

Milestone 1: Game Idea

Fall For Me!

Team Cicisoft

Andrea Solanas de Vicente

Ankur Deria

Michael Dey

Bendegúz Timár

1. Game Description

Our game **Fall For Me!** is a fusion of the platformer and fighting genres, drawing inspiration from popular titles like Super Smash Bros., Super Mario Bros., Brawlhalla or Stick Fight: the Game. In this local multiplayer arcade game, players team up to compete on an infinite roller coaster serving as a dynamic platform, complete with obstacles and collectible items that spawn along the track. Throughout the game the players need to collect as many items as possible and deposit them at the team's designated station for points. The stations appear at fixed intervals, alternating between teams. To hinder their opponents' progress, teams can knock their opponents off the roller coaster, causing them to lose their collected items. The players can only spawn at their respective base after they fall off the roller coaster. The team to collect and register the most collectibles wins. The game also offers a menu where players can customize the win conditions, like setting time limits or the number of collectibles needed to win.

Context and Narrative

You are at the renowned Doomsday Land theme park, standing amidst the bustle of carnival games, thrilling rides, and delicious food stands. The air is electric with excitement as the screams of thrill-seekers fill your ears. You take a moment to soak it all in, feeling grateful for this opportunity to let loose and have fun.

Out of the blue the park's director, Archis Barchivald, announces that the park is closing down, much to the frustration of the visitors. As a means of distracting the angry crowd, Dr. Barchivald orders a shower of sweets to rain down from the sky.

In the midst of the chaos, you spot a vantage point on the coaster tracks that could give you an advantage in the sweet collection. Without a second thought, you climb into the nearest coaster car and begin to collect as many sweets as possible because who doesn't like free stuff.

In this bizarre adventure, you ride a roller coaster that goes through perilous twists and turns. You encounter obstacles and opponents trying to knock you off track and snatch your sweets. You must use your wits and quick reflexes to stay on course and gather as many sweets as possible. With the ride speeding up and the sweets raining down, the excitement reaches a fever pitch as you compete against your fellow passengers to be the ultimate sweet collector.

The game befits the theme of the course by having the roller coaster as not just location to the game, but also as a central component that affects the gameplay greatly.

Gameplay

The players form two teams. They ride an infinite roller coaster and aim to collect as many items as possible while dodging obstacles and other players. The coaster moves automatically. Stations respective to the teams appear in set time intervals, where the players of the team can

deposit their collected items. The stations appear alternating between the teams. The players can attack each other to push one away. If the player gets knocked off the roller coaster, they lose all their collected items that were not deposited before. The game goes on until a set win condition.

Player mechanics

The players move left and right to reach collectible items but also to keep up with twists and turns of the roller coaster. Players can also jump to reach higher platforms or avoid obstacles, but while jumping, the player loses some momentum gained from the coaster. This makes the roller coaster leave the player a bit behind, therefore players must time their jumps carefully to avoid falling off or missing out on valuable items. To knock other players off the coaster, players have a punch mechanic. The players can deposit their collected items at their designated stations to earn points for their team.

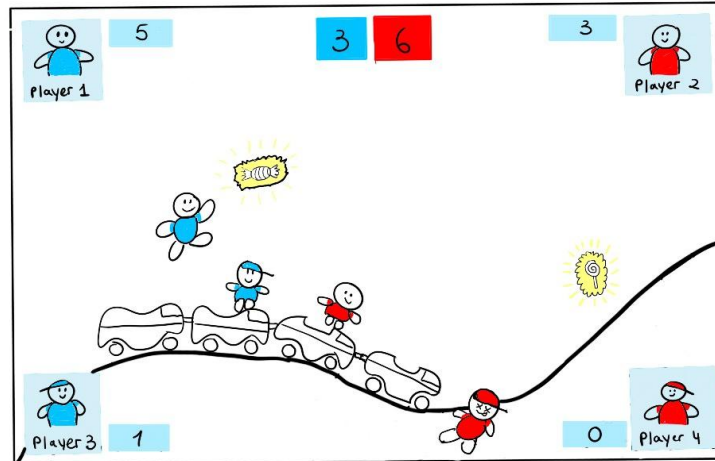
Roller coaster mechanics

The roller coaster will behave like a real one. It will slow down while moving up and pick up speed while moving down. It will have a fixed minimum speed. The track will consist of ups and downs (and loops, being part of the desirable target). The roller coaster never stops, even if it reaches the base of a team. The players need to jump to their base and then jump back on the roller coaster after registering their collectibles. The momentum of the roller coaster would carry on to the players (high target).

