

Présentation de PRE Intermédiaire

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

Présentation

1. Il y a certains propriétaires



SANTA CLAUS IN TROUBLE... AGAIN!

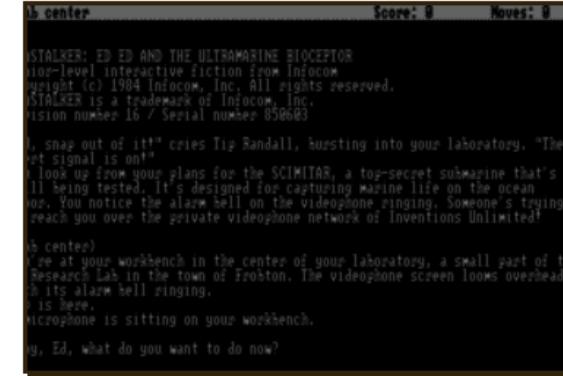
WIN

2004



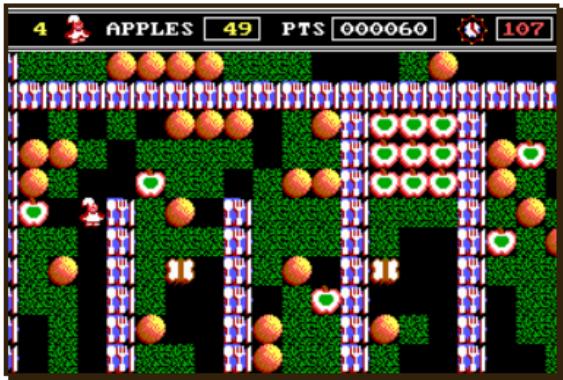
SPACE HARRIER

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



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



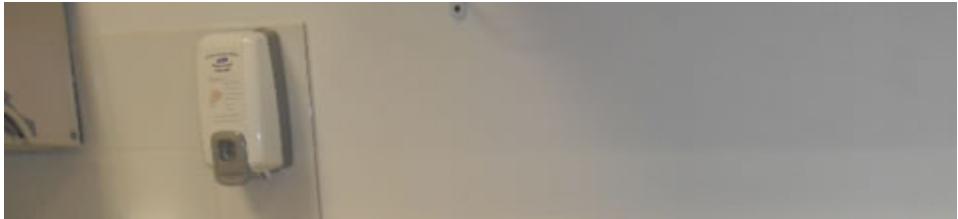
STAR CONTROL II

DOS



HOOK

DOS, AMIGA, ATARI ST



Enfant cerebrolese qui fait

<http://www.romansferrari.fr/>

ergotherapie_centre_romans_ferrari.html

