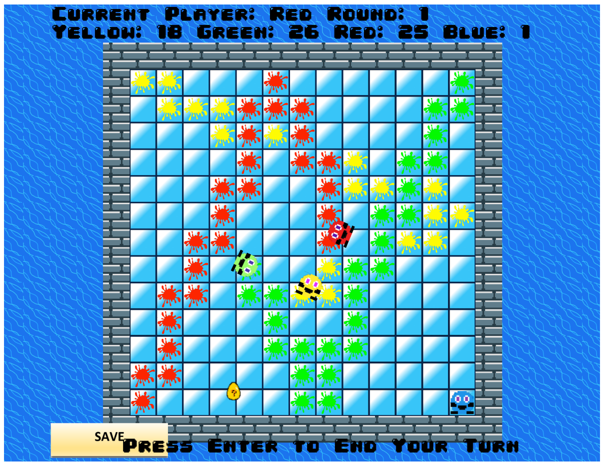
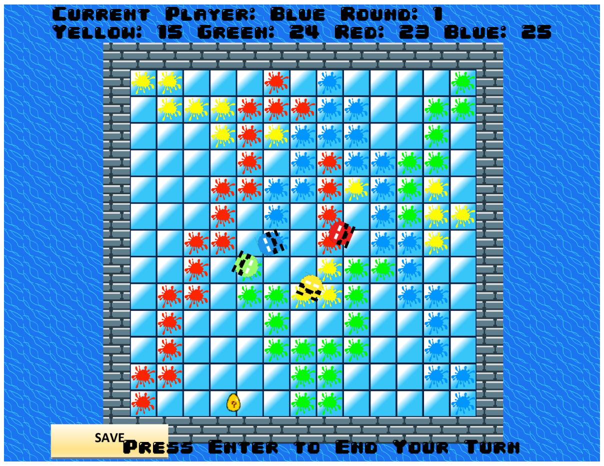
Game Balancing

Our game uses a **turn-based system.** There are 5 rounds in total and each of the 4 players gets to move their character blobule once in each round. This means that there are a total of 20 moves per game. The goal of the game is to colour over as many tiles on the board as possible, as whoever has the most tiles in their colour wins.

However, an inherent problem with the way our game is set up is that **whoever moves last on any given round has a significant advantage.** This is because they are able to colour over and eliminate any progress other players have made with relative ease. An example of the shown below. Please note that in our current implementation, the Blue Blobule moves last.

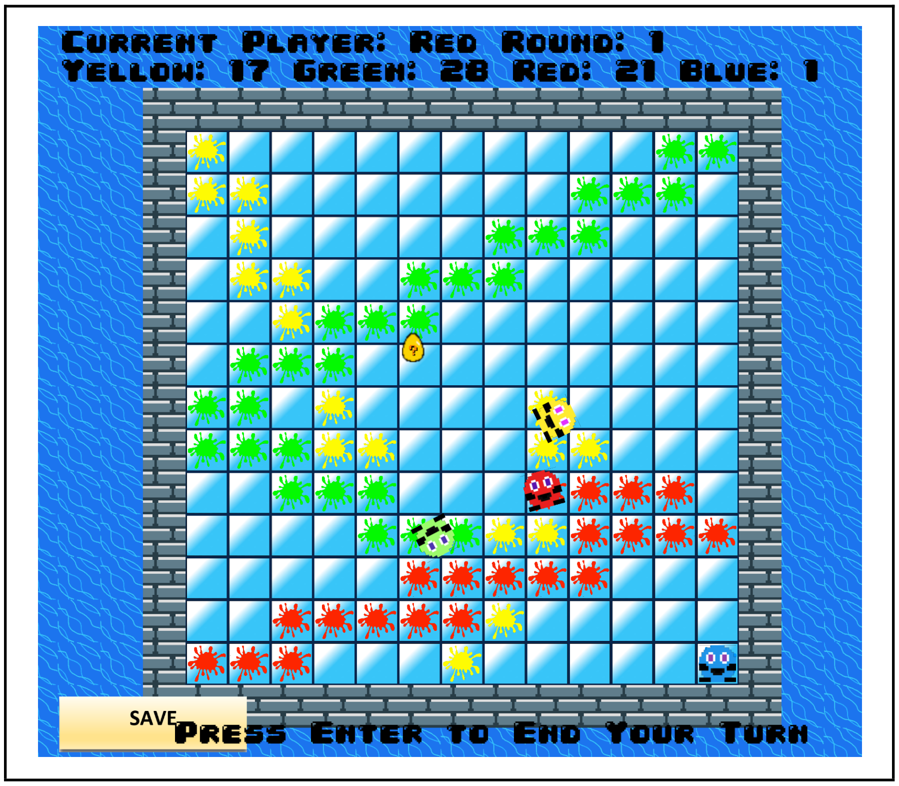
Before the Blue Blobule Moves	After the Blue Blobule Moves.
 CURRENT PLAYER: RED ROUND: 1 YELLOW: 18 GREEN: 26 RED: 25 BLUE: 1 SAVE PRESS ENTER TO END YOUR TURN	 CURRENT PLAYER: BLUE ROUND: 1 YELLOW: 15 GREEN: 24 RED: 23 BLUE: 25 SAVE PRESS ENTER TO END YOUR TURN

Notice how the Blue Blobule has more points compared to the others at the end of the round. Furthermore, take note of how many points the other Blobules have lost as a result. This will only serve to frustrate players who have worked hard to gain an advantage. Hence, our solution to this problem is **to reduce the maximum velocity for the last player to move in each round.** However, what should this maximum velocity be?