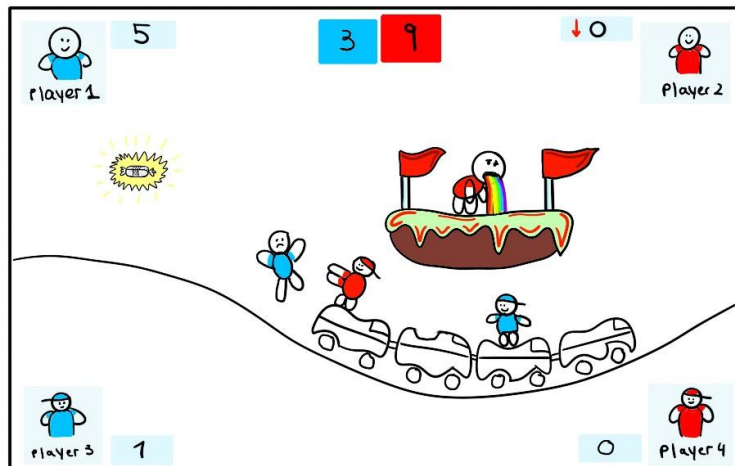
Concept art

Gameplay

We did a few sketches of what we wanted the main gameplay to look like without much detail. In the first sketch, the general gameplay can be seen, showing the rollercoaster as the base for the characters to jump on and move. There are also some collectibles on the screen which the characters are trying to obtain. We added respective huds to the corners of the screen so the players could see what number of collectibles each character has, with the overall sum of the collectibles collected and dropped off in the bases being at the top of the screen.

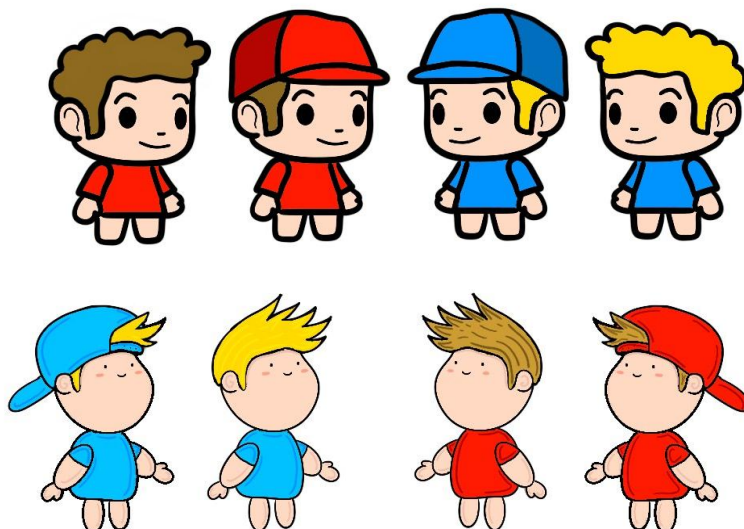


In the second design we added an example of what a base could look like and how the characters can drop off the collectibles. To show how the players can push each other off the rollercoaster we added it in this sketch as well.



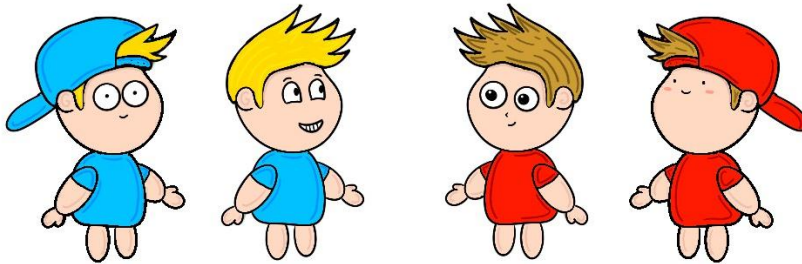
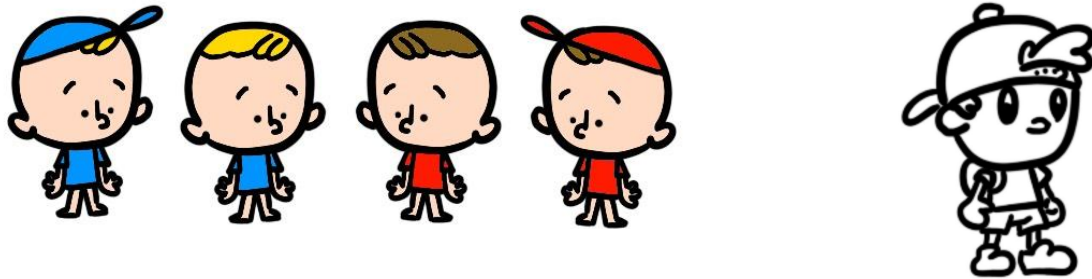
Characters

We did the basic design of what the characters and how the screen would look like. Later on we started designing more in depth what the characters' style would be like. We did multiple options of sketches to have different ideas to choose from.