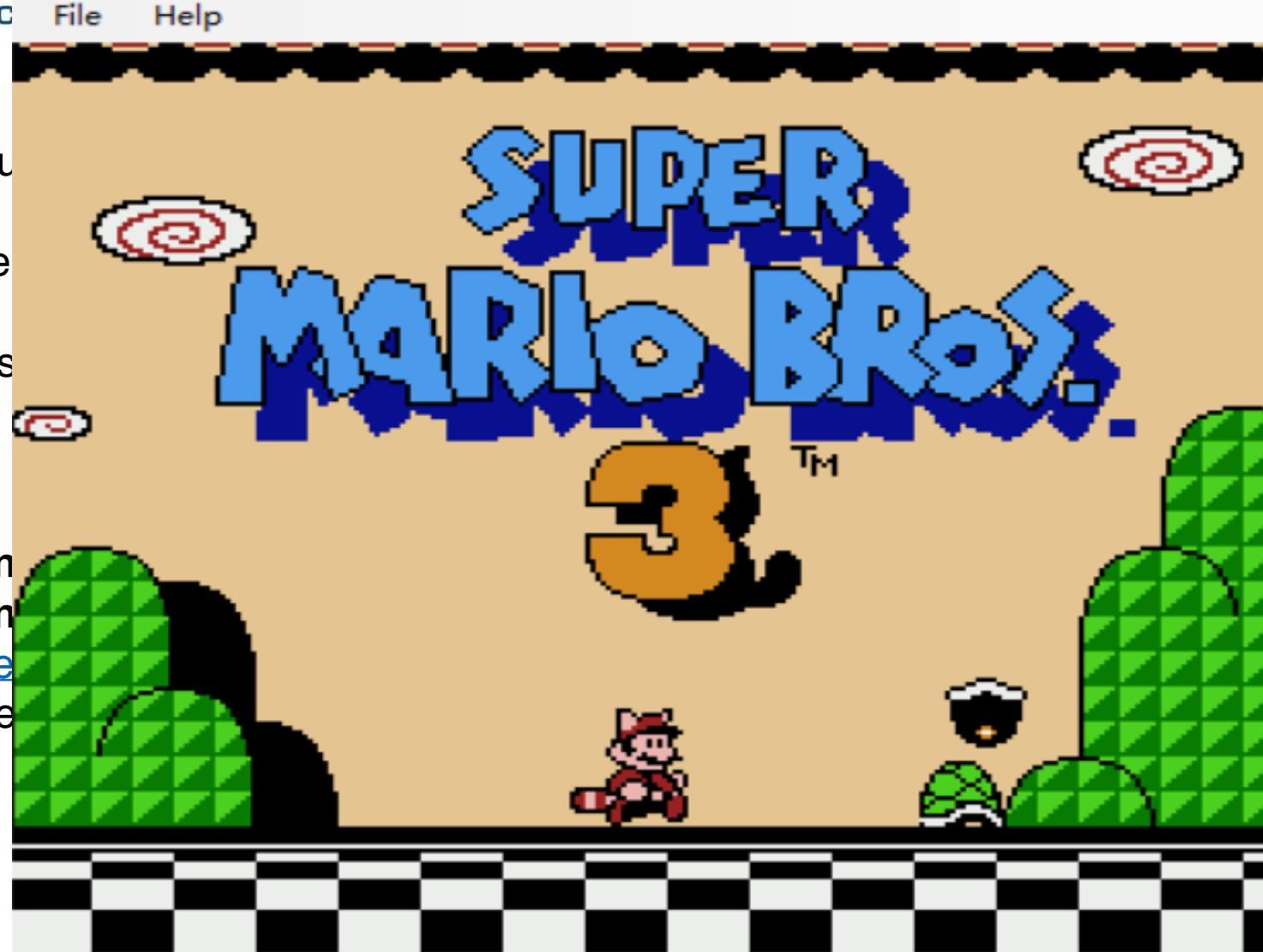
Présentation du contexte

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

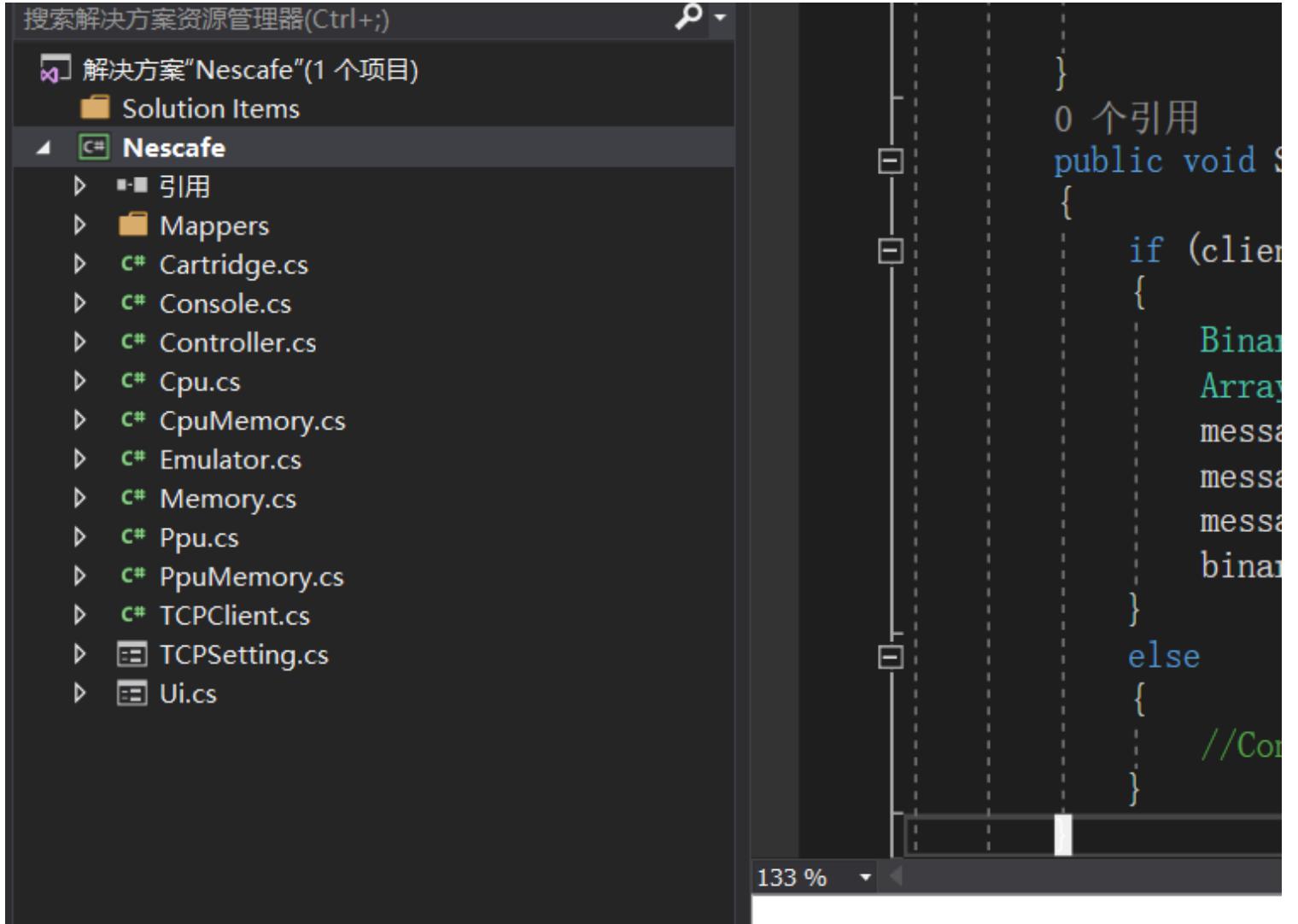
OpenS

NES:

La ou le Ninten
appelée Ninten
japonaise [Ninte](#)
(<https://fr.wikipedia.org>

Présentation du travail effectué

NEscafé:



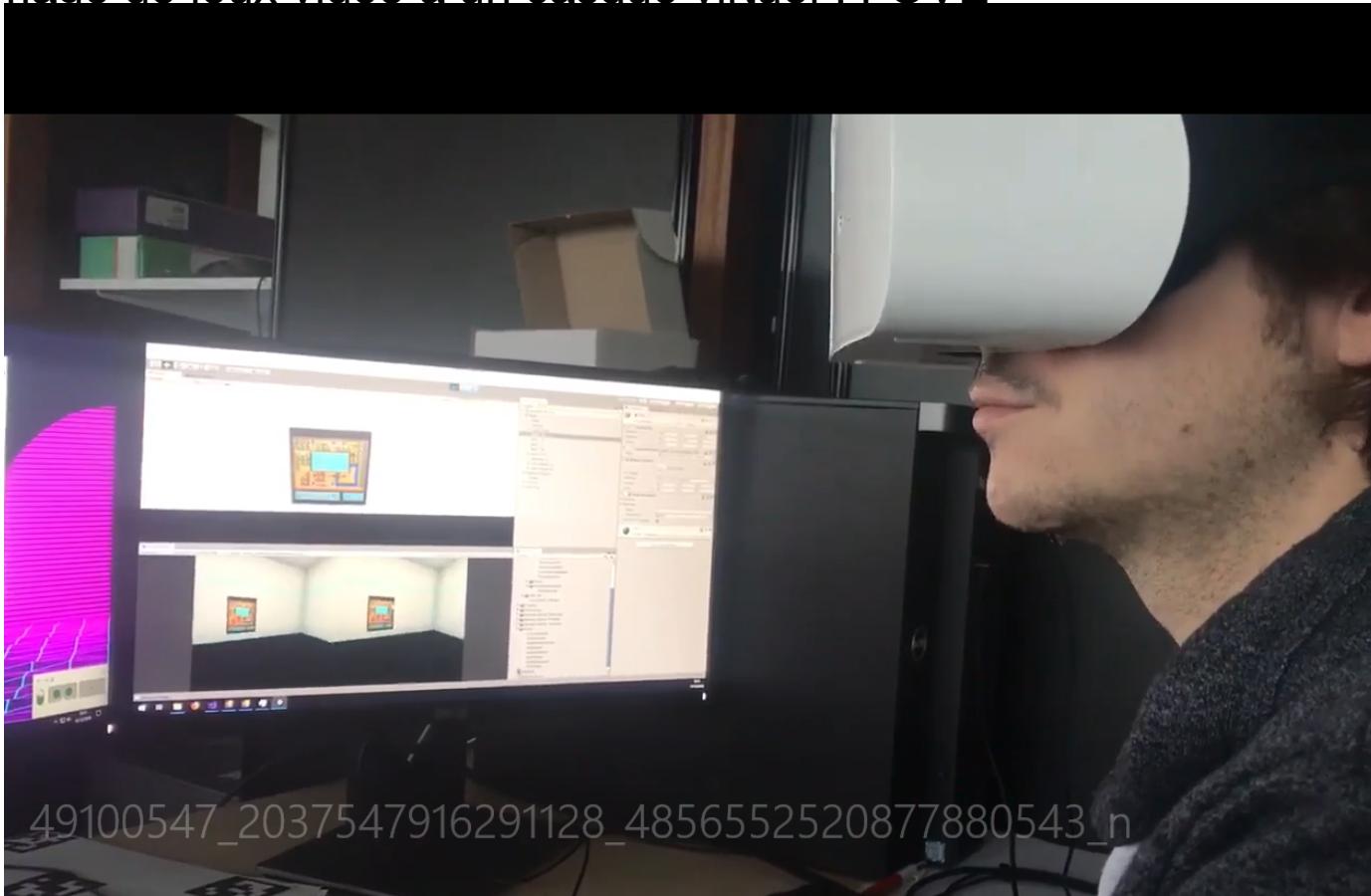
The screenshot shows a dark-themed IDE interface. On the left, the Solution Explorer displays a single project named "Nescafe" containing a "Solution Items" folder and several C# files: Cartridge.cs, Console.cs, Controller.cs, Cpu.cs, CpuMemory.cs, Emulator.cs, Memory.cs, Ppu.cs, PpuMemory.cs, TCPClient.cs, TCPSetting.cs, and Ui.cs. On the right, the code editor shows a portion of a C# class with the following code:

```
        }  
        0 个引用  
        public void S  
        {  
            if (client  
            {  
                Binary  
                Array  
                messa  
                messa  
                messa  
                messa  
                binary  
            }  
            else  
            {  
                //Cor  
            }  
        }  
    }  
}
```

