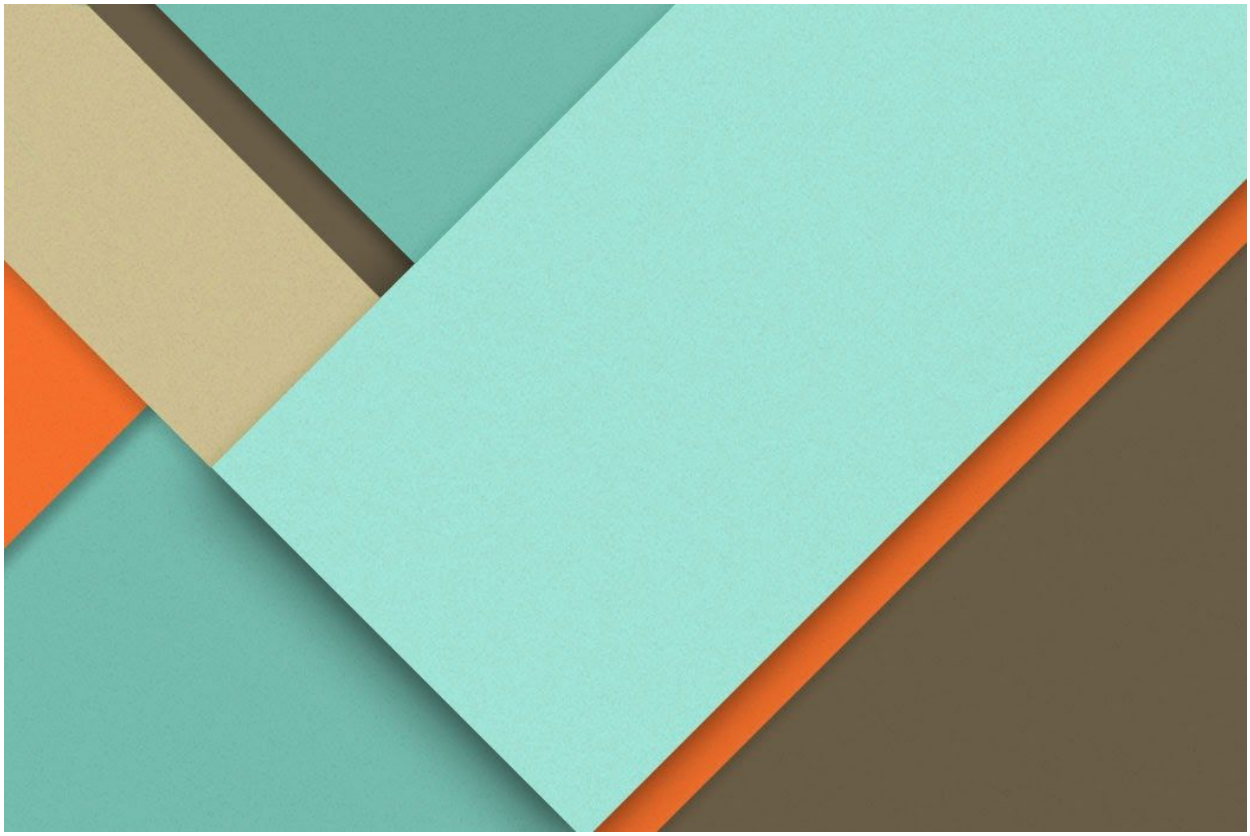


Game Prototype:

Aquario Kart: Double Splash!!

November 2020

Team Markedly Minified Olives (MMO)
Mark Pilgram
Min-Shan Luong
Oliver Jung

Goals

As the core gameplay of our game has already been proven to have the potential to excite players by several pre-existing games in the genre (Mario Kart series, Sonic & All-Stars Racing series, etc.), the primary goal of our prototype is to test the effects of the part of our game idea that should make it stand out from the games mentioned above: The water tank system.