For the different characters, we followed the same idea of two different teams (red and blue). There had to be key differences between them, like the hair color or their shirt color, so they are easy to see on the screen. Also, we decided to do the same character but with different elements (like adding a cap

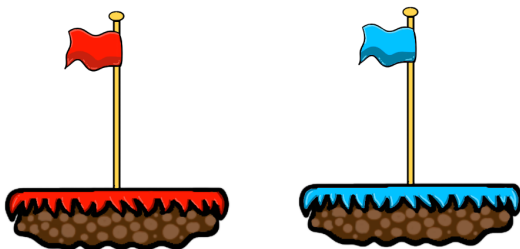
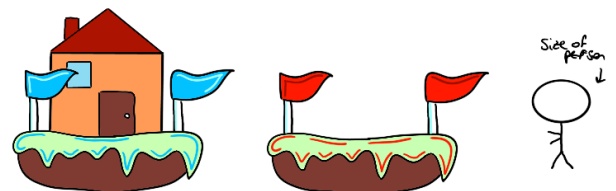
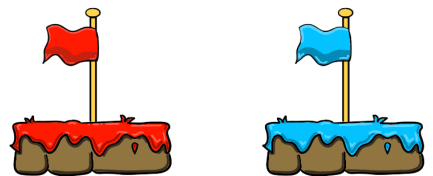
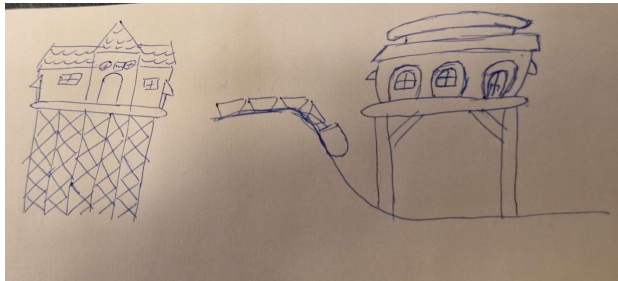
for the second player of each team). Here are some of the different designs we thought of.



In some of the designs, we tried different features to see which one would look better.

Bases

For the bases, we thought of making them similar except for the color. Each team has a specific base where they can drop off their collectibles. This is why the colors are what identify both bases from each other. We also thought of adding a characteristic component like a flag or house to symbolize that this object is a base.



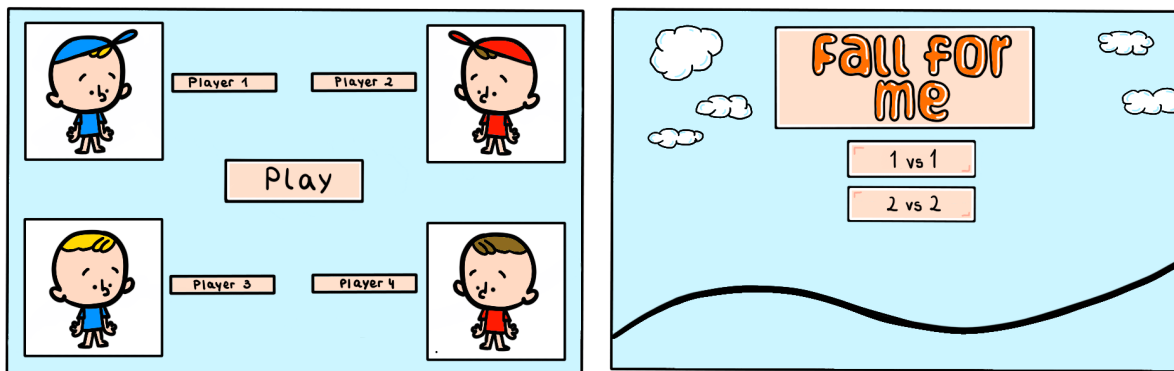
Collectibles

For the collectibles we thought, since we are in an amusement park, the food you usually get there is candy. Also, the main characters that the players can use are little kids and they love candy. We did a few sketches and colors for what we want the items to look like.



Main Menu

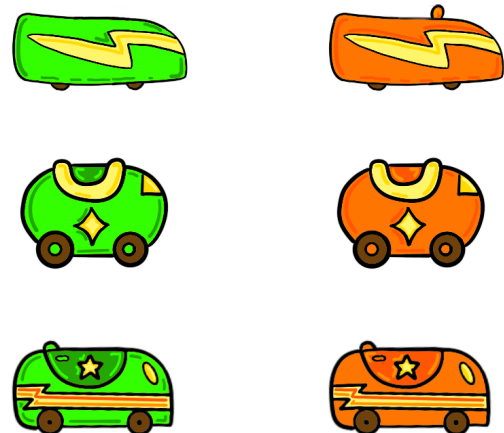
We also did very quick sketches of what the main menu or the choosing of characters' screen might look like. It will probably change in terms of design, but we wanted to get some idea of what it might look like.



Roller coaster

For the design of the rollercoaster, we needed something that would serve as a base for the players. It also can't clash with the colors of the teams. We designed a few sketches for what it could look like and different colors we could do.

The track will be composed of a line (more likely brown) so that the roller coaster can lay on top of it and move according to it.