The status bar at the bottom indicates a zoom level of 133%.

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

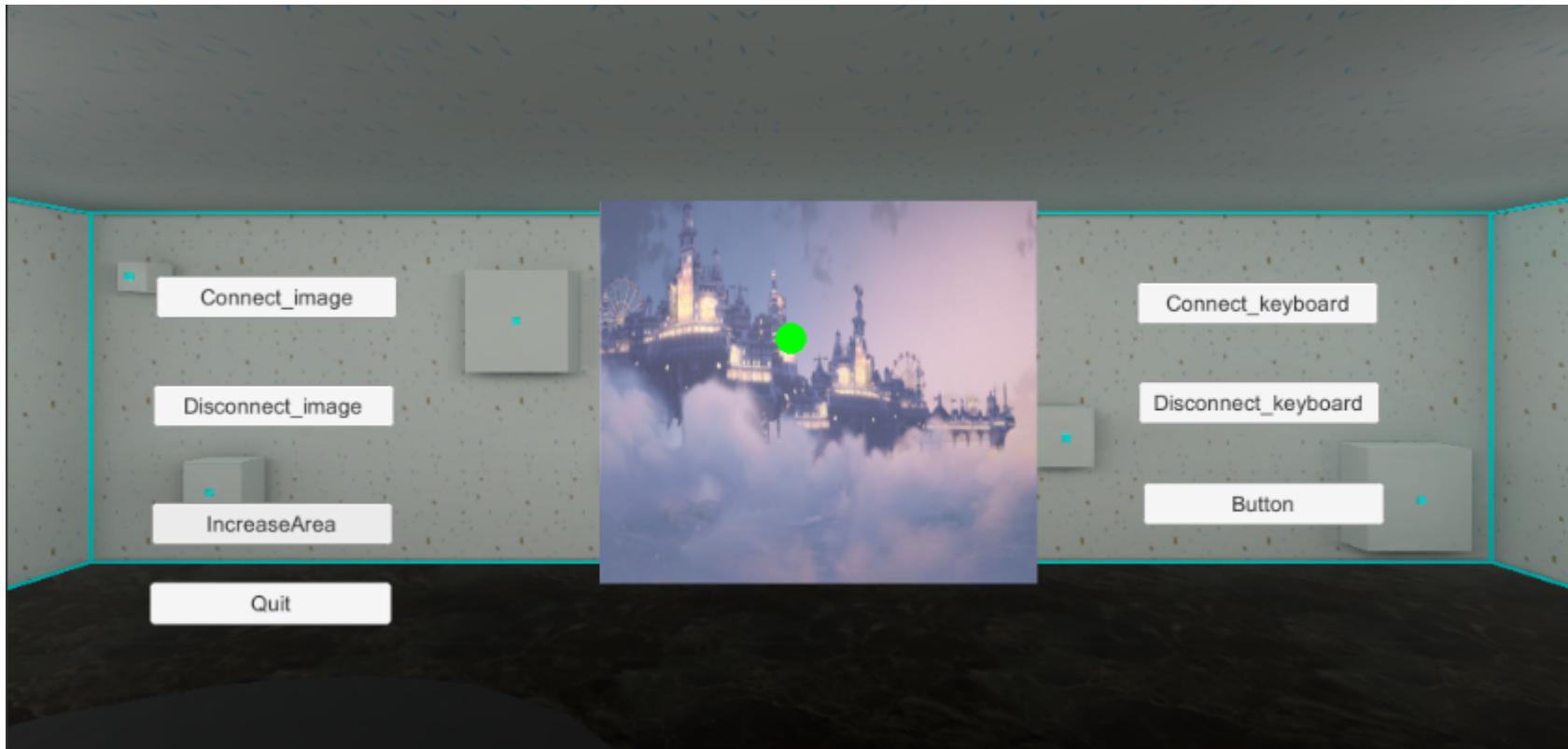
Présentation du travail effectué

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

```
    [不引用]
    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;
            }
        }
    }
```

