**Project Outline**

1. **Project Name: LIVE-ON**

* Project Type: Android Game
* Game Type: 2D Horror game

1. **Build Tool:**

* **Main develop tool:** Unity3D engine
* **APK building tool:** Android studio

1. **Game Design:**

* **World Overview:** In the year 2024, the game unfolds in the troubled region of Grania. Grania has become a battleground due to a localized war. The conflict escalated, and intense urban warfare became a daily occurrence, affecting the lives of its residents. Many civilians fled the area to escape the turmoil, but the neighboring countries, in an attempt to prevent further escalation of the conflict and an influx of refugees, decided to seal off the borders, leaving the suffering people of Grania to their fate. The main character is a robotics engineer named Graciano, and his wife, Sando, is an artificial intelligence engineer who once led a promising AI project called "Sandoleathy". The two's peaceful life is broken by a group of unidentified intruders, Sando is taken away, and Graciano escapes. So he embarks on a quest to find his wife and gradually unearths the truth about her, the invaders, and the project Sandoleathy.
* **Core Gameplay:** Because this is a horror game, I came out an idea to make game more excited. It’s also the core game play of the game. Many functions and system will be built around it.
  + **Heartbeat System:** The heart rate system is a system that reflects a character's stress level, and it has the potential to replace systems such as strength and health, but since heart rate is not a direct reflection of a person's state, it will be used in the game in conjunction with the strength system, health system, medicine system and damage system. The main function of the heart rate system is to reflect the state of the character through the change of the heart rate, and create a sense of stimulation for the player, so that the player can maintain a tense state when the character is in a high heart rate, and increase the sense of immersion.
  + **Detail information：**There are two types of heart rate: state heart rate and additional heart rate
  + **State heart rate:** Determined by the player character's environment and physical state, the state heart rate is similar to the base.
  + **Additional heart rate:** Additional heart rate is an added value added to the state heart rate. When an additional heart rate condition is triggered, the heart rate will be increased or decreased in accordance with the corresponding requirements of the additional heart rate. Additional heart rate is also divided into two types: fatal heart rate and non-fatal heart rate
  + **Fatal heart rate:** This heart rate has no upper or lower limit, that is, as long as the condition of increasing/decreasing heart rate is triggered, the additional heart rate will be added or subtracted from the state heart rate. Once the heart rate is in the range that will cause the character to die due to this change, the character is likely to die.
  + **Non-fatal heart rate:** there is an upper and lower limit of non-fatal additional heart rate. When the heart rate meets the upper and lower limit conditions, the heart rate will be increased or decreased according to the corresponding requirements of the additional heart rate. At the same time, the heart rate is checked for the upper limit at intervals, so if the heart rate crosses this threshold before the next test, the heart rate does not increase after the next test and slowly drops to the upper limit.
  + The heart rate will start timing after it exceeds 220, and once the timing exceeds 15 seconds the character will die of cardiac arrest due to tachycardia.
  + The timing starts when the heart rate is below 45 and above 35, and if it continues for more than 15 seconds the character will enter a semi-coma due to bradycardia. If it continues for more than 10 seconds, the character will die of bradycardia. If your heart rate drops below 35, you die directly from bradycardia.
  + **Implementation Plan：**The increase and decrease of heart rate function consists of a queue of independent threads, from which a value is extracted every 2 seconds to update the heart rate. Each heart rate change is pushed into the queue by the corresponding heart rate change method, and the object contains the upper and lower limits of heart rate change and the changed value.
* **Work Plan(From Week7)**
  + **Week7-Week9:** Finish the development of the Scene 1, master the development environment of Unity3D, allocate all development resources (art resources, scripts, audio materials), and submit the game content of Scene 1 as Alpha submission.
  + **Week10-Week12:** Develop all the contents of Scenes 2 to 4, further master the development environment of Unity3D, and submit the first four scenes as the content of Beta submission.
  + **Week13-Week15:** Finish developing the final scene, test the game as a whole, and add the rest of the art and audio. Package the final version of the game as a final submission