The goal of this system, in addition to the item boxes found in kart racing games, is to create a more dynamic racing experience where no player can be certain of their victory or defeat until they reach the finish line. Similar to how many racing games have a sort of “rubber banding” system that allows AI competitors to exceed the limits of what their vehicles should be capable of in order to keep up with potentially smarter players, these mechanics are designed to give players towards the rear of the pack advantages that allow them to catch up to the rest of the group. As a result, the lead between first and last place should be kept fairly small despite potentially significant skill differences between players and the order of racers should change frequently.

These differences in race-progress between competitors and the shuffling of competitors’ positions shall be the focus of our prototype.

Prototype structure

For our prototype most of the aspects of a racing game are reduced to simple numbers and dice rolls in a turn-based system. Each racer covers a certain amount of distance per turn, moving ever closer towards the finish line. Their goal is to maximize the amount of distance covered each turn while minimizing the distance their competitors travel.

Our race track is modeled simply by its length, which combined with the average speed of racers controls the number of turns before the race is over. Item boxes and predefined water sources are placed along the path at fixed intervals, potentially granting racers items or water when they pass them between turns. Both the item boxes and water have a pickup-chance variable and water sources also define an amount by which water tanks are filled. This means that picking up an item or filling the tank with water is decided by a dice roll when such an event occurs.

> ***Example:*** *The race track is 3000 meters long. At intervals of 400m, there are item boxes that are picked up with a chance of 80%. Various water sources such as puddles, streams and waterfalls are placed along the track as well. Puddles have a pick-up chance of 30% and fill the tank by 40%. Streams and waterfalls have a pickup chance of 100%, with the stream filling the tank by 60% and the waterfall by 20%.*

The next part of our model are the racers themselves. Racer speed (the distance they advance in a round) is controlled by several factors:

- **Base speed:** Guarantees progress no matter what.
- **Skill:** a random base speed modifier designed to model player skill that remains unchanged throughout the race. Each player's skill value is defined at the start of the race.
- **Luck:** A random speed modifier that is updated each round via dice roll, designed to represent random/unpredictable events that may occur during the race. Driving over a puddle reduces the value by a predetermined amount for one turn.
- **Water tank speed multiplier:** A multiplier applied to the sum of all values above. The multiplier is defined as the water tank fill level times the maximum water boost.

Speed = (Base Speed + Skill + Luck) * (1 + maximum boost * tank fill level)

It should be noted that if water is available in the tank, a little bit is used each round to provide the speed boost multiplier.

> ***Example:*** *Racers have a base speed of 30 meters per round. At the start of the race, each racer is assigned a skill bonus of between 0 and 5 meters per round. The Luck variable ranges from 0 to 15 meters per round. The maximum water speed increase is a multiplier of 1.25x. If water is available, 10% of the maximum water level is used to increase speed each round.*

The final part of this model are the game's items. Items essentially affect the progress or vehicles. Some items will give the player that is using the item a progress boost, others will negatively affect the progress of one or more competitors within a certain range. Items also define a water cost to be used, an amount of water that the item can give if it is discarded and properties of the puddle that is left behind when using or discarding the item. Vehicles that are negatively affected by an item also lose a certain amount of water.

The item obtained from an item box is random, though the pool of available items may vary depending on the position of the player picking the item up. To summarize, items have the following properties:

- **Item Name/Icon:** For identification purposes.
- **Area of Effect:** Self, single opponent or multiple opponents.
- **Range (if AoE is not self):** defines how close the item's target(s) need to be to be affected.
- **Progress & water level modifier**: Amount of progress gained or lost by the affected target(s), amount of water lost by targets
- **Water cost:** Amount of water from the water tank that is consumed on use.
- **Water gain on discard:** Amount of water added to the water tank when the item is discarded
- **Puddle on use:** Water pickup chance & water gain, if item leaves a puddle
- **Puddle on discard:** Water pickup chance & water gain from puddle left by discarding the item and converting it into water
- **Pickup Condition:** Allows items to be defined as unobtainable for racers

To simplify the process, puddles created by items may be instantly applied to vehicles behind the item user rather than requiring them to pass the point on the track where the item is used.