2. Technical Achievement

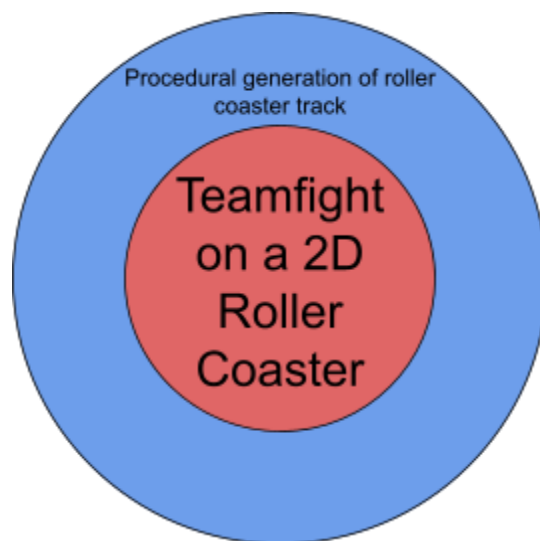
The technical achievement that we have decided to implement into our game is procedural generation. Procedural generation is a computer algorithm that can automatically create content by using a set of pre-written rules. It can either generate everything from scratch or combine hand-crafted assets in an indefinite loop. In our case, we will use this to generate the roller coaster track while playing the game.

The game starts with a set of basic coaster segments, such as straight sections, curves and loops. These segments have different properties, such as length, angle, and direction. Obstacles, islands and bases are added sporadically as well. The game randomly selects segments from this set and combines them to create the track for the roller coaster. For example, the game may start with a straight section, then add a curve, then a loop, and so on.

To ensure that the track is always navigable and doesn't have impossible or dangerous sections, the game uses procedural constraints to guide the segment combination. For example, the game ensures that the coaster always has a certain minimum height, a maximum drop angle, and a minimum radius for each curve.

To make each play through unique, the game uses a seed value to generate a random number sequence. This sequence determines the order and selection of the coaster segments.

3. Big Idea, “Bulls-eye”



The core of our game combines fast-paced item collecting with fighting, where players are separated into two teams and have to obtain a number of collectables higher than the opposite team. All of this while being on top of a roller coaster. The players can push each other off the rollercoaster to make their opponents lose the collectables they have obtained before going to their base to drop off their own.

Our main technical achievement will be procedural roller coaster generation. To begin with, we make a small segment of the track and spawn it multiple times outside the view of the player. We merge those segments to form a track. Depending on the position and rotation of the segments, it can be made to look

like the track is going up or down. The position of the segment can be decided based on a noise pattern (Perlin noise for example) or a combination of different noise patterns. The rotation can be adjusted according to the position of the previous segment. We can also make large custom segments of the track and spawn it, for example - a looping track.

4. Development Schedule

Task List

Artwork

- Characters design
- Background images
- Character animation
- Intro animation
- Ending animation
- Menu design

- Menu background
- Head up display artwork
- Bases
- Sound
 - Music and SFX
- Compose Music
- Select / create SFX

Programming

- Character controlling
 - Movement (moving, jumping)
 - Interaction with collectibles (pick up, drop at base)
 - Interaction with other players (colliding, punching)
 - Dying and Respawn
- Roller coaster
 - Roller coaster movement
 - Track segmentation
- Effect of roller coaster on players
- Procedural generation of track
 - Base spawning
 - Platforms spawning
- UX
 - Head up display
 - Menu implementation

Layers

FUNCTIONAL MINIMUM

- 1 vs 1
- Simple Roller Coaster
- Collect collectibles

LOW TARGET

- Bases & collectibles
- Procedurally generated track
- Simple Menu (start, exit...)
- Simple Music

DESIRABLE TARGET

- Fighting (Punch button)
- Obstacles and Islands
- Main Menu (Game Modes,...)
- Background Images

