

Université de la Manouba
Ecole Nationale Des Sciences De L'informatique



Rapport du Projet de Programmation

Sujet

jeux sérieux:
* escape-room *

Réalisé par

Aymen Fakhfakh

Ines Fadhli

Skander Abid

Encadré par notre professeur :

Yemna Sayeb

Année Universitaire 2021/2022

Introduction	3
Motivation	3
Problématique	3
Objectif	3
Etude Préalable	3
Les jeux sérieux	3
Jeu choisi: Historique et variantes	4
Concept et historique	4
Variantes	4
Jeu Proposé	4
Décortication du projet en objets	5
Les classes généraux :	5
Les Classes relatifs à chaque scène :	7
Menu	7
Room1	8
Room2	9
Room3	12
Room4	13
Réalisation	15
Langages et technologies	15
Chronogramme	16
Affectation des tâches	16
Conclusion	17

1. Introduction

1.1. Motivation

Comme étant le seul projet au cours de cette année et le premier dans notre parcours éducatif , notre groupe a fourni un effort considérable et une énergie énorme lors de sa réalisation. Ainsi, l'originalité du sujet de ce projet (les jeux sérieux) nous a impressionné et encouragé pour aboutir à la fin , à un jeu sérieux pas comme les autres .

1.2. Problématique

Au cours de notre formation à l'ENSI et pendant le deuxième semestre de première année , on nous a proposé un projet de programmation. Le sujet du projet pendant cette année parle de jeux sérieux . on est alors chargé de créer un jeu sérieux bien fonctionnel. **C'est quoi alors un jeu sérieux ? Quels sont réellement les objectifs de ce projet ? Quelles sont les compétences techniques à savoir pour le créer ? Quelle est la valeur ajoutée de notre jeu en particulier ?**

1.3. Objectif

L'objectif de ce projet est essentiellement d'avoir une amélioration sur plusieurs niveaux.

-Compétences techniques : ce projet offre l'opportunité de découvrir d'autres langages de programmation. comme pour notre cas , on a découvert le C# et un moteur de jeu multiplateforme qui est UNITY .On a alors acquis une nouvelle compétence technique dont on a absolument besoin dans l'avenir. De plus, le projet permet d'appliquer nos connaissances et ce qu'on a appris au cours des séances de web et de programmation .C'est le cas de ceux qui ont utilisé HTML , CSS, C++... pour la création de leur jeu .

-Côté intellectuel : le concept des jeux sérieux est nouveau pour plusieurs étudiants . Donc le fait de l'avoir comme projet va imposer de faire des recherches à son propos. Pour notre cas, cette recherche a été très enrichissante, elle nous a permis d'avoir de nouvelles informations à propos du domaine de jeux en générale , l'utilité des jeux sérieux , son impact ,son historique et sa place dans nos jours .

-soft skills : le fait d'avoir un travail en équipe améliore les capacité de communication, de défendre une idée, faire une critique constructive, partager des opinions et d'accepter ceux des autres , sont améliorés et bien développés .

2. Etude Préalable

2.1. Les jeux sérieux

un jeu sérieux est un jeu plus une fonction utilitaire.
les fonctions utilitaires se divise en 3 :

-diffuser un message : qui peut être message éducatif , informatif , publicitaire

-prodiguer un entraînement: le but de jouer est de renforcer les capacités physique du joueur

-favoriser l'échange de données

Les jeux sérieux apparaissent dans les années 1970 aux États-Unis. Il prend son essor au milieu des années 1990. La popularité montante des consoles de jeux sérieux , contribue à faire évoluer les mentalités et à montrer que l'on peut faire passer des connaissances par le biais d'un jeu.

2.2. Jeu choisi: Historique et variantes

2.2.1. Concept et historique

Dans notre projet on a choisi l'escape room comme idée. Son concept est très simple : un joueur ou un groupe de joueurs doivent résoudre des énigmes afin de sortir d'une salle et de passer à la suivante jusqu'à terminer toutes les salles. Ces derniers représentent des niveaux qui sont de plus en plus difficiles.

Le premier jeu vidéo d'escape game, CRIMSON ROOM, a vu le jour en 2004 et a été créé par Toshimitsu Takagi.

2.2.2. Variantes

En 2007 , la société japonaise SCRAP a voulu faire évoluer le concept d'escape room en créant les 1ères live escape game. L'histoire d'escape room s'est par la suite développée dans le reste de l'Asie avant d'arriver en Europe.

