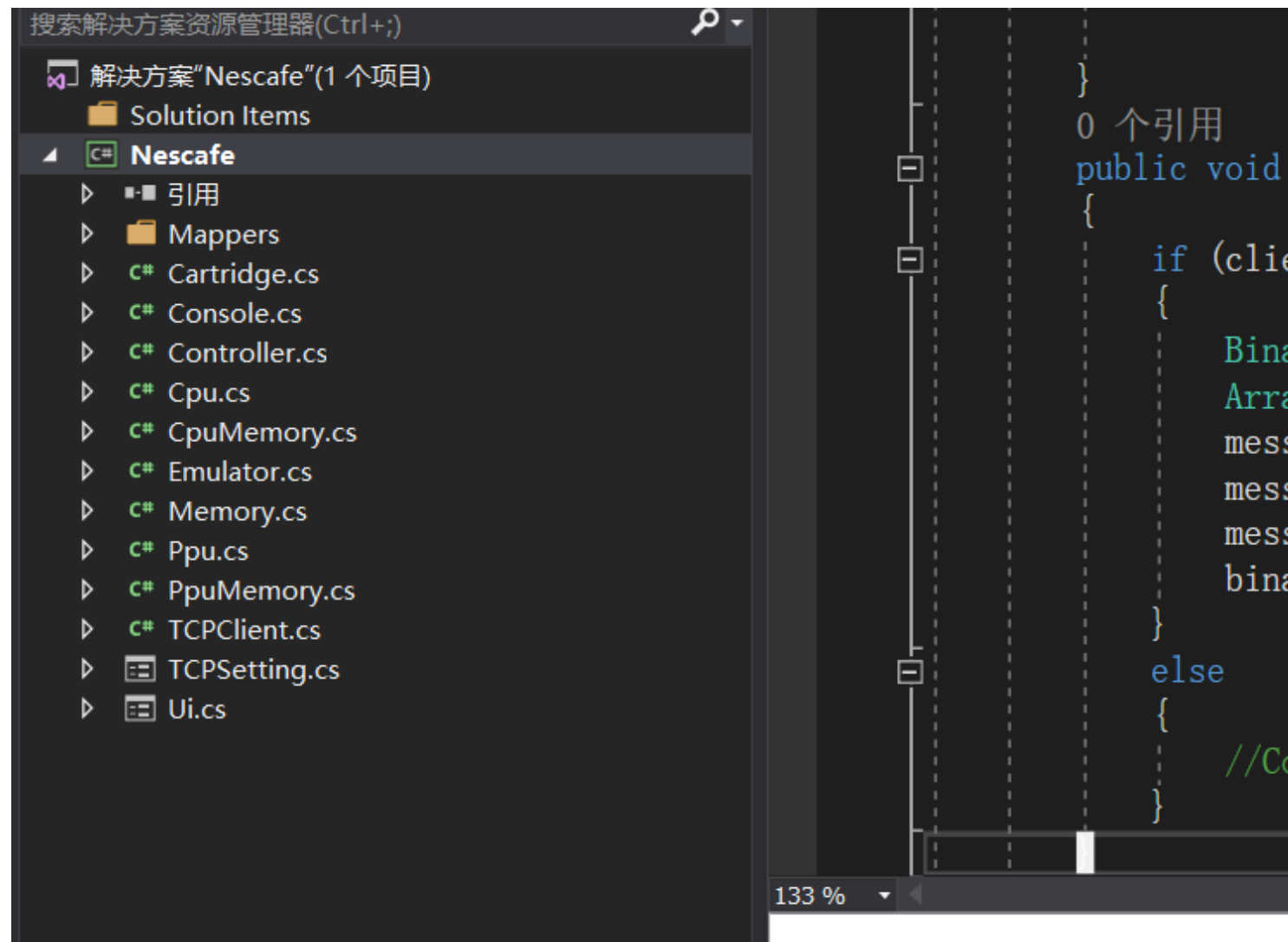
La ou le Ninten  
appelée Ninten  
japonaise [Ninte](https://fr.wikipedia)  
([https://fr.wikip](https://fr.wikipedia)