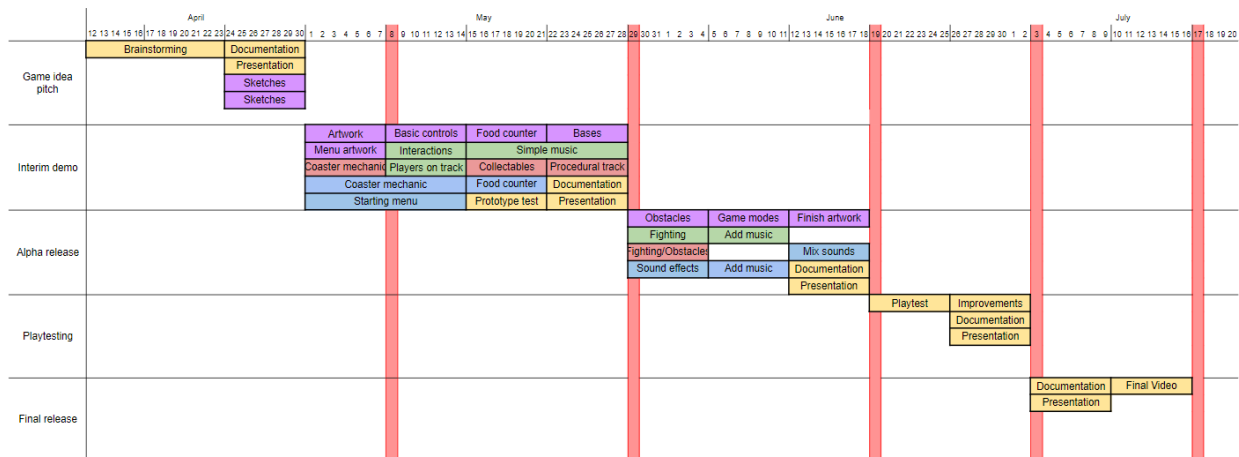
HIGH TARGET

- Animations, Sounds
- Stupid AI
- Momentum on players
- Skills, more attacks, earning collectibles in a different way
- Power ups
- Custom Music
- Custom Sound effects
- Sound mixing
- Settings (Audio, Graphics, Game Play)

EXTRAS

- AI
- Settings (Graphics, Controls)
- Easter Eggs
- Day-night cycle
- Animated procedural textures

Timeline



Milestones

1. Game idea pitch

Week	Goal	Tasks	Assigned to	Hours expected	Actual hours
Week 1: 10.04 - 16.04	Brainstorming	Research	All	5	5
		Miro Board with ideas	All	2	2
Week 2: 17.04 - 23.04	Choose an idea	Gather all ideas and decide on one	All	3	5-6
Week 3: 24.04 - 30.04	Milestone 1 documentation	Sketches	Andrea, Bende	8-9	10
		Document	All	3	5
		Presentation	Ankur, Mischa	5	4

2. Interim demo

Week	Goal	Tasks	Assigned to	Hours expected	Actual hours
Week 4: 01.05 - 07.05	Artwork	Basic character & collectables	Andrea	10-15	12
		Basic rollercoaster	Andrea	5	2
		Simple menu artwork	Andrea	3	3
	Roller coaster mechanics	Moving roller coaster	Ankur, Mischa	10-12	11
	Simple Menu	Start of game	Mischa	2	2
Week 5: 08.05 - 14.05	Character mechanics	Basic controls	Andrea	10-15	12
		Interaction of characters	Bende	15	14.5
		Characters on roller coaster	Bende	12	10
Week 6: 15.05 - 21.05	Collectables	Availability to collect	Ankur	2	2
	HUD	Food Counter	Mischa, Andrea	6	6
	Simple music	Gameplay track	Bende	6-8	7
	Prototype	Finish and test	All	6	8
Week 7: 22.05 -	Adding new game	Bases (make	Andrea	5	6

28.05	mechanics	artwork)			
		Procedural track	Ankur	10-12	14
	Simple music	Menu track, Game over track	Bende	6-8	10
	Milestone 2 documentation	Document	All	4	5
		Presentation	All	5	5

3. Alpha release

Week	Goal	Tasks	Assigned to	Hours expected	Actual hours
Week 8: 29.05 - 04.06	Desirable Target mechanics	Fighting	Ankur, Bende	5-6	
		Obstacles	Ankur, Andrea	10-15	
		Add sound effects	Mischa	8	
Week 9: 05.06 - 11.06	Main menu	Artwork and different game modes	Andrea	5	
	Sound	Add Music	Mischa, Bende	2	
Week 10: 12.06 - 18.06	Background artwork	Finish other artwork missing	Andrea	10	
	Sound	Mix Sound	Mischa	3	
	Milestone 3 documentation	Presentation	All	5	
		Document	All	4	

4. Playtesting

Week	Goal	Tasks	Assigned to	Hours expected	Actual hours
Week 11: 19.06 - 25.06	Playtest	Prepare playtest sessions	All	10	
		Arrange sessions	All	20	
Week 12: 26.06 - 02.07	Add improvements	Feedback from playtesting and improvements	All	10-15	
	Milestone 4 documentation	Presentation	All	5	
		Document	All	4	

5. Final release

Week	Goal	Tasks	Assigned to	Hours expected	Actual hours
Week 13: 03.07 - 09.07	Milestone 5 documentation	Document	All	6	
		Demo presentation	All	5	
Week 14: 10.07 - 16.07	Final video	Prepare video and editing	All	10	

5. Assessment

The major strength of the game is combining a fast-paced combat system with the collecting aspect and the platformer aspect. The fighting is simple and therefore easy to learn, letting even newcomers enjoy the game from the start. They can be further helped by a more experienced team member. The aim of the game is also easily understandable, as games revolving around collecting and dropping off items have been around for a long time, such as capturing the flag modes or escort missions. The procedural generation will keep even experienced players on their toes, as they not only need to defend themselves from other players but also keep an eye on the track and for possible obstacles lest they fall off. There is a strategic part of the game as well because of the collection of the items as a team and registering them at their respective bases, which only spawn alternatively.

