**Positive and negative feedback loops**

**Positive feedback loops**

With a positive feedback loop, something happens that causes the same thing to happen again, but that thing is stronger with each iteration. This allows for players in front to get a massive advantage if they do everything well on their first try. For example with a racing game, if one player is able to avoid the obstacles, whilst the other is not, a positive feedback loop could be used to further reinforce their positions, by giving the better player rewards for doing well (perhaps power-ups that destroy obstacles) and punishing the worse player by making the obstacles harder and harder for them to avoid without the power-ups.

Positive feedback loops destabilize the game, as it purposely is pushing the winning player further and further away from the losing players.

They tend to cause the games to end faster, whether it is through the fact that the winning players get benefits to help them win faster. Or whether the players just decided to end the game, as the winner is already decided. This also shows that positive feedback loops put emphasis on the early game, since the effects of the decisions made in the early-game are increased over time. If one player avoids an early-game obstacle and the others do not, that will put the winning player on an advantage, as they will be able to collect power-ups that could further put other players back or help them get past future obstacles, without the risk of other players catching up to him.

**Negative feedback loops**

On comparison, negative feedback loops are supposed to balance the game out. If something happens that puts one player on an advantage, then negative feedback loops are implemented to reduce the chance of that same thing happening again. That way negative feedback loops make it easier for the opponent players to catch up to the winning player.

The idea of ‘rubber-banding’ can be applied to negative feedback loops, as what they are essentially doing is creating scenarios that make the players ahead fall back behind, and players that are behind to catch up. This makes the negative feedback loops to stabilize the game, by moving all players closer together and creating a scenario in which there is no-clear winner.

In return, this may cause the games to get longer, as all players have a fairly equal chance of winning. This puts the emphasis on late game rather than early game, as in the end the players would all be relatively close to each other in terms of who will win the game. So, if one players dodge the obstacle at the beginning, rather than making it easier for them to dodge future obstacles, it will make it harder for them, thus gravitating them towards the centre of the group.

All feedback loops have three components: a sensor, that monitors the game state; a compactor, that decides whether an action needs to be taken; and an activator, that modifies the game if the compactor decides that it is the right thing to do. As an example, in a racing game, the sensor monitors the distance between the players. The compactor will then check the sensor to see whether one of the players is further ahead or behind, depending on some kind of threshold, and if they are, then the activator comes into play, by either speeding up the losing player to allow them to catch up, or slowing down the winning player to let the others catch up. Or instead, slow down all the other cars to let the losing player catch up, or speed up the other cars to let them catch up to the winning player.

In our game we already have a positive feedback loop. The node is spawned and moves on the belt at a certain speed, node is called out by customer, node is tapped by the player in the expected range. Depending on these early-game decisions that are made by the players, then the game could technically be won by won of the players early on. If one of the players is really successful at the game and during their first few turns they get perfect taps, then they will steal more customers from the opposite players, and were the opposite player fail, or only just get the tap, they will steal a smaller amount of customers which in turn puts them on a disadvantage as they are already trading with what were originally their customers.

So, in other words, if each player had 20 customers at the start: (20:20)

* Player 1 gets perfect tap and steals 3 customers from player 2 (23:17)
* Player 2 gets a tap and steals only 1 customer (22:18)
* Player 1 gets another perfect tap and steals 3 customers (25:15)
* Player 2 gets another normal tap and steals only 1 (24:16)
* Player 1 gets another perfect (27:13)
* Player 2 gets another normal tap (26:14)
* And so on.

In the end, all the players do is shuffle player 2’s customers around and there is no implication of any threat for player 1. Should that scenario occur, that would put player 2 at a big disadvantage and would potentially discouraged them from playing the game further as they could see that there is already a winner.

Of course, the reverse of the situation could happen, where player 1 is really terrible and misses all their taps in the first few rounds. This would result in player 1 being put at a disadvantage in the later game as they will need to take longer to recover all the lost customers.