[Penguin/nesc](https://penguin.nescafe))

ment  
par l'entreprise

## Présentation du travail effectué

NEScafé:



The screenshot displays the Visual Studio IDE. On the left, the Solution Explorer shows a project named 'Nescafe' with the following files and folders:

- 引用 (References)
- Mappers
- Cartridge.cs
- Console.cs
- Controller.cs
- Cpu.cs
- CpuMemory.cs
- Emulator.cs
- Memory.cs
- Ppu.cs
- PpuMemory.cs
- TCPClient.cs
- TCPSetting.cs
- Ui.cs

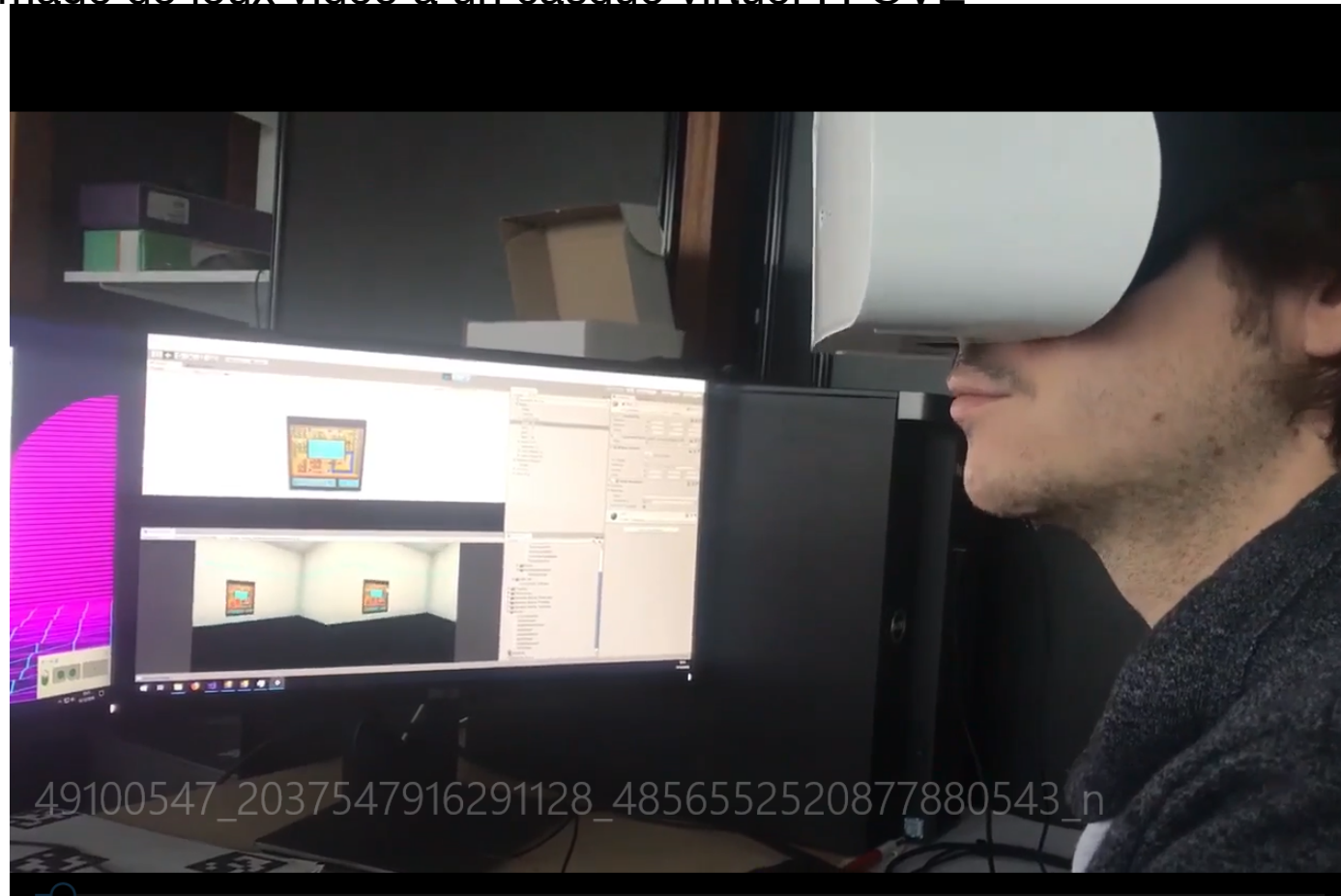
The main editor area shows a C# code snippet with a search bar at the top. The code includes a method signature and an if-else block:

```
0 个引用  
public void S  
{  
    if (clie  
{  
        Binar  
        Array  
        messa  
        messa  
        messa  
        binar  
    }  
    else  
{  
        //Co
```