The game is inspired by games such as Super Smash Bros., Super Mario Bros., Brawlhalla or Stick Fight: the Game. In such games, tight controls and natural feeling movement are the most important aspects, thus these are the criteria playing the biggest role in our game. Another point is making the procedural generation suitable for such a control scheme, so it compliments the fact that the players are restricted to simple movements.

Milestone 2: Interim Demo

Fall For Me!

Team Cicisoft

Andrea Solanas de Vicente

Ankur Deria

Michael Dey

Bendegúz Timár

Critiques

In this section, we reflect on the critiques from other students.

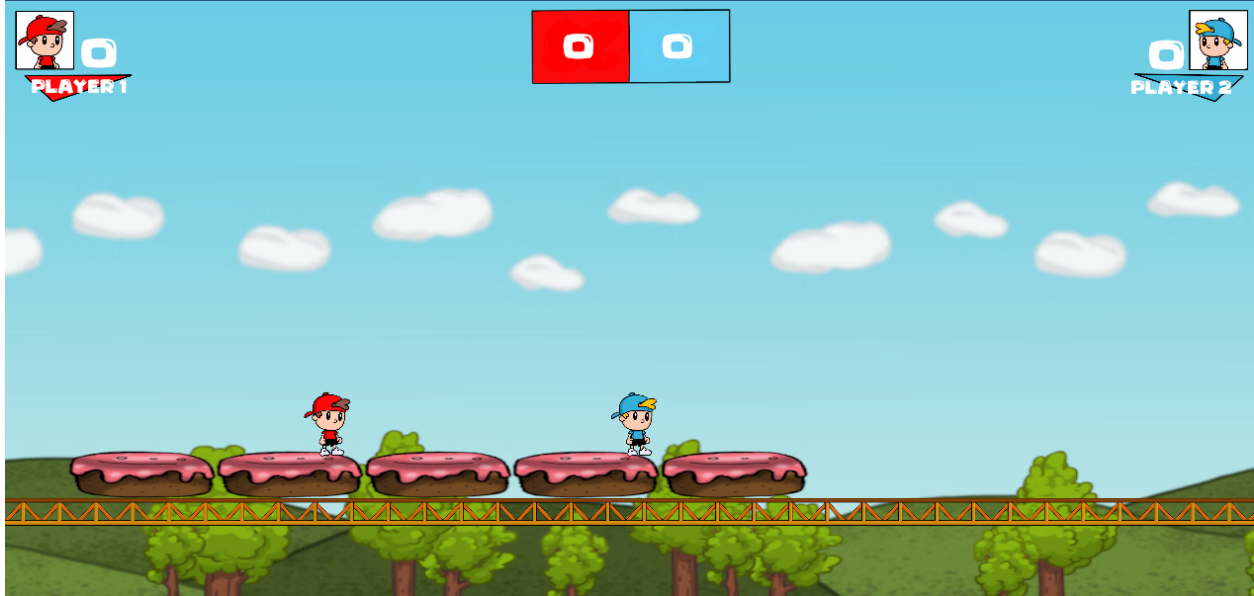
Critique	Reflection
Frustration when killed	Timeout is the only reward for pushing off. It would be frustrating to have opponents come back too fast. In the future, we can do a hanging mechanic where the other teammate has to help the player come up to revive. → This is not set in stone → Possibility: Respawn at any base
Better fighting mechanics	The game's focus is on fighting and balancing on the roller coaster. → Focus may shift towards fighting later
Team Combo in fighting	This is a high target. Simple combos are possible because of the current implementation.
NPCs	This is (now) part of the extras

Task progress

Our goal for the interim demo was to produce a first working version of our game that implements most of the main gameplay of the final game. In order to reach this goal, we first wanted to implement the mechanics of the rollercoaster, the gameplay and controls of the characters, add the collectibles and a basic UI.

Basic UI design

For the UI design, we decided to base our ideas on the video games we mentioned in the previous report. There should be a main menu where the players can choose the mode of the game (1v1 or 2v2). For this report, we were not able to add a way for the players to choose their teams just yet, but that will be the next improvement. After the main menu, the game will start with the number of players that have been chosen. Each player has a specific counter where their candy will be shown. Then there is an overall counter for both teams. This counter will be updated when the candy is dropped off at the bases.



For the main menu, we created a simple logo with the name of the game. The player can then choose between the two options of playing, as explained earlier.



Background

The backgrounds are static but their UVs are offset with respect to time and the game speed to make it seem like they are moving from right to left. The backgrounds are separated into different objects, for example, the mountains, clouds and trees. This is done so that they can move at different speeds to give the viewer a parallax effect. The current background is not the finalized one, as we plan to make it look like a theme park by adding a few amusement park attractions. We may even add simple animations to these attractions. Given enough time we

might make these attractions interactable, for example having a ferris wheel close to the roller coaster serving as additional platforms for a short amount of time.

