



CFPT INFORMATIQUE

ATELIER TECHNICIENS 1&2 - RAPPORT

ThermiScan

Étudiants :

Lorenzo Bauduccio

Tom Ryser

Kevin Moreno

Enseignant :

Francisco Garcia

4 mars 2020

Table des matières

1	Cahier des charges	3
1.1	Description du projet	3
1.2	Travail à réaliser	3
1.3	Outils utilisés	3
1.4	Comment avons-nous procédé ?	3
2	Réalisations	4
2.1	Application de sélection des jeux	4
2.1.1	Ajout de jeux	4
2.1.2	Lancement automatique	4
2.1.3	Login automatique à une session	5
2.1.4	Description du code	6
2.2	Habillage	8
2.2.1	Impression	8
2.3	Tablettes de commande	9
2.3.1	Configuration des touches	11
3	Conclusion	11
3.1	Bilan	11
3.2	Problèmes rencontrés	11
3.2.1	Téléphone ne permettant pas la diffusion en direct	11
3.2.2	Affichage d'un graphique	12
	Table des figures	13

1 Cahier des charges

1.1 Description du projet

Nous devons réaliser une application en Web qui doit nous permettre d'afficher une vidéo venant du smartphone CAT S60, qui possède une caméra thermique. L'application Web doit permettre de gérer plusieurs vidéos venant de plusieurs caméras, et permettre d'afficher une courbe de température.

1.2 Travail à réaliser

- Réaliser une application en Web qui s'approche un maximum d'un point de vue fonctionnel de l'application FLIR Tools Mobile.
- Affichage d'un graphique montrant la température maximal, minimal et la moyenne.
- Affichage de la vidéo.
- Gestion de plusieurs vidéos venant d'une seule caméra.

1.3 Outils utilisés

- Visual Studio Code
- EasyPHP V 14.1
- MySQL
- Python 3
- Balsamiq Mockups 3

1.4 Comment avons-nous procédé ?

- Réalisation d'une maquette du front-end
- Création d'un exécutable à partir d'un programme python
- Création d'une base de données pour les utilisateurs, les caméras et les vidéos
- Réalisation du front-end avec easyPHP
- Ajout d'un script JS permettant d'afficher un graphique
- Mise au propre du code

2 Réalisations

2.1 Application de sélection des jeux

L'application permettant de sélectionner les jeux à été réalisé avec Unity 3D, nous avons utilisé l'asset gratuite *SliderMenu* que nous avons un peu modifié pour quel corresponde à nos exigences.

2.1.1 Ajout de jeux

Pour ajouter un jeu sur l'application, il suffit d'ajouter un dossier nommé par le nom du jeu et contenant les dossiers respectant la hiérarchie suivante :

- App
 - *Contient les fichiers du jeu*
 - **game.exe** (L'exécutable du jeu doit être nommer comme-ça)
- image.jpg

2.1.2 Lancement automatique

Pour pouvoir lancer automatiquement l'application de sélection de jeux, il suffit de placer un raccourci de l'exécutable de l'application dans le dossier de démarrage automatique d'application accessible en effectuant la commande `shell:startup` sur la boîte de dialogue **Exécuter** (Win + R).

2.1.3 Login automatique à une session

Tout d'abord, il faut se rendre sur la boîte de dialogue **Exécuter** (Win + R) et entrer la commande suivante : `control userpasswords2`

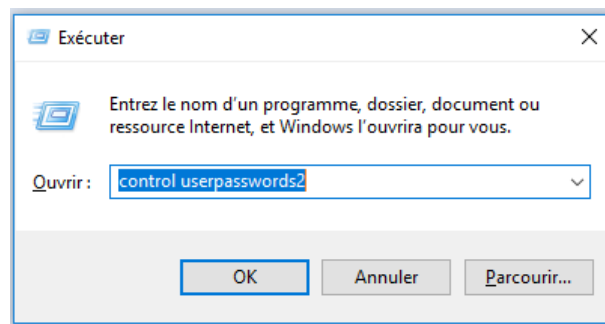


FIGURE 1 – Commande d'accès aux contrôle des mots de passes

Ensuite, il faut cliquer sur l'utilisateur que l'on veut déconnecter automatiquement et décocher la case *Les utilisateurs doivent entrer un nom d'utilisateur et un mot de passe pour utiliser cet ordinateur.*