## Présentation du travail effectué

### 2. Réalisé à afficher l'image de jeux vidéo à un casque virtuel : FOVE



Présentation du travail effectué

2. Réalisé à afficher l'image de jeux vidéo à un casque virtuel : FOVE



Quad

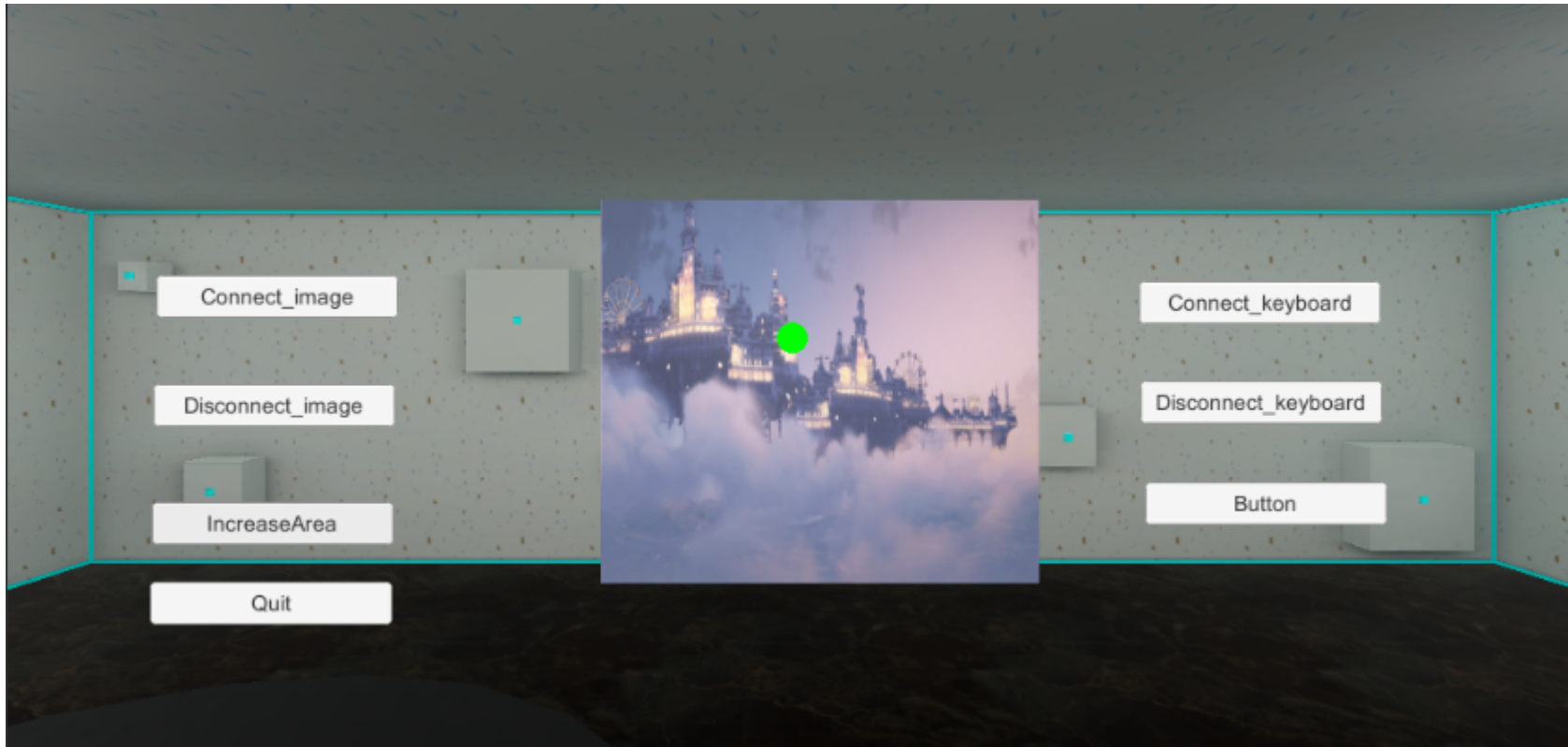
## Présentation du travail effectué

### 2. Réalisé à afficher l'image de jeux vidéo à un casque virtuel : FOVE

```
private void ReceiveImage()  
{  
    BinaryFormatter binaryFormatter = new BinaryFormatter();  
    while (client.Connected)  
    {  
        if (client.Available > 0)  
        {  
            mainStream = client.GetStream();  
            //pictureBox1.Image = (Image)binaryFormatter.Deserialize(mainStream);  
            isImageCanUse = false;  
            local_image = (System.Drawing.Image)binaryFormatter.Deserialize(mainStream);  
            isImageCanUse = true;  
        }  
    }  
}
```