Example:

***Item Name**: Rogue Wave*
***Area of Effect:** multiple opponents*
***Range:** 80 meters (forwards **or** backwards)*
***Progress Modifier:** -50 meters, -20% water*
Water Cost: 40%
Water gain on Discard: 50%
***Puddle on use:** Does not leave a puddle*
***Puddle on discard:** pickup chance: 40%, water gain: 20%*
***Pickup Condition:** Third place or lower*

With these systems in place, the race is simulated round by round until all contestants have passed the finish line. Evaluations can be made by looking at how the difference in progress and order of the racers changes over the course of the race.

As a more mathematically-driven board-game concept, this process can be realised in a physical or digital form. For the purposes of prototyping and not creating a board game the digital approach offers several advantages, as saving a snapshot of the game's state after each round for later review can be done with more consistency and less effort. It also allows us to play the game together while maintaining social distancing, with each player controlling one or more racers.

Rules & Procedure

As mentioned in the previous section, a set of data is stored for each racer that is used to determine their current capabilities and visualize their progress in the race. Additionally we have certain global parameters, like the water racers lose each round and the speed multiplier granted by a full water tank.

At the start of each round, this data is used to perform a number of actions for each racer. We used the following order of actions in our testing:

1. Recalculate current velocity using the formula

   Current Velocity = (Base Speed + Skill Speed + Luck Speed)
   *** (1+ Full tank Speed increase *Tank fill%)**

   The table shown on the following page can be used to assist in this process

2. Update progress

   New Progress = Old Progress + Current Velocity

3. If a player is using a shield item, decrease their shield active round counter
4. Allow players to use or discard items
5. Pick a new luck speed value for each player (in our case: random number between 0 and 15)
6. reduce tank fill by the water loss per round value (in our case: 10%) for all players with water left in their tank
7. Account for pre-placed water sources on the track. We used these parameters:
   puddle: 50% pickup chance, +40% water, luck speed -2,
   Waterfall: 100% pickup chance, +20% water,
   River: 100% pickup chance, +60% water, luck speed -4
8. Account for items on the track. Grant players that drive past an item box a random item from a list of items dependent on their position. We used following items & item lists:
   a. 1st:bomb/shield/freeze
   b. 2nd: bomb/shield/boost/freeze
   c. 3rd: wave/bomb/shield/boost/freeze
   d. 4th/5th: wave/bomb/shield/boost
   e. 6th: wave/shield/boost
9. Update graphical visualization of the race

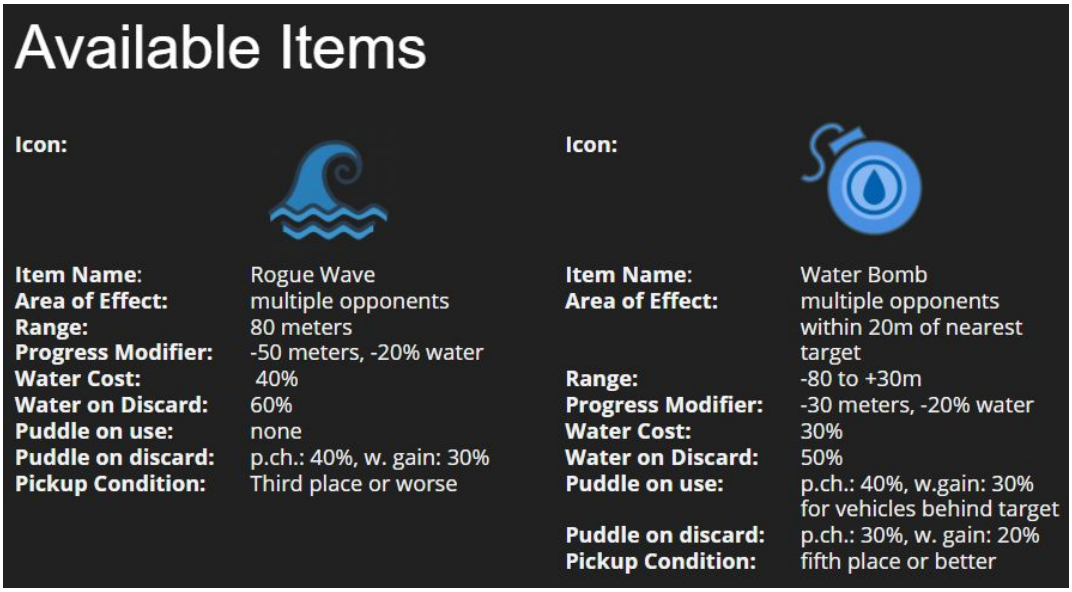
Water Level	Speed																				
	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
0	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
10	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
20	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	53
30	32	33	34	35	37	38	39	40	41	42	43	44	45	46	47	48	49	51	52	53	54
40	33	34	35	36	37	39	40	41	42	43	44	45	46	47	48	50	51	52	53	54	55
50	34	35	36	37	38	39	41	42	43	44	45	46	47	48	50	51	52	53	54	55	56
60	35	36	37	38	39	40	41	43	44	45	46	47	48	49	51	52	53	54	55	56	58
70	35	36	38	39	40	41	42	43	45	46	47	48	49	51	52	53	54	55	56	58	59
80	36	37	38	40	41	42	43	44	46	47	48	49	50	52	53	54	55	56	58	59	60
90	37	38	39	40	42	43	44	45	47	48	49	50	51	53	54	55	56	58	59	60	61
100	38	39	40	41	43	44	45	46	48	49	50	51	53	54	55	56	58	59	60	61	63