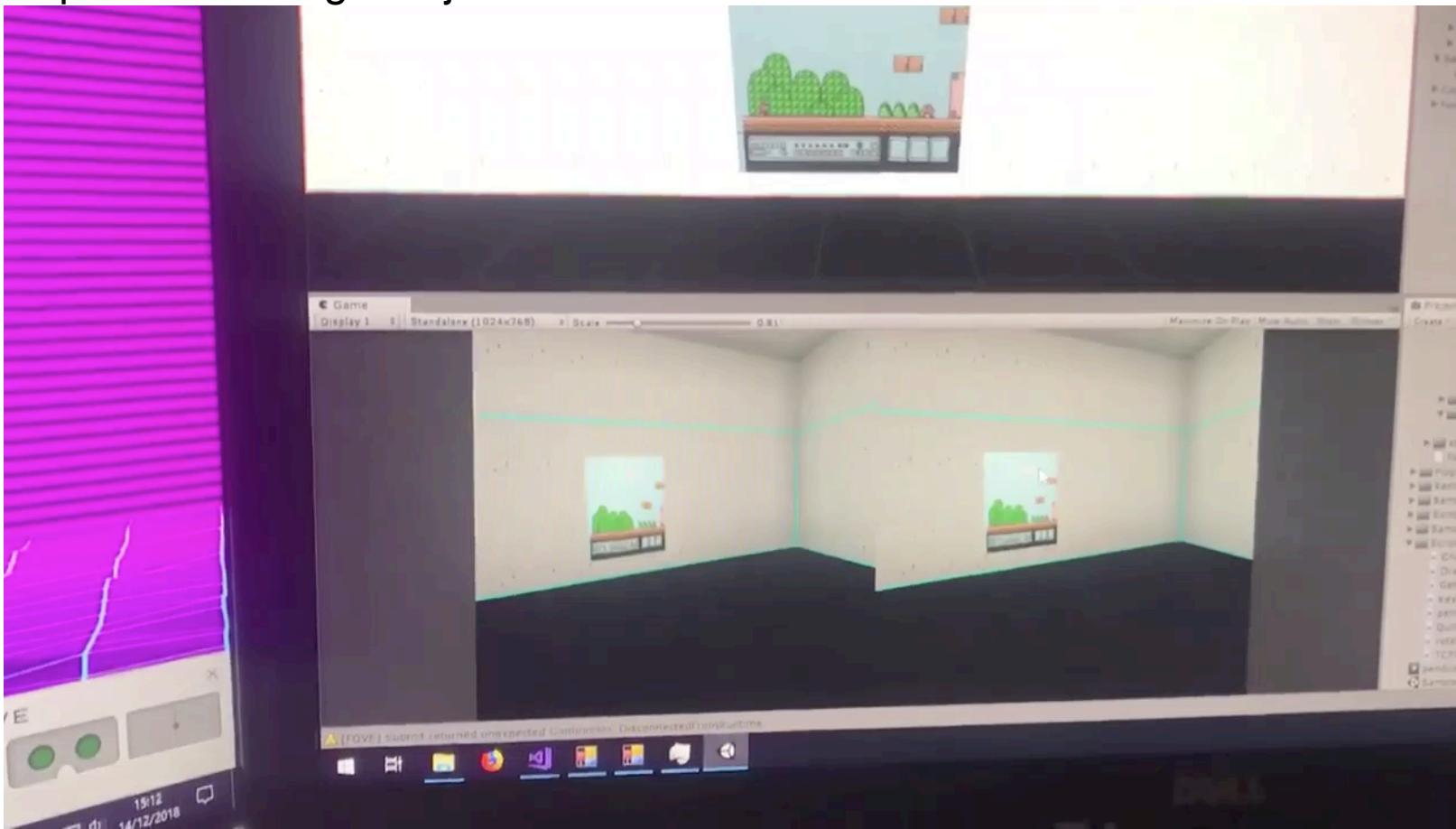
Présentation du travail effectué

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