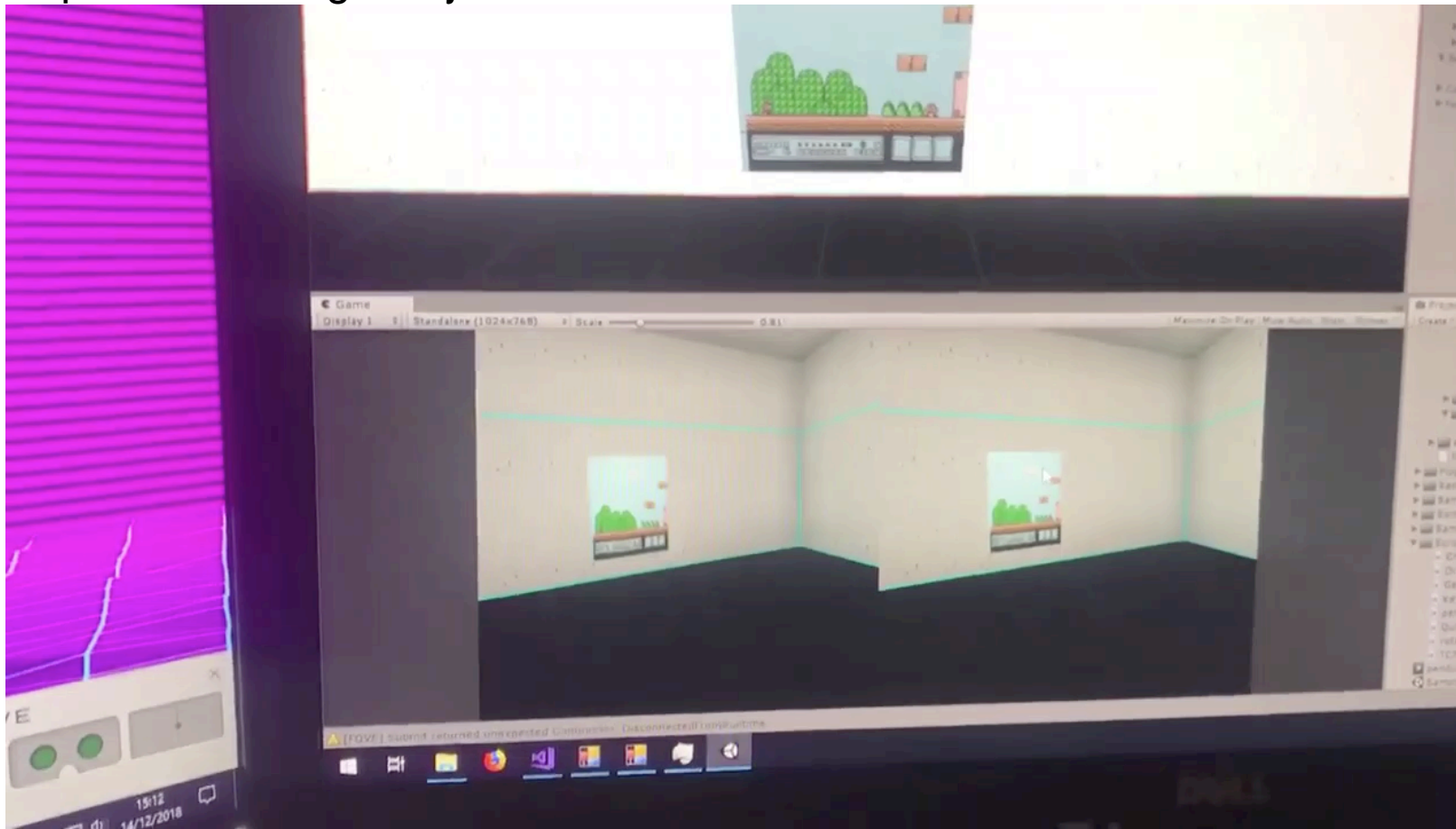
## Présentation du travail effectué

### 3. On peut changer la taille de l'image de jeux vidéo



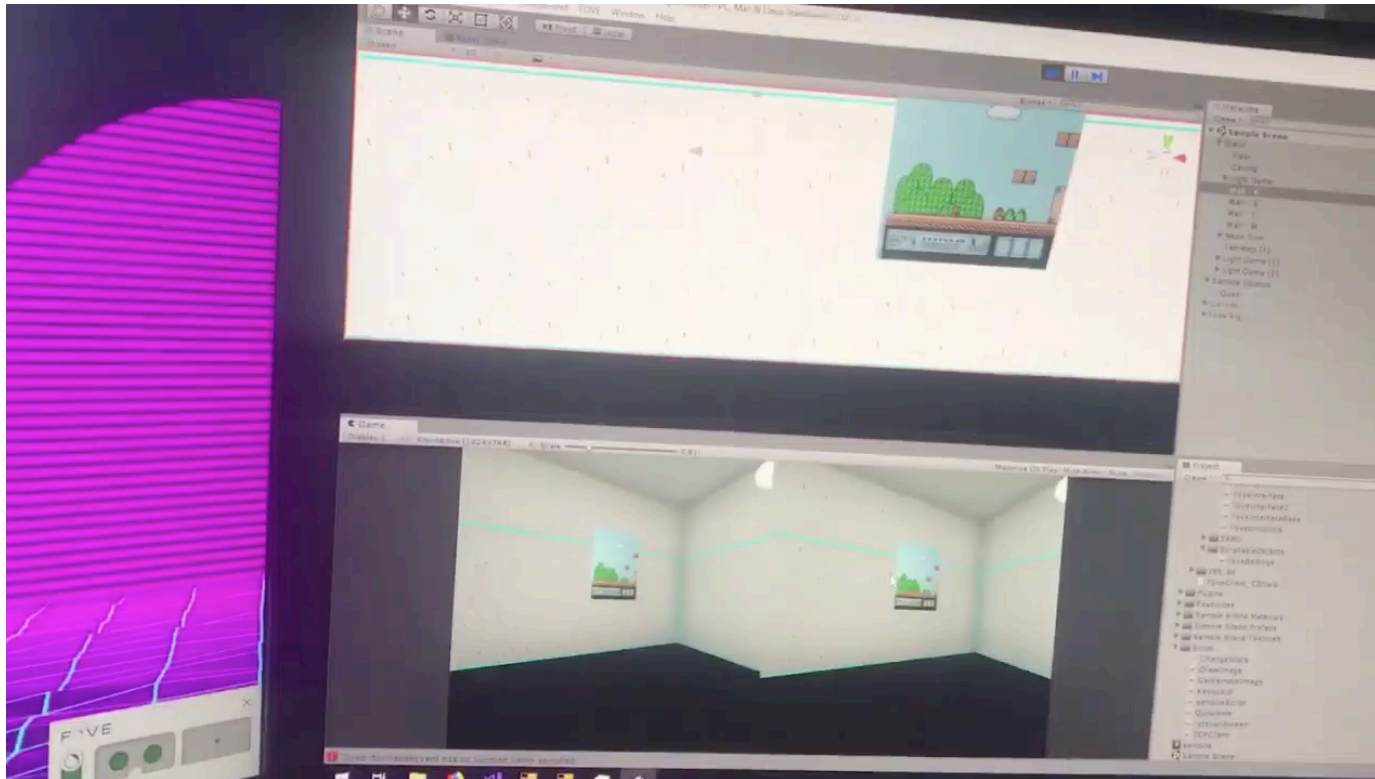
## Présentation du travail effectué

### 4. On peut tourner la position d'image de jeux vidéo



## Présentation du travail effectué

5. On peut faire l'interaction avec le jeux vidéo par clavier et souris.



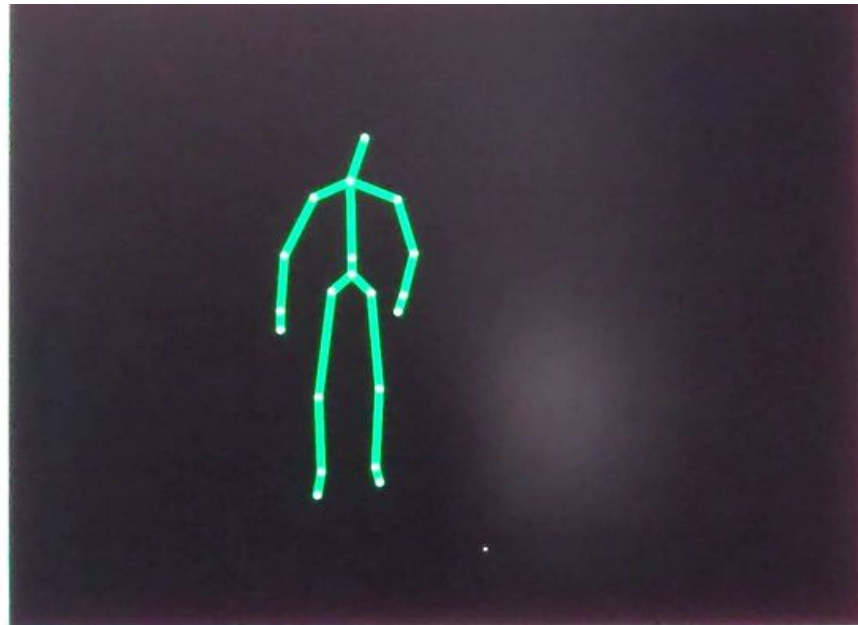
## Présentation du travail effectué

### 6. Réaliser à contrôler le jeux vidéo à partir des mouvement des mains



Présentation du travail effectué

