Analytics Technical Report

Events

We set up 6 Custom Events to keep a track of:

1. Player getting hit by the bullet - To check if the shooting mechanic is easy to use and players are able to hit other players, this can help fine tune aiming and determine if we need to implement aim-assist.

2. Player shooting the bullet - To check if players are drifting enough to shoot a bullet, this can help fine tune drifting.

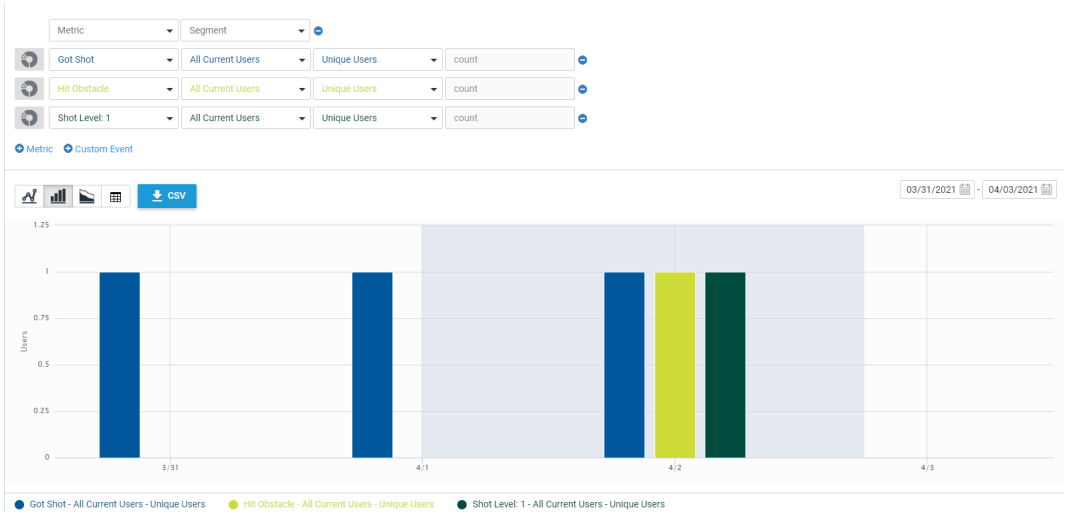
3. Player shooting level 1 bullet - To see if player drifts upto level 1

4. Player shooting level 2 bullet - To see if player drifts upto level 2

5. Player shooting level 3 bullet- To see if player drifts upto level 3

6. Player hitting an obstacle - To see how often does the player run into an obstacle such as wall, this can help fine tune placement of obstacles and level design

We successfully set up all of our events to be tracked. Unfortunately the data did not refresh in time so for this report we manually tracked all of the events through in person observation and not taking.



Unity Analytics

Implementation

We implemented the data tracking in 2 of our scripts: Scr_Movement_Controller.cs and Scr_Shooting_Controller.cs.

```
if (collision.gameObject.layer == LayerMask.NameToLayer("Damage"))
{
    health -= 1;
    //Destroy(collision.gameObject);
    AnalyticsResult result = Analytics.CustomEvent("Got Shot");
```

Event 1 found in Scr_Movement_Controller.cs

```
    AnalyticsResult result = Analytics.CustomEvent("Bullet Shot");
```

Events 2 found in Scr_Shooting_Controller.cs

```
    }
    public void Shoot(float percent)
    {
        if (percent > 0 && percent < 33.33f)
            Analytics.CustomEvent("Shot Level: " + 1);
        else if (percent > 33.33f && percent < 66.66f)
            Analytics.CustomEvent("Shot Level: " + 2);
        else if (percent > 66.66f)
            Analytics.CustomEvent("Shot Level: " + 3);

        StartCoroutine(ShootACoroutine(percent));
```

Events 3, 4, and 5 found in Scr_Shooting_Controller.cs

```
        Analytics.CustomEvent("Hit Obstacle");
    }
```

Event 6 found in Scr_Movement_Controller.cs

Data Collected

Data manually collected for each event over two multiplayer games:

1. Players got hit by bullets 12 times
2. Bullets were shot 78
3. Players shot 61 level 1 shots
4. Players shot 7 level 2 shots
5. Players shot 10 level 3 shots
6. Players hit obstacles 35

Data Analysis

Matches were balanced with both players having been shot many times before one winning.

As both players were new to the game it took some time to get adjusted to the new controls. We can observe from the data that it took the players a few minutes to learn how to properly drift and shoot before attempting to try and hit each other. It is possible that most of the successful hits were more luck based or that another player was unable to dodge as they were unsure how to. In order to be sure we would need to record the pattern of successful versus unsuccessful hits while also comparing times when players were colliding with an obstacle.

Shooting Accuracy

The data shows that over 2 games players shot 78 times. This number is too high considering our current mechanics and intended experience. Player's require 4 shots to die and have no means of dodging besides the basic movement controls which don't provide much freedom against the speed of bullets. This means that shot accuracy is too low, considering how many of the recorded shots must have failed to hit the intended target.

This result was expected, as we've yet to implement our redesigned shooting behaviors which make use of bouncing bullets and homing missiles. However, it's helpful in showing how hard conventional shooting is when paired up with our movement behavior.

Shot level frequencies

The more powerful shots took time to charge via drifting and this was hindered by the fact that players were still unfamiliar with the controls. It is likely they were using the low damage, high rate of fire (ROF) shots over the high damage, low ROF in an attempt to successfully hit something. Players need a greater incentive to spend time drifting, and thus moving slower, than to do a bit more damage.

Player hitting Obstacles

Data showed that the instances of player hitting obstacles was quite high. But the rate of players hitting obstacles will likely go down the longer they are able to play. They will get better at steering the cow and ultimately have more control at avoiding the obstacles. We are currently trying to implement being able to reverse as a quick solution to this problem.

Reflection

The first challenge we faced was implementing the Analytics as this is something we had never done before and the (24hrs) delay in the results showing up on the dashboard only added to the problem. The trickiest problem we faced was getting the events to pop up on the dashboard and it turns out that Custom Events will not show up if you have an Ad-Blocker enabled (thank you unity forums). Coming up with criteria on what kind of events we need to keep a track of and how we can use the data to fix or improve our game was also challenging.