Handling player input

The player input is processed using Unity's new input system. The basic idea behind the system flow is incorporating so-called Action Assets, a mapping of input events to event handler groups called actions. These actions are totally arbitrary, and only have semantic meaning to the developer. Two Action Maps are created, one for the gameplay and one for the menu; this enables registering different handlers in different scenarios for the same input events. The Action Assets are wired into Unity's Player Input component, which automatically creates and exposes an interface of the related action handlers to be implemented in code.

Since both movement, punching and jumping are physics-based, the event handler methods only set action flags for the respective player. These flags are then checked in the FixedUpdate method to ensure that movement physics is accurate and calculated in a fixed timerate. There, punching and jumping is implemented using impulse forces, while movement is a direct smoothed modification of velocity.

Character animation

The animation of movement is done using sprite sheets and Unity's Animator component. We drew a number of frames for all the types of movement, and the animations themselves are rotating these frames around. We created a central Animator component that is the same for each player, defining the animation states and the transitions between them. To be able to use different sprite sheets for the different characters, each character type has its own Animator Override Controller. This component is coupled onto the central Animator, and redefines the animations used in the respective states to achieve different looks for each character.

Collectibles

The collectibles are spawned along the track every time a new track segment is added with a 70% chance of spawning. The speed of the collectibles is tied to the track, i.e., if the track moves fast, so will the collectibles. On top of this, the collectibles can have a movement curve independent of the track, for example falling from the sky or levitating along a sinus curve. Spawning in a new object every time would be a huge waste of resources. Instead, an object pool of collectibles is used, where the newly activated object gets its translation set. This is an elegant solution, since we control the spawning rate of the collectibles, and therefore can give an estimated upper limit for the size of the object pool. The players interact with the collectibles based on simple 2D colliders. The colliders are set to be only triggers, which shoot events related to entering or leaving the collider's area, but have no physical consequences, such as pushing the player away when they touch the candy.

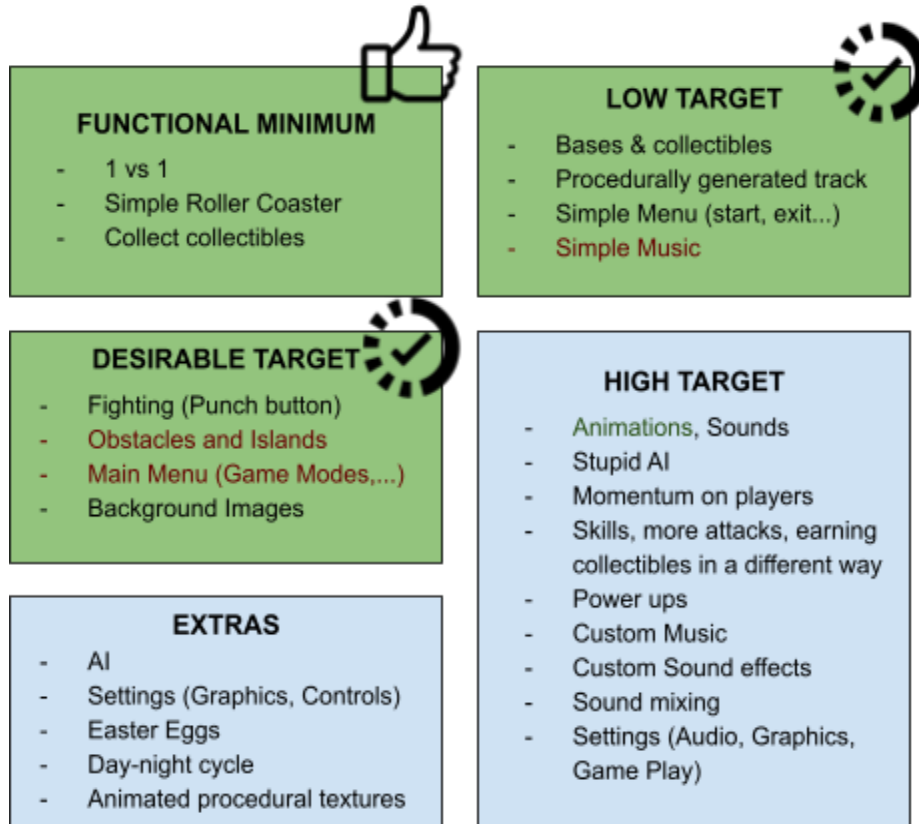


Procedural generation of the roller coaster track

For the procedural generation of the track, we used cubic bezier splines. When the left-most point of the spline leaves the screen, a new point is added to a random location on the right of the spline. The distance between the points remains constant as the random point is sampled from a unit sphere's arc (angle constraint). The curvature of the track can be adjusted using the tangents of the spline. The entire track moves from right to left and the Y position is adjusted so that the track always stays at the center of the screen irrespective of if it's going up or down.