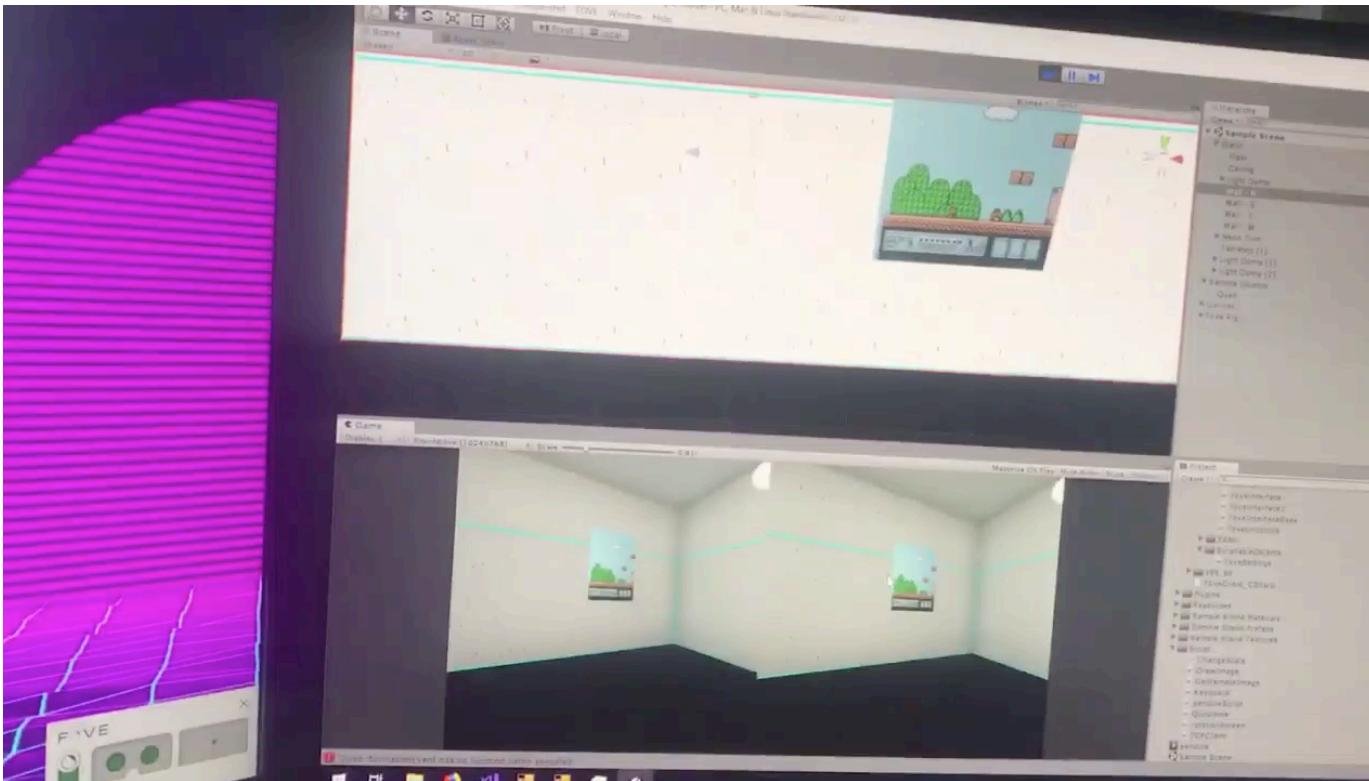
Présentation du travail effectué

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.



