**PFE Speech**

**[Slide 1]** Hello and welcome, first, we would like to thank you for being here today for our final project presentation titled "Conception and Development of a 3D Multiplayer Online Mobile Game." This project was supervised by Ms. Salima Mnif. Thank you madame.

**[Slide 2]** To present the work we have done; we have adopted this plan:

* **First**, we will begin with an introduction, where we will provide an overview of the company and the problematic and our proposed solution to give you some background information.
* **next**, we will delight you with the development methodology,
* **Afterwards**, we will move on to specifying the requirements and designing the solution
* **Then** in the next phase, we will focus on the design and conception.
* In **the 5th** part we move to the realization stage
* **Finally**, we will conclude the presentation with a summary and discuss the prospects of this project.

**[Slide 3]** Lets begin with the introduction and problematic.

**[Slide 4]** We had our internship within CGI Studio company.

CGI Studio is a professional IT office founded in 2018 by its CEO’s Amen Allah Ben Achour, he works mainly in developing PC and mobile video games.

**[Slide 5]** The global game market industry is rapidly growing, meanwhile in Tunisia the game market is slowly increasing.

As developers, how can we keep up with the global game market?

**[Slide 6]** Our solution revolves around designing, developing, and producing a video game tailored for the Tunisian market. However, our goal is to expand its reach to the global market.

**[Slide 7]** It is an RPG 3D online game in a third-person perspective, with the theme of low-poly worlds, battel royale,

where 4 players get spawned on a map and battle AI generated enemy bots and level up in game to get more stronger and powerful

As the game ends after 3 min countdown, the last-standing player or the player with most kills will be announced to be the winner of the match.

We have chosen the name “Let’s Hunt” for our video game.

**[Slide 8]** Moving on to the Development methodology.

**[Slide 9]** To facilitate the elaboration of this work we used the scrum methodology as it will help us set our goals and achieve them in a planned way by dividing the project into 4 different sprints.

**[Slide 10]** After detailing the functional requirements we divided our project into 4 sprints, each sprint must be done within 4 weeks except the third sprint 7 weeks.

And it will be planned as following:

The first sprint considers character and maps creation.

The second is entitled « user interface design » and it considers designing game interfaces such as main menu and in-game interface.

In the third sprint we focus on animating the characters and enemies and all game controls.

And finally, the fourth sprint is dedicated to implementing the authentication system, game enhancement and deployment.

**[Slide 11]** In this part we are going to illustrate the requirements of the application.

**[Slide 12]** Our application offers these functional requirements as described here.

It offers the possibility to: ….

**[Slide 13]** Moving to Non-Functional requirements we ensure that our game is performant, scalable with friendly interface that is easy to interact with, and of course reliable.

**--Abouda**

**[Slide 14]** Now we move on to the design section where we present how the system is designed in each sprint.

**[Slide 15]** Before digging deeper into conception and diagrams we need to talk about the first sprint “3D Modelling” since it doesn’t require a detailed conception.

We used Blender software to create hunters and map objects for our game.

**[Slide 16]** to be done, The **modelling** phase took us about a month since it went through different stages

To model a character we need to :

* **Create the base me**sh which will look like a humanoid shape but still looks like clay.
* **Once the base mesh me is done** we move on to the UV mapping which means **unwrapping** the mesh to look like it’s printed on a 2D surface .
* **And then** we move to the **texturing** stage, in which we select the faces that we want to texture and we apply the desired color
* **Moving to** the rigging part which means adding a skeleton to the character so we can apply animations and moves using Mixamo .
* **Finally** after we rigged the character it looked like this.

**[Slide 17]** We created our maps by arranging 3D objects around a terrain.

We created 2 different maps nature and city , by those 3D objects and we utilized some others from the unity asset store to gain more time for the programming phase

And those are our final maps.

**[Slide 18]** the global use case diagram represents the main actor which is the "player", and how he interacts with each use case.

The player accesses the main menu after being authenticated.

This menu interface grants him the options to check updates notifications of the day or leaderboard, adjust the settings, access the hunters panel, visit the shop panel, view the profile panel.

Until he decides to play and start match so he can choose the game mode and wait for other players to play where he have the possibility to …..

**[Slide 19]** Moving to the global class diagram.

**[Slide 20]** Our class diagram is composed of 22 classes, this figure represents these elements and the relationships between them.

We will try to briefly explain this diagram.

* The **SaveManager** class handles saving and loading game data.
* The **ConnectToServer** script is responsible for connecting to the Photon networking server.
* The **HunterSelection** script handles selection and all hunter information.
* **The Player** class which will be explained further, inherits all the associations, relationships, and attributes of the Hunter game object.
* A Player can attack an **Enemy**, and the opposite can also happen.
* **After** defeating an enemy, the player can collect either a Coin or XP which displayed by this association.
* **Each Player** is followed by a camera.
* **Finally**, The Player, Enemy, Terrain, and Canvas game objects are weakly aggregated into the "Map" scene .