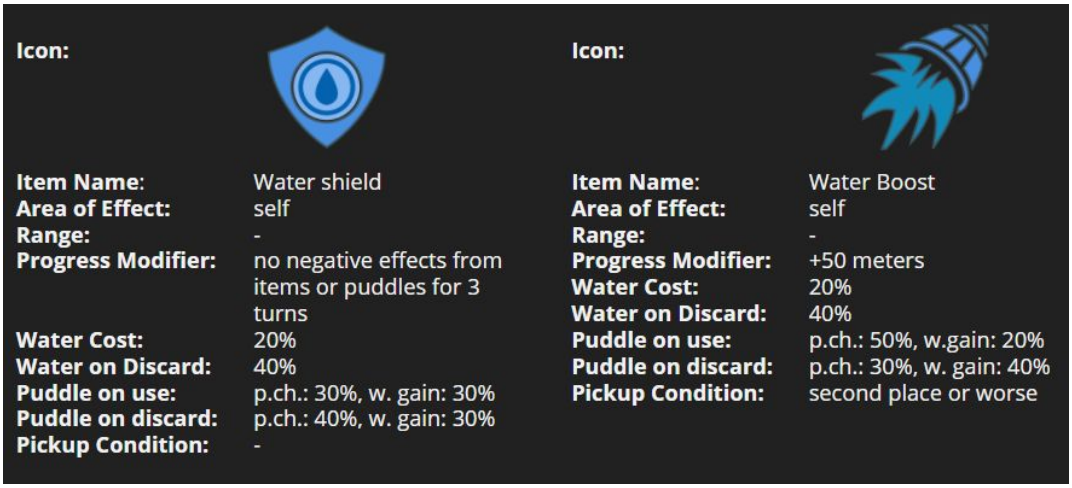
Lookup table for velocity calculations assuming a full tank speed multiplier of +25%

E-Paper Prototype

For our digital version of our prototype we used google slides, as this allowed us to easily play through our prototype, with each of our 3 group members performing the calculations and making decisions for two of the six total racers. This total number of contestants is an arbitrary value, we estimated that 6 was a large enough value to represent a typical kart-racing scenario while also being small enough to manage the calculations needed for each round.