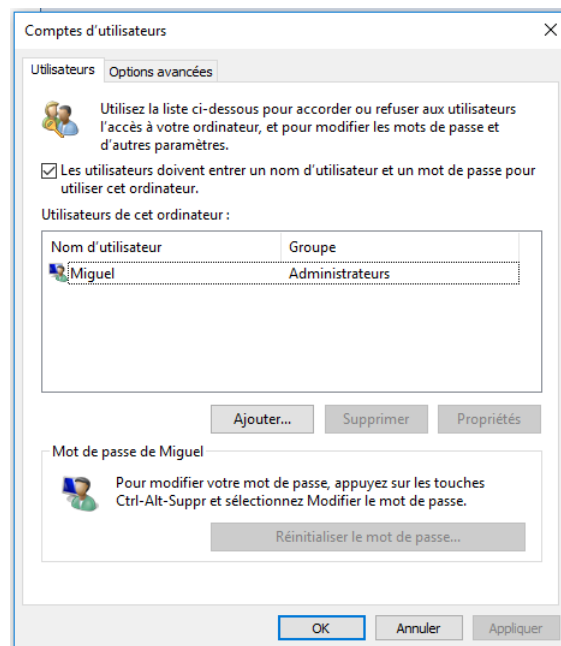


FIGURE 2 – Dés-activation du mot de passe pour un compte utilisateur

Il suffira ensuite de confirmer le retrait du mot de passe pour le compte pour lui permettre de se connecter automatiquement.

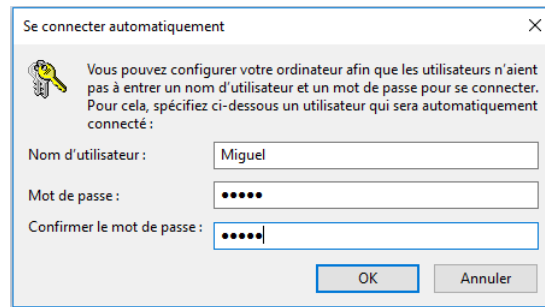


FIGURE 3 – Confirmation du passage en mode connexion automatique

2.1.4 Description du code

Le code gérant les jeux se trouve dans `GameLoader.cs`, d'abord la fonction `Start` (appelé lors de l'initialisation de la classe). Elle initialise le dictionnaire de jeux, récupère la liste de slides se trouvant sur la classe `SliderMenu` se trouvant sur la caméra principale, on désactive la visibilité du curseur et on charge la liste des jeux.

```

1 void Start()
2 {
3     this.Games = new Dictionary<string, string>();
4
5     GameObject camera = GameObject.Find("Main Camera");
6     SliderMenu script = camera.GetComponent<SliderMenu>();
7     Slides = script.Slides = new List<GameObject>();
8
9     Cursor.visible = false;
10
11     LoadGames();
12 }

```

La fonction `Update` est appelée à chaque *frame*, dans cette fonction, on vérifie les boutons appuyés par les joueurs et on effectue une action appropriée pour chaque entrée clavier. Lors de l'utilisation des joysticks vers la gauche ou la droite on invoque les cliques des boutons permettant de changer de slide. Lors de l'appui sur le bouton *8 (Dollars)*, on coupe le volume de la musique du menu puis récupérons le nom du jeu puis un *Process* va lancer l'exécutable du jeu sélectionné.

```

1 void Update()
2 {
3     if (Input.GetKeyDown(KeyCode.LeftArrow) || ←
4         Input.GetKeyDown(KeyCode.A))
5     {
6         btnLeft.onClick.Invoke();
7     }
8
9     if (Input.GetKeyDown(KeyCode.RightArrow) || ←
10        Input.GetKeyDown(KeyCode.D))