6. Réaliser à contrôler le jeux vidéo à partir des mouvement des mains



L'image des joints de Kinect



```
private void SensorSkeletonFrameReady(object sender, SkeletonFrameReadyEventArgs e)
{
    Skeleton[] skeletons = new Skeleton[0];
    using (SkeletonFrame skeletonFrame = e.OpenSkeletonFrame())
    {
        if (skeletonFrame != null)
        {
            skeletons = new Skeleton[skeletonFrame.SkeletonArrayLength];
            skeletonFrame.CopySkeletonDataTo(skeletons);
        }
    }
    using (DrawingContext dc = this.drawingGroup.Open())
    {
        dc.DrawRectangle(Brushes.Black, null, new Rect(0.0, 0.0, RenderWidth, RenderHeight));
        if (skeletons.Length != 0)
        {
            foreach (Skeleton skel in skeletons)
            {
                RenderClippedEdges(skel, dc);
                if (skel.TrackingState == SkeletonTrackingState.Tracked)
                {
                    this.DrawBonesAndJoints(skel, dc);
                }
            }
        }
    }
}
```

L'image des joints de Kinect

## Présentation du travail effectué

### 7. UI

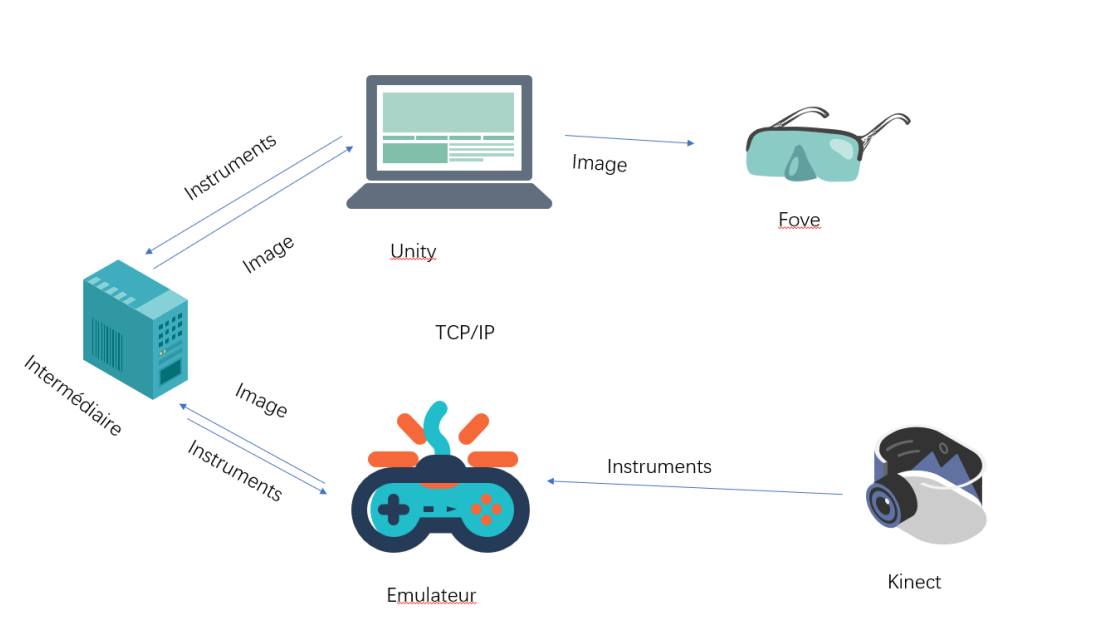


## Présentation du travail effectué

### 8. Structure de système

Intermédiaire est un le serveur de TCP/IP

Il y un client de TCP/IP dans l'Unity et l'émulateur





## Conclusion

1. Réalisé à afficher l'image de jeux vidéo en casque virtuel
2. Réalisé à contrôler le jeux vidéo par clavier et souris
3. Réalisé à construire un système à simuler la 'clique' à partir de mouvement des mains de l'utilisateur

## Poursuite du Projet

1. Résoudre le problème de stabilité
2. Augmenter la précision de la détection des mains par Kinect.

Source de code:

<https://github.com/guigui00700/Interaction2DinVR>

**Merci de regarder  
Et  
votre question?**