Mi 2021 , nous comptons près de 20500 salles dans plus de 870 enseignes à travers la France , dont environ 200 salles à Paris. ce type de jeu n'est plus accessible en ligne seulement mais il est mis en scène.

3. Jeu Proposé

Dans le cadre des jeux sérieux , nous proposons un escape game revisité. En effet , on a ajouté un côté intellectuel pour l'escape game. Pour passer d'une salle à une autre , il faut trouver des indices cachés dans cette salle . répondre correctement aux questions proposées par cet indice pour passer d'une salle à une autre. Ainsi, les règles sont les suivantes

Règles:

1. chercher la fiche contenant les informations, cette fiche est cachée dans l'un des objets de la salle. En cliquant sur l'objet correct, cette fiche s'ouvre.

2. lire attentivement la fiche et essayer de bien comprendre les informations données

3. continuer à chercher les indices cachés qui sont maintenant des questions à propos la fiche d'information trouvée en premier lieu

4. essayer de répondre correctement aux questions pour avoir le score suffisant au passage à la salle suivante.

4. Décortication du projet en objets

4.1. Les classes généraux :

```
public class ouvrir_fermer : MonoBehaviour
```

```
{  
    [SerializeField] private GameObject popup_1 ;  
    [SerializeField] private GameObject popup_2 ;  
    public void ouvrir()  
    {  
        popup_1.SetActive(true);  
        popup_2.SetActive(false);  
    }  
}
```

```
public class ouvrir : MonoBehaviour
```

```
{  
    [SerializeField] private GameObject popup ;  
    public void ouvrir()  
    {  
        popup.SetActive(true);  
    }  
}
```

```
public class retour : MonoBehaviour
```

```
{  
    public void retourner()  
    {  
        SceneManager.LoadScene(0);  
    }  
}
```

```
public class score : MonoBehaviour
```

```
{  
    [SerializeField] private GameObject popup ;  
    [SerializeField] Text codeText;  
    private Score s;  
    private string path;
```

```

private bool test = true;

void Start(){
    path = Application.dataPath + "/Data/gamedata.json";
    if (SceneManager.GetActiveScene().buildIndex == 1)
    {
        s = new Score();
        File.WriteAllText(path, JsonUtility.ToJson(s) );
    }else{
        s = JsonUtility.FromJson<Score>(File.ReadAllText (path));
    }
}

void Update ()
{
    if (popup.activeInHierarchy == true && test)
    {
        s.Upd() ;
        string json = JsonUtility.ToJson(s);
        File.WriteAllText(path, json );
        test = false;
    }
    else if (popup.activeInHierarchy == false) test = true;
    codeText.text = s.Get().ToString();
}

[System.Serializable]
class Score{
    public int val;
    public Score() { val = 100; }
    public void Upd() { val -= 10; }
    public int Get() {return val; }
}
}

```

4.2. Les Classes relatifs à chaque scène :

4.2.1. Menu



```
public class sceneMenu : MonoBehaviour
```

```
{  
    public void Play()  
    {  
        SceneManager.LoadScene(1);  
    }  
    public void how_to_play()  
    {  
        SceneManager.LoadScene(5);  
    }  
    public void Quit()  
    {  
        Debug.Log("Quit!");  
        Application.Quit();  
    }  
}
```

4.2.2. Room1



public class passer_room_2 : MonoBehaviour

```
{
    [SerializeField] Text codeText;
    string codeTextValue = "";
    void Update ()
    {
        codeText.text = codeTextValue;
        if (codeTextValue == "7531")
        {
            SceneManager.LoadScene(2);
        }
        if (codeTextValue.Length >= 4)
        {
            codeTextValue = "";
        }
    }
    public void AddDigit(string digit)
    {
        codeTextValue += digit;
    }
}
```



```
}
}
```

