Writeup Valorant System

Group Members: Bonny, Xuan, Yifei, Yuting

Description of your project:

A system based on the Valorant video game. Valorant is a tactical 5v5 first person shooter game where you play as an agent and each agent has their own unique 4 abilities. Our Valorant system will have all the main components of the game to simulate a round. Each player profile will have their own inventory and settings.

What would the final product look like?

- What functionality will it have?

  Be able to simulate a round of Valorant. Each player profile will be assigned to an agent. Each round will be played on a map with one of the five game modes. There will be five players on each team and each agent will be able to shoot each other and take damage. You will be able to play as an agent with 4 unique abilities, buy a primary, secondary weapon, buy skins to add to their inventory, change their game sensitivity, crosshair and pick a gamemode to play. Depending on your agent, your abilities may be able to deal damage, blind, decay or reveal the enemy.

  - Guns (Xuan)
  - Agents (Yifei)
  - Abilities (Bonny)
  - Player Profile (Yuting)
  - Inventory (Xuan)
  - Maps (Bonny)
  - Gamemodes (Yifei)
  - Settings (Yuting)
  - Shop

- What is the target demographic?

  - Anyone age 12 and over
  - Mostly teens and young adults
  - Valorant's colorful design, fun gameplay, and cool looking skins, overall really appealing graphics and user interface
  - Free to play and not pay to win

- Fairly easy to run compared to other fps games, can run on any working computer

o What are some similar products in the field? How might they be an inspiration? What would be different from their product?

Other FPS games like CS GO, Apex Legends, Overwatch etc

Differences:
Characters, abilities, shorter time to kill, different game modes, maps, map size, weapons.

Development Plans

o Which flavour(s) of agile will you use?

- Scrum

§ How will you demonstrate this?

- Break work into small chunks called sprints
- Each sprint should add functionality and work towards the final product
- Testing throughout
- Three sprints

o How will you ensure everyone has equal responsibility for their parts?

- Distribute workload and tasks equally to people in the group
- Regularly start meetings to check on each individual's progress
- Communicate outside of school on discord
- Make sure everyone agrees/checks work before proceeding

o How are your initial plans for the sprints?

- Three sprints
- First sprint we will make the first iteration of the UML diagram with relationships, attributes but without methods and will be able to print out an agent, abilities, weapons, and the player profile playing the agent. We will have all the classes created with their attributes. The agents will be able to use weapons, cast abilities and view player profiles.
- Second sprint we will continue to work on the UML diagram by adding the methods for each class.

        - Maps, gamemodes, settings, and inventory will be implemented with the second sprint and integrated with the classes made in the first sprint. For example, each player profile will be playing on a gamemode and map. Settings will include things like crosshair and gun skins will be taken from the inventory. At this sprint you should be able to play as an agent, use abilities, weapons, have skins, choose gamemode, change settings and play on a random map.
    - Last sprint
        - Create main methods for each class that shows the entire functionality of the class
        - Add Javadoc in one class for each group member and indicate which methods demonstrate the conditions required in the assignment
        - Main class demonstrating all of the classes
        - At this point everything should work and our Valorant System should be good to go.
        - We should be able to simulate a round of a Valorant just like in the game.
    - During every sprint we will be testing for bugs

o How are you going to test if your program is working?

- Have a github repository so everyone's code is up to date
- Check integrity of the code by testing at each sprint
- Run the code together and debug together