**[Slide 21]** In the second sprint we focused on designing the user interface of navigating in the main menu use cases.

**[Slide 22]** In the third we focused on programming all in game controls ….

--**Abouda**

**[Slide 23]** The "HunterInGame" which we utilize to optimize the “Player” class in global class diagram, is aggregated by seven scripts.

Explaining each script as follow :

* **The hunter** is composed of **Rigidbody** , **CapsuleCollider**, **Animator**, and **Transform** which are unity components responsible for bringing the character to life in game.
* The "**PlayerMove**" script is responsible for moving the hunter and it is controlled by **Joystick.**
* The **PlayerAttack** script handles the attack functionality.
* The **PlayerHealth** script is responsible for managing the health of the hunter.
* The **LevelUp** script allows the player to level up during the match.
* The **PlayerScore** script is responsible for counting the score and ranking of the player in-game.

We hope we this made clear and now we move to the enemy class

**[Slide 24]** To not end up with a crowded global class diagram, we also optimized the enemy class.   
Where we also attached to it the **PhotonView** script to sync the animations and controls across all players.

And the needed unity components to control the AI enemies.

* **EnemyHealth** handles managing health, taking damage, and spawning Coins and XP after being defeated.
* **EnemyControl** script is responsible for moving and animating the enemy character according to its state.

**[Slide 25]** Due to a delay in the game's development, we were unable to deploy it on the Google Play Store. Because it’s still under review for more of 20 days

As an alternative, we have created an official website for the game using Angular so the users can download the game.

Since we can’t have access to firebase and google play services, as an alternative we implemented PlayFab into our game. To add authentication and data storing systems.

**[Slide 26]** In this part we will talk about the realization of our game.

**[Slide 27]** To elaborate this work, we used these software development tools:

- **Unity3D** as game engine

- **Blender** to create our characters and scene objects

- **Visual** **studio** code as a code editor

- **Photoshop** for designing the user interface like buttons, panels ….

- **Mixamo** rig our game characters

**[Slide 28]** And these programming languages and frameworks

C# and Angular which you already familiar with. And PlayFab and Photon Network that we will explain in the next Slides.

**-- Souhail**

**[Slide 29]** let's dive into implementing photon for multiplayer functionality in our project

**1 /** Photon is a real-time multiplayer networking framework specifically designed for games.

**2 /** The process began by importing the Photon package and setting up the project accordingly

**3 /** Allowing players to connect with each other in a shared virtual space. Either by creating private room for friends and share the Unique ID or joining public rooms for a competitive challenge.

**4 /** Synchronizing animations across all players is another significant aspect of multiplayer experience. By attaching the PhotonView component to relevant game objects, we was able to ensure that animations were replicated accurately across all connected devices.

**5 /** Communication is vital in multiplayer games, so we integrated an in-game chat system using Photon. Players can now interact and communicate with each other in real-time.

After we arranged to add the multiplayer functionality in our game, we focused on implementing a backend platform to manage players accounts and data storing

**[Slide 30]** Moving on to the PlayFab implementation

**1 /** PlayFab is a backend platform for games development, which provides authentication, data storage, game currency management, and more.

**2 /** We integrated PlayFab's authentication system into our game, enabling players to register, login, and securely access their accounts.

**3 /** To enhance account security and user convenience, we implemented PlayFab's password resetting feature By consuming PlayFab's API within Angular.

**4 /** to manage player data, we utilized playfab’s data storing functionalities. This includeds player progress, achievements, and other relevant information.

**5 /** We also utilized PlayFab's news management feature to keep the players engaged even when they are not playing by adding news updates that came in the format of notifications or within the newsletter.

**[Slide 31]** This is an example of the game interfaces where the user navigates in the main menu and playing.

The rest of the interfaces will be displayed during the demonstration.

**[Slide 32]** And this is an example of the website where the user can download the game.

The rest of the interfaces will be displayed during the demonstration. **--Souhail**

**[Slide 33]** We end this presentation with conclusion and perspectives.

**[Slide 34]** In conclusion, we are incredibly proud to have created this game using all these technologies for the first time.

This journey has been an exciting experience, pushing us to explore new technologies and dive into the world of game development.

Creating a game from scratch and exploiting the potential of these tools was no small feat.

It required learning new concepts, overcoming challenges, and dedicating countless hours of hard work.

**[Slide 35]** Now that we finalized our project, we hope that it met your expectations.

We aim to work more on our game with CGI Studio company and to succeed in expanding it to the global market after it is published in Play Store and enhance it more.

**[Slide 36]** Thank you for your attention and let's move on to the exciting part!

we would like to showcase a live demonstration of Lets Hunt.