```

```
9      {
10          btnRight.onClick.Invoke();
11      }
12
13      if (Input.GetKeyDown(KeyCode.Alpha8))
14      {
15          audioSource.volume = 0f;
16
17          string selectedGame = ←
18              slides.transform.GetChild(slides.transform.childCount ←
19                  - 1).name;
20
21          UnityEngine.Debug.Log(audioSource.volume);
22
23          string path = @"\" + Games[selectedGame] + ←
24              "\\App\\game.exe";
25
26          Process gameApp = new Process();
27          gameApp.StartInfo.FileName = path;
28          gameApp.Start();
29      }
30 }
```

La fonction `OnApplicationFocus` est appelée lorsque le menu à de nouveau le *focus*, alors on remet le volume sonore de la musique au maximum.

```
1 void OnApplicationFocus(bool hasFocus)
2 {
3     if (hasFocus)
4         audioSource.volume = 100f;
5 }
```

La fonction `OnApplicationFocus` va charger la liste des jeux, créer les objets nécessaire pour être ajouté à l'objet *Slider* avec toutes les données nécessaire : nom, texte, image.

```
1 private void LoadGames()
2 {
3     DirectoryInfo dir = new DirectoryInfo(GAME_DIRECTORY);
4
5     foreach (DirectoryInfo d in dir.GetDirectories())
6     {
7         this.Games.Add(d.Name, d.FullName);
8     }
9
10    foreach (var game in this.Games)
11    {
12
13        GameObject gmObj = Instantiate(gameExample, ←
14            transform.position, transform.rotation);
15        Text gmObjText = gmObj.GetComponentInChildren<Text>();
16        Image gmObjImage = gmObj.GetComponentInChildren<Image>();
```

```
17 //Load values
18 gmObj.transform.SetParent(slides.transform);
19 gmObj.name = game.Key;
20 gmObjText.text = game.Key;
21
22 //Load image
23 byte[] data = File.ReadAllBytes(game.Value + ↵
24     "\\image.jpg");
25 Texture2D texture = new Texture2D(64, 64, ↵
26     TextureFormat.ARGB32, false);
27 texture.LoadImage(data);
28 gmObjImage.sprite = Sprite.Create(texture, new Rect(0.0f, ↵
29     0.0f, texture.width, texture.height), new ↵
30     Vector2(0.5f, 0.5f), 100f);
31
32 Slides.Add(gmObj);
33 }
```

3 Conclusion

3.1 Bilan

Dans l'ensemble, nous sommes plutôt satisfaits de notre projet. Nous avons atteint les buts les plus importants.

3.2 Problèmes rencontrés

3.2.1 Téléphone ne permettant pas la diffusion en direct

Le problème majeur rencontré dès le début du projet est l'indisponibilité de live streaming du Cat S60. Selon les pages qu'on parcourait, il était mentionné que la Cat S60 possédait une option de live streaming, ce qui n'est au final pas le cas, contrairement à la nouvelle génération du Cat, le S61. Étant donné que le cahier des charges était de faire un live directement via une application web, nous nous sommes retrouvé à devoir modifier le cahier des charges, qui est passé d'un live sur une application web à juste afficher une vidéo.

3.2.2 Affichage d'un graphique

Plusieurs problèmes se sont joint lors de la création du graphique. Premièrement, il nous fallait trouver un script JS permettant d'afficher un graphique à partir de valeur venant d'un fichier CSV. Lors de nos premiers tests, le problème principal était l'importation de nos valeurs dans les tableaux, les graphiques ne possédant pas de tableau souple au niveaux des données entrées. Nous avons essayé avec un graphique

venant de `www.amcharts.com` et un autre venant de `www.highcharts.com`. En fin de compte, Nous nous sommes occupé d'en faire un, en reprenant des parties de code provenant des anciens tests.

Table des figures

1	Commande d'accès aux contrôle des mots de passes	5
2	Dés-activation du mot de passe pour un compte utilisateur	5
3	Confirmation du passage en mode connexion automatique	6
4	Interface SmallCab P.1	9
5	Interface SmallCab P.2	10
6	Dimension face avant	11