**Question-1: This is our main product and the landing page https://csgoempire.com/roulette**

**How would you approach the testing of roulette? Describe the tests that you would do and the things you'd pay attention to.**

**Add your solutions to a repo and share the link to it with us below**

**Answer:**

**Testing Approach:**

The purpose of Quality Assurance is to ensure that the software meets its quality standards and performs as expected. I would approach Roulette testing in the following ways.

1. Define the **scope of testing**- To identify what all areas to test and what not to test
2. **Analysis of Game math** and other dependencies.
3. Creation & analysis of **test plan** document.
4. Designing appropriate **test cases based on priority** and avoiding duplicate test cases.
5. **Test data preparation** for execution of test cases.
6. Analysing and executing the **market regulation** for each state that is required and setting up high priority fin executing the same.
7. **Analysing and allocating resources, time frame** based on available time for test completion.
8. Requesting the **necessary seeds and emulation procedure** from development.
9. **Analyse the risk** involved in testing the game.
10. Identify **areas of manual and automation testing** in order to reduce the effort involved during manual testing.
11. Selecting the **right tools for automation**.
12. Performing both **White box & Black box testing.**
13. Performing **Compatibility testing** to ensure that application works as expected across **different platforms** like windows, mac, etc. and also across **different browsers** like Chrome, Firefox, Edge, etc.
14. Performing **Functional and non-Functional testing** including performance testing, load testing, stress testing, auto play, etc.
15. **Analysing the code changes** with **every drop/ build** provided in order to identify any breakage/ changes in code.
16. Performing **Regression test** at the end or final build.
17. Analyse the application through **end user perspective**.
18. **Review test results** thoroughly and ensure that the product is ready for submission.

**Test to do:**

The following below are the basic test that I would do.

**Pre-condition:**

In order to place a bet, the player should already be a registered user with steam community and must sign in.

**a. General Scenarios:**

1. Check whether the link navigates to the appropriate URL and landing page of **CSGO empire-roulette** is displayed correctly
2. Verify that the following **disclaimer is displayed** to the player in the landing page “**For the month of January were hosting 10,000 coin daily roulette races. Scroll down to see more.”**
3. Verify that during the **month of January 2024** Roulette race is hosted for **total prize award of 10,000 coins.**
4. Verify that the player is able to **Sign In** using the Sign-In button provided.
5. Verify that the details are displayed in **default language- English**.
6. Verify that **Race Rules** are displayed to the player at the bottom of the page.
7. Verify that **Chat room** messages are viewable to the player.
8. Verify that in the chat room player is able **to select a language of his choice** from the list of languages displayed and that only the corresponding language chat messages are displayed when a particular language is chosen.
9. Verify that **player is not able to send/ type any message** in chat box **unless they have logged in.**
10. Verify that different options like **Roulette, Match betting, Coin Flip, Cases** are available and displayed to the player.
11. Verify that the **Daily Roulette race table** displays the **top 20 player names** and updates for each spin of the Roulette with the amount wagered by the player on a daily basis.
12. Verify that a **24-hour** **timer** runs in order to showcase to the player the amount of play time remaining for day.
13. Verify that the **Previous rolls** represent the symbolic representation of **last 10 roulette** drawn and gets updated with last won value after each spin.
14. Verify that the **Last 100** displays the count of each symbol won (2X , 14X (Dice) or 2X (Star)) summing it to 100.

**b. Functional Scenarios:**

I would pay attention to the following scenarios below.

**Pre-condition:**

In order to place a bet, the player should already be a registered user with steam community and must sign in.