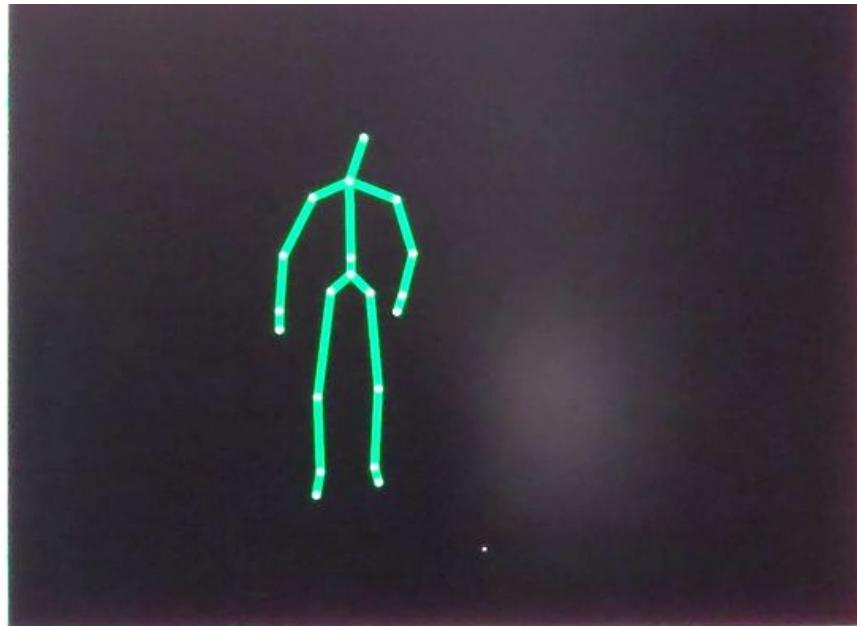
Présentation du travail effectué

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



Présentation du travail effectué

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



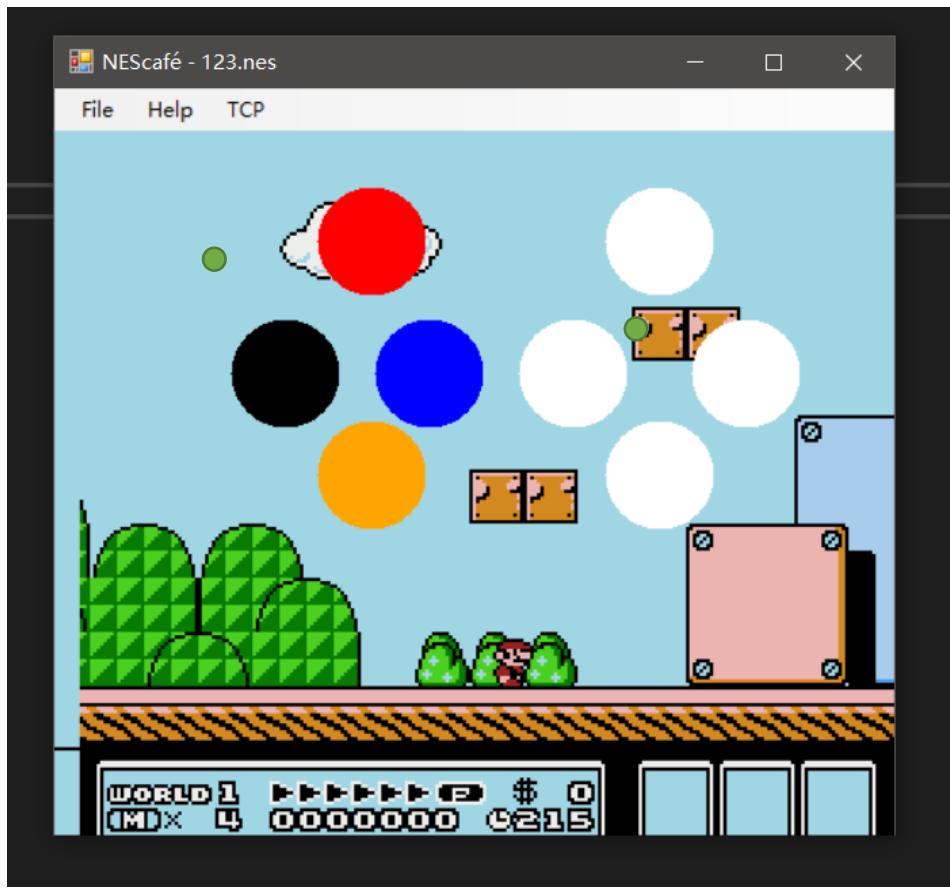
L'image des joints de Kinect

```
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