We first created a set of slides that defines the stats of all all items and events in our prototype:





Icon: 

Item Name: Freeze Mine/Freeze spray
Area of Effect: single opponent
Range: -infinity to +20m
Progress Modifier: -20m
(if used backwards: 50% chance to miss target until a target is hit or all have passed)
Water Cost: 20%
Water on Discard: 30%
Puddle on use: -
Puddle on discard: p.ch.: 40%, w. gain: 30%
Pickup Condition: third place or better

Item availability:
1st:
bomb, shield, freeze
2nd:
bomb, shield, boost, freeze
3rd:
Wave, bomb, shield, boost, freeze
4th/5th:
Wave, bomb, shield, boost
6th:
Wave, shield, boost

Event Descriptions

Item Box
Icon: 

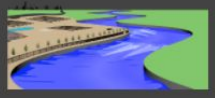
Effect: Player receives a random item
Effect Chance: 100%

Waterfall
Icon: 

Effect: Player Receives 20% water
Effect Chance: 100%

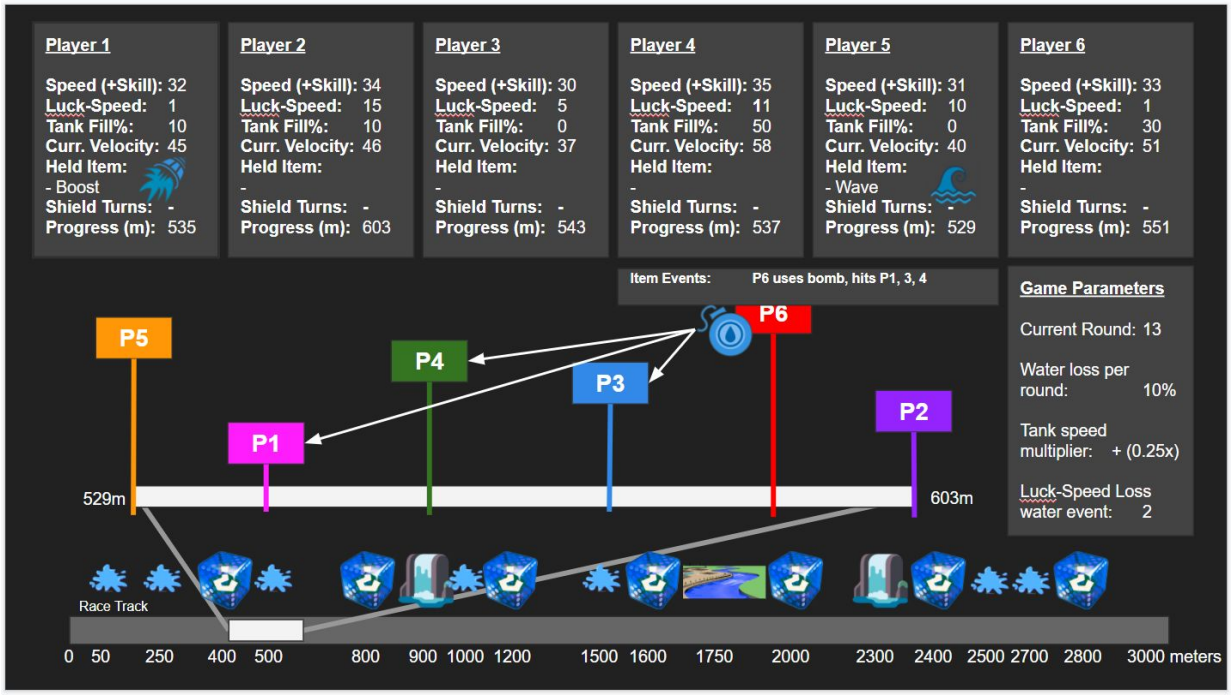
Puddle
Icon: 

Effect: Player Receives 40% water
Luck speed decrease
Effect Chance: 50%

River
Icon: 