1. Verify that the player is able to play roulette game only after **depositing cash** using the deposit button provided and buying some coins.
2. Verify that the player is able to **withdraw cash** using the withdraw button provided.
3. Verify that the **RNG seed used is random, transparent** and not repetitive.
4. Verify that the **secret seed** on which the result is based on **matches with the hashed format public seed** provided for that round/ rolls.
5. Verify that after every **15 seconds timer** the Roulette starts to spin and lands on a symbol randomly.
6. Verify that the player is able to **place a bet of his choice** by entering the corresponding value in the “**Enter Bet Amount**” text field.
7. Verify that **correct value of bet amount** is updated in the “Enter Bet amount” field when the player presses the **following buttons**-> +0.01, +0.1, +1, +10, +100, ½, \*2, MAX, etc.
8. Verify that the **Bet amount is cleared** when the player presses the **clear button**.
9. Verify that a **maximum wager of 100000 coins** is only allowed by the game under any circumstance.
10. Verify that the **maximum wager limit** of 100000 coins **can be increased upon request**.
11. Verify that the **Roulette starts to** **spin from Right to Left** once the 15 seconds timer elapses each time.
12. Verify that upon **placing a wager** only the **corresponding coin value is deducted** from the players coin balance and remaining amount is retained in the players bank/ wallet.
13. Verify that the **total bet amount** for the **winning** **symbol is** **multiplied** and applied to the correct Place bet value – (2X or 14X or 2X) respectively and each **players wager is multiplied** by the **correct place bet value** – (2X or 14X or 2X) based on the outcome of each game spin.
14. Verify that **history records** all the **last outcomes of each player correctly** and **the last 10 outcomes** are displayed in the previous rolls.
15. Verify that the **player is allowed to** **choose/ place a bet** of any kind (2X, 14X or 2X) at any time before the **15 seconds timer runs off/ elapses**.
16. Verify that the **player is not allowed to choose/ place a bet** of any kind (2X, 14X or 2X) at any point of time when the **Roulette starts to spin and until the spin is completed.**
17. Verify that the **Bet total is reset to 0** and each **player winning information of last played game is reset** for all the 3 place bet values (2X, 14X and 2X) before the start of next Roulette spin.
18. Verify that the Roulette **resets to its default position (Dice at the centre)** before the **start of next spin** timer initiation.
19. Verify that the player who has **wagered the most** is ranked first and followed by other players in descending order in the **Daily Roulette race table**.
20. Verify that players are awarded a **prize amount of 10000 coins** spitted among the **top 20 players** ranging from 3500 coins for the player leading the Daily Roulette table as the 1st position and X number of coins for 20th position as shown in the table below.

**Note:** When clicking on 2nd page in the daily roulette table it prompts to sign in hence unable to see prize award range for players occupying position 11-20. Hence, I have mentioned as X number of coins in point-8 and have listed the prize for only up to 10th position in the below table. The same needs to be validated for the top 20 wagers.

|  |  |  |  |
| --- | --- | --- | --- |
| **Position** | **Player Name** | **Wagered amount** | **Prize amount** |
| 1st | Player-1 | XXXXXXXXXX | 3500.00 coins |
| 2nd | Player-2 | XXXXXXXXX | 2500.00 coins |
| 3rd | Player-3 | XXXXXXXX | 1300.00 coins |
| 4th | Player-4 | XXXXXXX | 850.00 coins |
| 5th | Player-5 | XXXXXX | 500.00 coins |
| 6th | Player-6 | XXXXX | 300.00 coins |
| 7th | Player-7 | XXXX | 200.00 coins |
| 8th | Player-8 | XXX | 150.00 coins |
| 9th | Player-9 | XX | 125.00 coins |
| 10th | Player-10 | X | 100.00 coins |

1. Verify that the data of number of **players choosing a particular bet** for the current Roulette spin is **displayed for each bet category (Place bet 2X, Place bet 14X or Place bet 2X)** respectively starting with the player who placed a higher bet and so on.

**Player Perspective Testing:**

Testing has to be carried out based on player perspective. Hence the following can be verified.

1. Verify that **sound and animation** play properly and are in sync with each other.
2. Verify that the application is understandable and all **rules of play** are clearly mentioned to the player.

**Compatibility Testing:**

The game must be tested for its compatibility and support with the underlying hardware/ software. It can be verified through the following.

1. Verify the game for **compatibility testing** across all supported **platforms/ devices like Mac, Windows**, etc.
2. Verify the game for **compatibility testing** across all supported **browsers like Firefox, Edge, Google Chrome,** etc.

**Interrupt Testing:**

The game must be tested whether it is impacted due to any external factors which may influence it.

1. Verify that player’s information including wager, prize won and last won **data is retrievable** at any point **during a network connection issue**.
2. Verify that at any time if the player **closes the web browser** **or tab,** **game spin continues in the back ground** irrespective of any lag or termination from player side.
3. Verify that if the player places a wager and **power failure** happened to the player the player is **awarded the right credits and fair play** has occurred.
4. Verify that **game play is not impacted** when the **player switches to a new tab or window**.

**Non Functional testing:**

**Performance testing:**

Performance testing plays a crucial role for the success of the game. It can be verified through the following points below.

1. Verify **load testing** where in more number of players must place a wager and play thereby memory usage and leak if any could be identified.
2. Verify **long run analysis** and **monitor** for any **slowness or lagging animations**.
3. Verify **stress testing** to analyse the **games peak performance/ threshold** beyond which it may crash when a large number of users try to access/ play it.
4. Verify other parameters like **throughput, response time** must be analysed and abnormalities must be identified.