To figure out the best maximum velocity for the last player, we can assign **a score** to a given velocity and determine its value from that. We will calculate the score using the given formula:

$$Score\ =\ (\#\ Opponent's\ Tiles\ Blue\ Colours\ Over)\ /\ (Total\ \#\ of\ Tiles\ Coloured\ over\ by\ Opponen$$

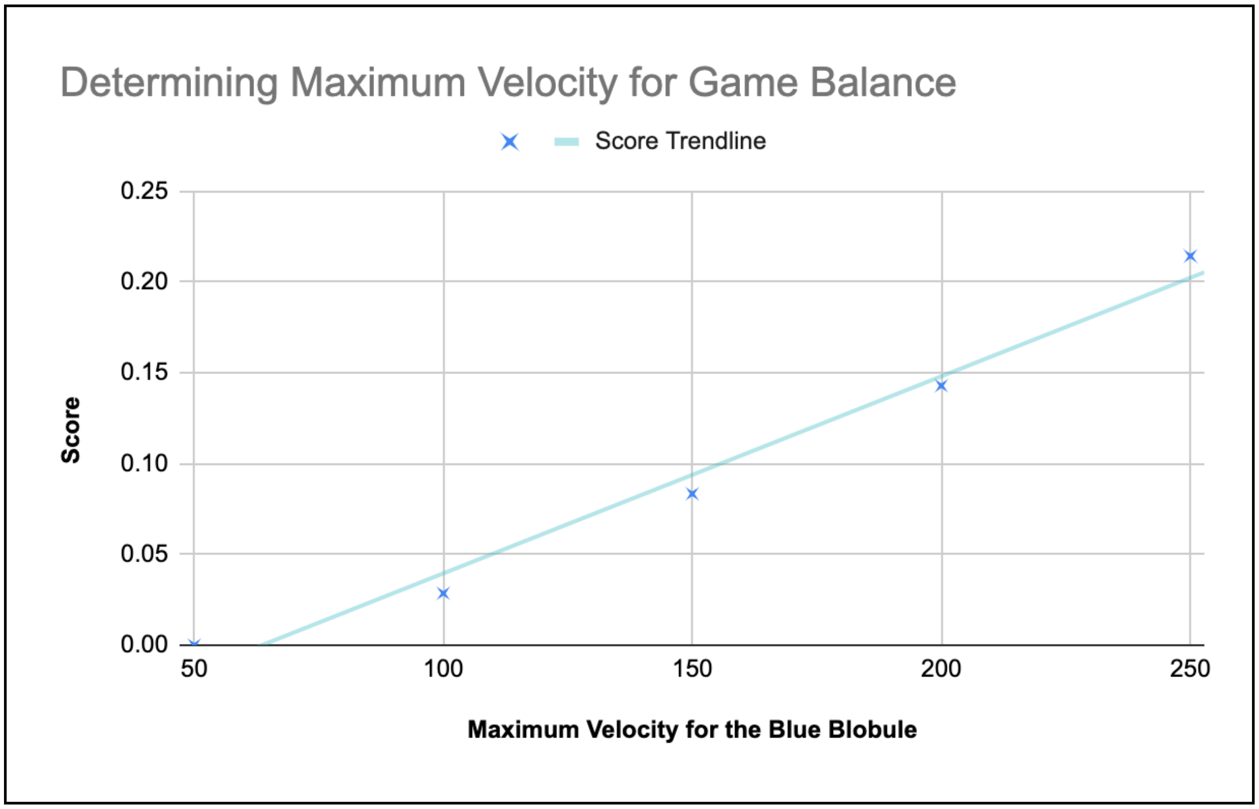
To measure these score values, we decided to create a map specifically for testing purposes. Below is what the map looks like.


CURRENT PLAYER: RED ROUND: 1
YELLOW: 17 GREEN: 28 RED: 21 BLUE: 1
SAVE
PRESS ENTER TO END YOUR TURN

It is void of anything other than walls along the border and ice tiles. Furthermore, it is large enough such that blobules will not be constantly bouncing off each other. We will **only play one full round** for **each value of maximum velocity** we decide to test. We save the state of the current board for a fair experiment. At the end of the round, we will calculate a score and record it. Note that this was done manually as it was difficult to program a CPU that plays optimally.

We then compare each score to the maximum score you can receive on this map with a maximum velocity equal to the other blobules. In this case, it would be 0.214.The idea behind this formula is that if the ratio is $\approx\ 0.214$,that means blue's velocity is too large; the player is able to colour over too many tiles. If the ratio is $\approx\ 0$,that means blue's velocity is too small; the player is not able to colour over any of their opponent's tiles. We want a value that is relatively close to half of 0.214, so 0.107.This ensures that the player can colour over their opponents' tiles without it being seemingly unfair.

Below is a graph of score against maximum velocity values.



As you can see, there seems to be a linear relationship between score and blue blobule maximum velocity. Through data processing, we can determine that the trendline of this graph has the equation $y = 0.00109x - 0.069$.Since we know that the score we want is 0.107,we can determine the optimal maximum velocity by substituting this value into $y$and solving for x. As it turns out, the optimal maximum velocity is $161\ (3\ s.f.)$ We will be using this value for our game.

For further balance, we decided to make our default map **symmetrical.** We noticed that on our original map, the blue blobule began with a large disadvantage as it started on the side with mud tiles as opposed to ice tiles which have less friction. Hence, we had to consider making balanced maps to play on as well. This was our solution to that problem.