Effect: Player Receives 60% water
Luck speed decrease x2
Effect Chance: 100%

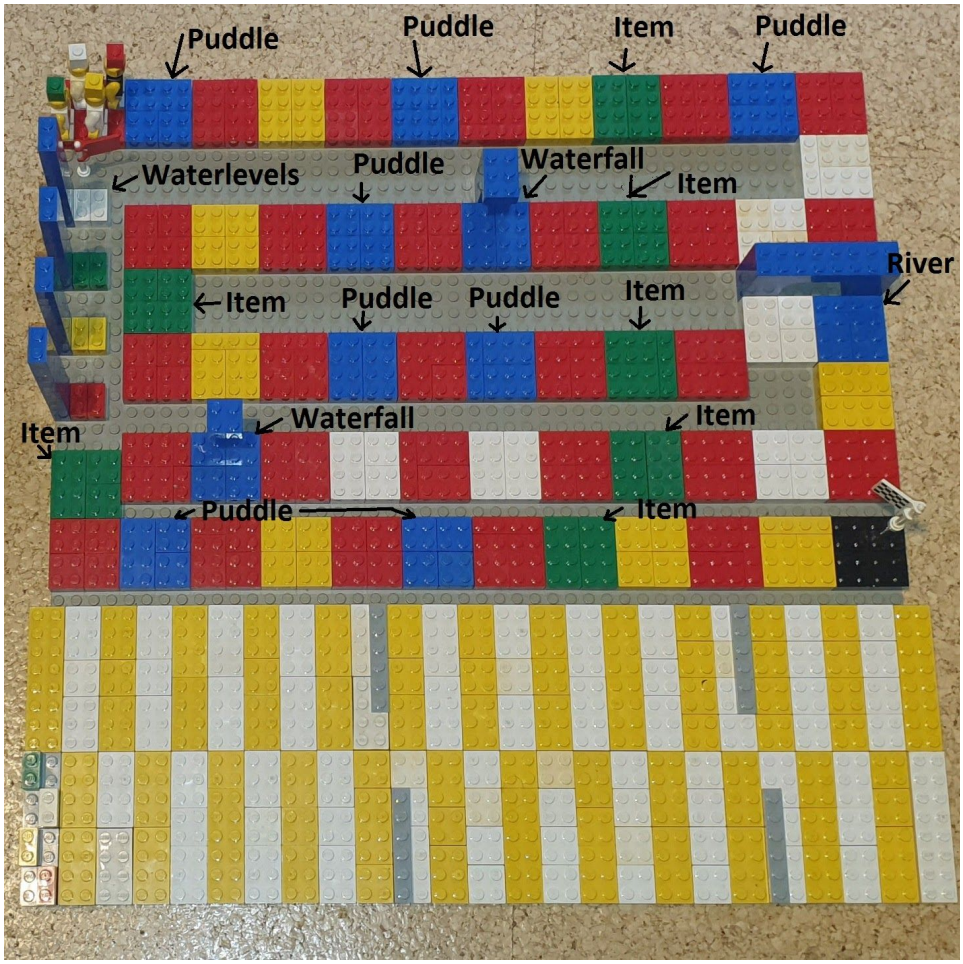
With these descriptions in place, our game board used to represent the current state of the game looked as follows:



We store all player-specific data at the top of each slide. Additional game parameters are displayed on the right. At the bottom we have our race track overview, denoting the positions of all pre-placed water sources and item boxes. The view above this is designed to highlight the current order of racers, the distance between first and last place and show the effects of item uses. A player's choice to use or discard an item is also noted in the item events list in the top right of that area.

At the end of each round, the state of the current slide is copied to a new slide, where the game can then continue. This way we were able to save the state of the game after each round for both error correction and later review. This way the events of a race can be replayed as a literal slideshow.

Physical prototype



Our physical prototype is designed to be played with four players. It features multiple item cards, a main racing track and a detailed tile progress view. The main racing course consists of 60 fields representing 50 meters each whereas the tile progress view has 50 fields and displays the player's progress inside a single field of the main track. If a player clears the last field, his figure moves one tile on the main track towards the finish line. Furthermore, the different colours on the main track represent different tile types:

- **Red/Yellow/White:** Normal tiles without effects
- **Blue:** Water (e.g. puddles, waterfall, river)
- **Green:** Items
- **Black:** Finish

Waterfalls and rivers are marked with special structures next to a blue tile. The players activitate the effects of a field when they enter it the first time.

An example of the board layout at a later stage in the game.