4.2.3. Room2



```
public class code : MonoBehaviour
```

```
{
    [SerializeField] Text codeText;
    [SerializeField] private GameObject popup_1 ;
    [SerializeField] private GameObject popup_2 ;
    string codeTextValue = "";
    void Update ()
    {
        codeText.text = codeTextValue;
        if (codeTextValue == "@*#")
        {
            popup_1.SetActive(true);
            popup_2.SetActive(false);
        }
        if (codeTextValue.Length >= 3)
        {
            codeTextValue = "";
        }
    }
}
```

```

    }
}
public void AddDigit(string digit)
{
    codeTextValue += digit;
}
}

public class codepanel : MonoBehaviour
{
    [SerializeField] Text codeText;
    [SerializeField] private GameObject popup_1 ;
    [SerializeField] private GameObject popup_2 ;
    string codeTextValue = "";
    void Update ()
    {
        codeText.text = codeTextValue;
        if (codeTextValue == "4682")
        {
            popup_1.SetActive(true);
            popup_2.SetActive(false);
        }
        if (codeTextValue.Length >= 4)
        {
            codeTextValue = "";
        }
    }
    public void AddDigit(string digit)
    {
        codeTextValue += digit;
    }
}

public class passer_room_3 : MonoBehaviour
{

```

```

[SerializeField] Text codeText;
string codeTextValue = "";
void Update ()
{
    codeText.text = codeTextValue;

    if (codeTextValue == "2468")
    {
        SceneManager.LoadScene(3);
    }
    if (codeTextValue.Length >= 4)
    {
        codeTextValue = "";
    }
}
public void AddDigit(string digit)
{
    codeTextValue += digit;
}
}

```

4.2.4. Room3



```
public class passer_room_4 : MonoBehaviour
```

```
{
```

```
    [SerializeField] Text codeText;
```

```
    string codeTextValue = "";
```

```
    void Update ()
```

```
    {
```

```
        codeText.text = codeTextValue;
```

```
        if (codeTextValue == "4589")
```

```
        {
```

```
            SceneManager.LoadScene(4);
```

```
        }
```

```
        if (codeTextValue.Length >= 4)
```

```
        {
```

```
            codeTextValue = "";
```

```
        }
```

```
    }
```

```
    public void AddDigit(string digit)
```

```
    {
```

```
        codeTextValue += digit;
```

```
}  
}
```

4.2.5. Room4



```
public class code_room_4 : MonoBehaviour
```

```
{  
    [SerializeField] Text codeText;  
    [SerializeField] private GameObject popup_1 ;  
    [SerializeField] private GameObject popup_2 ;  
    string codeTextValue = "";  
    void Update ()  
    {  
        codeText.text = codeTextValue;  
        if (codeTextValue == "9718")  
        {  
            popup_1.SetActive(true);  
            popup_2.SetActive(false);  
        }  
        if (codeTextValue.Length >= 4)  
        {  
            codeTextValue = "";  
        }  
    }  
}
```

```

    }
}
public void AddDigit(string digit)
{
    codeTextValue += digit;
}
}
public class win : MonoBehaviour
{
    [SerializeField] Text codeText;
    string codeTextValue = "";
    void Update ()
    {
        codeText.text = codeTextValue;

        if (codeTextValue == "4747")
        {
            SceneManager.LoadScene(6);
        }
        if (codeTextValue.Length >= 4)
        {
            codeTextValue = "";
        }
    }
    public void AddDigit(string digit)
    {
        codeTextValue += digit;
    }
}

```

5. Réalisation

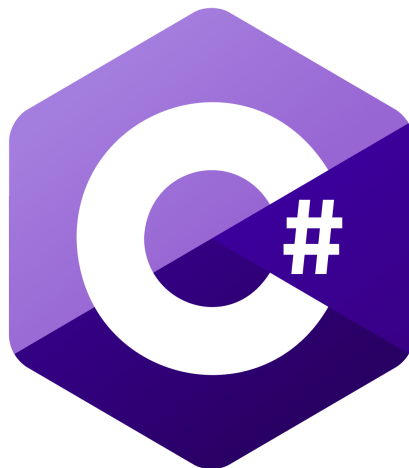
5.1. Langages et technologies

Unity est un moteur de jeu multiplateforme (smartphone, ordinateur, consoles de jeux vidéo et Web) développé par Unity Technologies. Il est l'un des plus répandus dans l'industrie du jeu vidéo, aussi bien pour les grands studios que pour les indépendants du fait de sa rapidité aux prototypages et qu'il permet de sortir les jeux sur tous les supports.