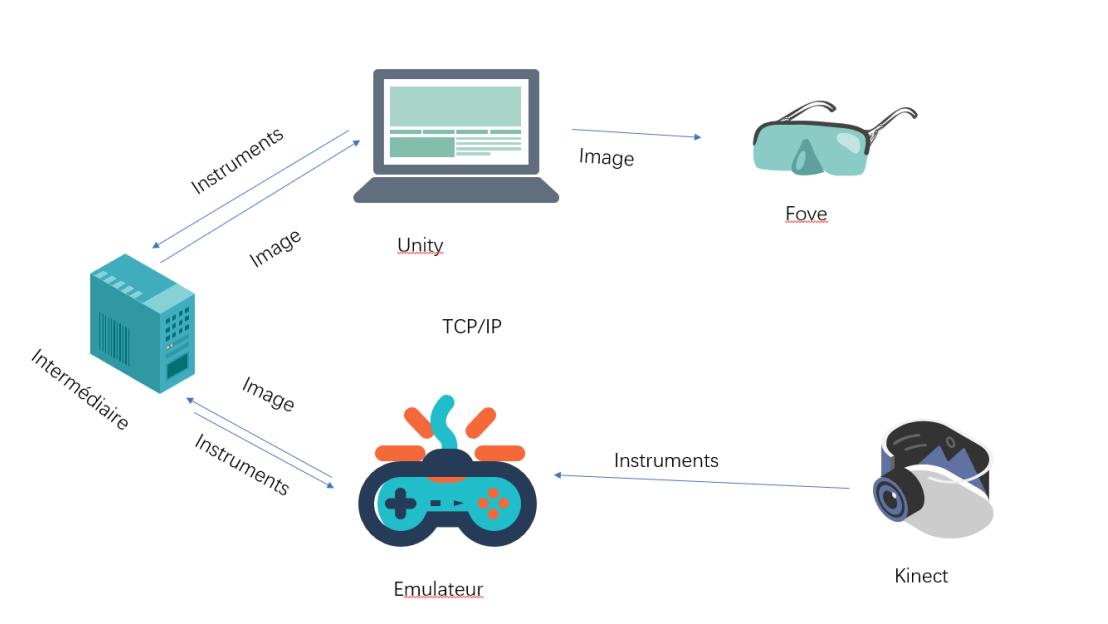


Présentation du travail effectué

8. Structure de système

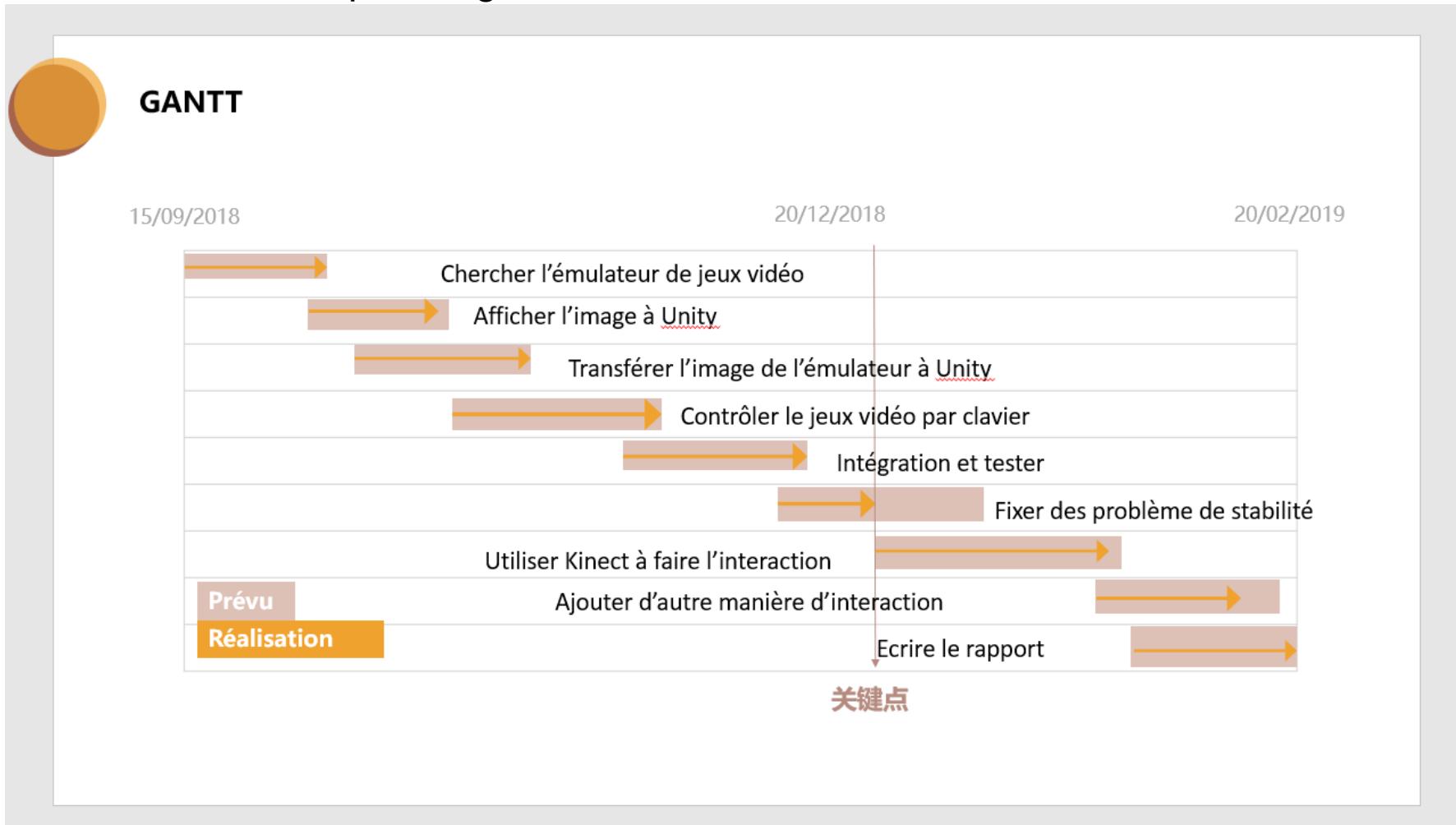
Intermédiaire est un le serveur de TCP/IP

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



Présentation du travail restant et du planning

4. Le GANTT



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?**