Experience

Performing the calculations for each participant in the race and updating the visual representation of the race each turn proved to be rather time consuming. Originally the goal was to have our race go on for around 65-70 rounds, but we stopped after about 40, as it had already taken us more than 8 hours to play this far. The calculation aspect could have probably been accelerated if it had been offloaded from the slides to some table-oriented software (e.g. google tables). Despite failing to complete our race track, it should however be noted that the length of the race track was arbitrary to begin with. We stopped at a point where we felt that we had seen the impact our various game mechanics could have.

Another issue with this design is that several aspects of a racing game had to be reduced to dice rolls to fit the turn-based board game format, even though player choice would play a more significant role in an actual racing game. In particular we felt that our water gathering mechanics were driven too much by luck in this prototype than they likely would be in the final game, where players would be able to choose for themselves if they attempt to drive over a puddle rather than having that decided by a random number generator.

Over the course of playing our prototype we also realized that we hadn't come up with a way to handle simultaneous item usages within a single turn, as the order in which items were used could drastically change the outcome of the race. Fortunately this problem could be addressed then and there, with our solution being to apply all item effects to the state of the game before item calculations were made, with stronger penalties overriding the effect of smaller ones in cases where a player was negatively affected by two items simultaneously. This behaviour is not too different from what you'd see in a proper kart racing game, where players may become temporarily impervious to negative effects after getting hit.

In summary the race that we played did feature a fair amount of dynamism regarding the order of its participants. The distance between first and last place stayed within a reasonable range, there was still a chance for the last place player to catch up at basically any point in time. The first and last place spent most of the race within 2-3x the original starting distance (25m on a race track with a length of 3000m) to one another. Our item & water system created scenarios where players had to choose between using and discarding items, though the balancing in that area did leave room for improvement.

Revisions made before testing the prototype

With this prototype our goal was to more clearly define our water tank mechanics and test their effects. Some changes were already made during the design phase of our prototype, as their benefits to the game's balance was already clear.

- **Constant water loss from driving:** To prevent a scenario where the player in first place could fill up their tank and speed away from the rest of the pack who would then be busy fighting each other, we determined it would be necessary for vehicles to gradually use up the water in their tank over time.
- **Discarding items also leaves a puddle behind:** In combination with the previous point, any attempts by the first place player (or players further ahead in general) to fill up their tank should also have the opportunity to benefit the players behind them, giving them more chances to use items or gain speed to catch up.
- **Position-dependent item distribution:** Which items can be obtained from an item box should depend on the player's position. This accomplishes two things: It allows us to prevent players from receiving items that are of little use in their current situation and allows us to give players who are further back more powerful items to help them catch up.
- **Water loss for targets hit by an item**: This change was perhaps more optional, but the idea was to give items more impact, and decrease the chance of scenarios where upon passing a player who was slowed down by an item, the player who was overtaken is immediately able to retaliate with full force.

Revisions and Discoveries made after testing the prototype

- **Water tank balancing:** With the parameters and race track design used in our prototype, situations where players had an empty water tank and couldn't use items were slightly more frequent than we would have liked. These balancing parameters can't be directly transferred to the full game, but are something we will have to look out for when designing our track(s) and balancing our game.
- **Item Cost & Player position**: Our prototype also highlighted the importance of item usage costs. With this parameter we could make it easier for players who have fallen behind to receive items that are cheaper to use, making it easier for them to use more items and maintain a relatively full water tank.
- **Water loss for targets hit by an item:** With the general lack of water that players faced in our prototype, this mechanic didn't really help improve that situation, so its impact and usefulness will have to be reevaluated for the final game.
- **"Blue Shell"-equivalent:** Though the ordering of competitors during our playthrough of our prototype remained fairly dynamic, we do think that having an item that can decrease the distance between first and last place might not be a bad thing to have. Here Is our idea for what such an item could look like:

    Snowstorm: Clouds form in front of the player in first place and the snow they drop freezes the road ahead of them. As a result players further forward (in general, not just the player in first place) have to deal with a more slippery environment. The ground unfreezes after a short period of time, meaning that players further back are not affected.

- **Item list rebalancing:** With the way we had our prototype configured, the last place player was able to obtain the shield item from item boxes, that was of relatively little use to them. The distribution system for items from item boxes is another balancing parameter we will have to pay attention to in our final game.