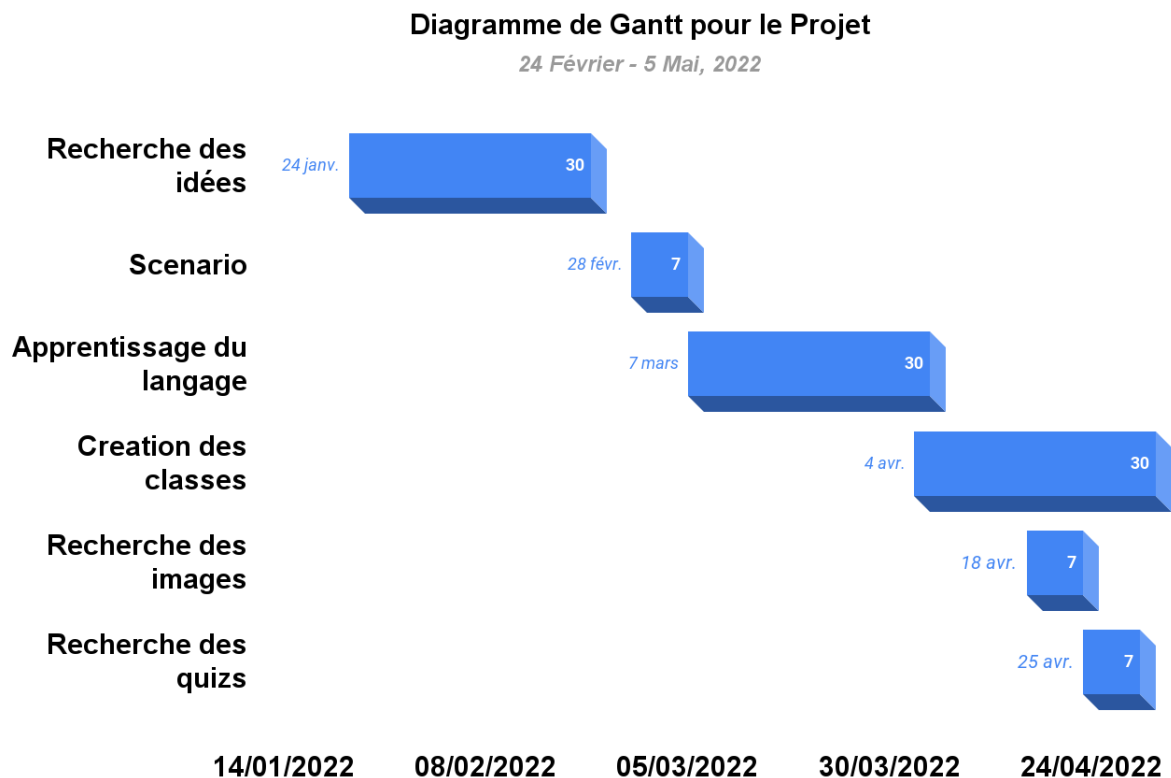


Le logiciel a la particularité d'utiliser du code (C#) sur la plateforme « .NET » avec l'implémentation Mono.

C# est un langage de programmation orientée objet, fortement typé, dérivé de C et de C++. Il est utilisé pour développer des applications web, ainsi que des applications de bureau, des services web, des commandes, des widgets ou des bibliothèques de classes. En C#, une application est un lot de classes où une des classes comporte une méthode *Main*.



5.2. Chronogramme



5.3. Affectation des tâches

Etapes	Répartition des tâches
Scénario	Travail collectif
Recherche des images	Travail collectif
classes	Chacun a pris une partie du code puis nous avons fait des réunions pour que chacun parle de sa classe et voir les problèmes de compilation et les messages d'erreur.
Recherche des quizzes	Chacun s'est intéressé au quiz d'une chambre .
Rapport	Travail collectif

6. Conclusion

Avant l'achèvement de notre projet, nos connaissances techniques étaient relativement insuffisantes. De ce fait, nous avons eu recours, en premier lieu, à un nombre important de cours et de tutoriels à propos du logiciel Unity et de la programmation en C#.

Nous nous sommes retrouvés devant l'obligation d'apprendre des nouvelles techniques de développement pour confronter les difficultés rencontrées.

Ce projet a amélioré notre esprit de partage et du travail en équipe et nous a motivé à travailler dans le domaine des jeux vidéo.

On a pu améliorer le jeu en ajoutant la possibilité de faire un login pour que le joueur puisse voir ses résultats précédents

Aussi on peut améliorer l'expérience du joueur en mettant le jeu en plein écran mais cela nécessite du temps et des connaissances que nous n'avons pas.