Layers

With all these accomplishments, we were able to achieve our functional minimum as well as parts of the low target and desirable target. We decided that some elements of the desirable target were more important than others in the low target, like the music and some artistic elements. The mechanics of the game were prevalent over these aspects.



Problems

Jittering Cars

The first method we tried was to animate the roller coaster using the spline offset value from the nearest point on the spline. But as the spline was being changed repeatedly, the offset value was fluctuating as well, resulting in the jittering of the roller coaster.

The next method we tried was adding an edge collider along the spline. This removed the jittering and the roller coaster could travel smoothly along the spline now. But once in a while, the cars would fall off due to the limitations of Unity's physics engine.

Finally, we solved it by taking into account the changed spline and also the previously calculated nearest point on Spline was in local space. So now that we have the nearest point on the spline, we can align the roller coaster car to it using its tangent and normal vectors. Animating the roller coaster this way not only made it travel smoothly along the spline but also ensured it would never fall off the track. We also add a force along the direction of the movement to make it look more dynamic.

Gravity on the roller coaster

Once the “nearest point on the spline” method worked, the roller coaster moved with constant speed. This was not realistic.

To address this issue, we initially attempted to reintroduce Rigidbodies and gravity into the simulation for a more realistic effect. However, we soon realized that this approach was overly complicated and didn't yield the desired results.

Our final solution is quite elegant. We sum up the angles of the forward vectors of all cars against the horizon. If the sum is positive, the cars on average travel upwards. Based on this value, we made adjustments solely to the background speed and not the rigidbodies. This approach resulted in a more realistic physics simulation, especially when considering that all cars in the roller coaster have the same mass, which is an assumption we made for simplicity.

Future work

Implementing different controlling schemes is definitely on the roadmap. In the main gameplay scene keyboard and controller input will need to be handled at the same time, since we have to imagine that some of the players will be playing with the keyboard while others with separate controllers. Additionally, the keyboard may be set up to support two players simultaneously. This is allowed by the very simple controls, and advised in order to achieve a more accessible user experience. The menu will need to be controlled by either controllers or keyboard. Currently the menu only supports mouse input. Instead of this, the menu will need to be able to change between being controlled by keyboard or controller. Furthermore, hint texts of the controls of the menu may be added onto the menu scene, which texts should follow the currently active input scheme.

In the future, we want to introduce a drag force on the players as they jump on the roller coaster, based on the speed of the roller coaster. This would limit the players jumping around too much and make them only jump when needed.

At the moment, our base spawning mechanic is not very stable. Players do not have much time to stay on the base. We plan to address this issue by changing the base spawning mechanic so the base moves along with the roller coaster for a fixed time, after which it goes away. This would give the players enough time to get on the base, register their collectibles and get back on the roller coaster without slowing down the roller coaster.

We will need to thoroughly playtest character fighting to fine tune the parameters and see what works best, and possibly come up with some combo ideas. We may add a slight upward force to the punches, which would allow the players to do a combo on an enemy. The upward force would stack if the enemy gets punched repeatedly.

A 2 versus 2 players mode is planned to be added. Adding such a mode is a bit more complicated than just spawning in two more players. We will need to fine-tune the spawning rate

of the collectibles to balance out the fact that there are more players in one team, and maybe the length of the roller coaster too.

Work log

Week	Goal	Tasks	Assigned to	Hours expected	Actual hours
Week 4: 01.05 - 07.05	Artwork	Basic character & collectables	Andrea	10-15	12
		Basic rollercoaster	Andrea	5	2
		Simple menu artwork	Andrea	3	3
	Roller coaster mechanics	Moving roller coaster	Ankur, Mischa	10-12	11
	Simple Menu	Start of the game	Mischa	2	2
Week 5: 08.05 - 14.05	Character mechanics	Basic controls	Andrea	10-15	12
		Interaction of characters	Bende	15	14.5
		Characters on roller coaster	Bende	12	10
Week 6: 15.05 - 21.05	Collectables	Availability to collect	Ankur	2	2
	HUD	Food Counter	Mischa, Andrea	6	6
	Simple music	Gameplay track	Bende	6-8	7
	Prototype	Finish and test	All	6	8
Week 7: 22.05 - 28.05	Adding new game mechanics	Bases (make artwork)	Andrea	5	6
		Procedural track	Ankur	10-12	14
		Fighting mechanics	Ankur, Bende	4-5	4
	Simple music	Menu track, Game over track	Bende	6-8	10
	Milestone 2 documentation	Document	All	4	5
		Presentation	All	5	5

Milestone 3: Alpha Release

Fall For Me!

Team Cicisoft

Andrea Solanas de Vicente

Ankur Deria

Michael Dey

Bendegúz Timár

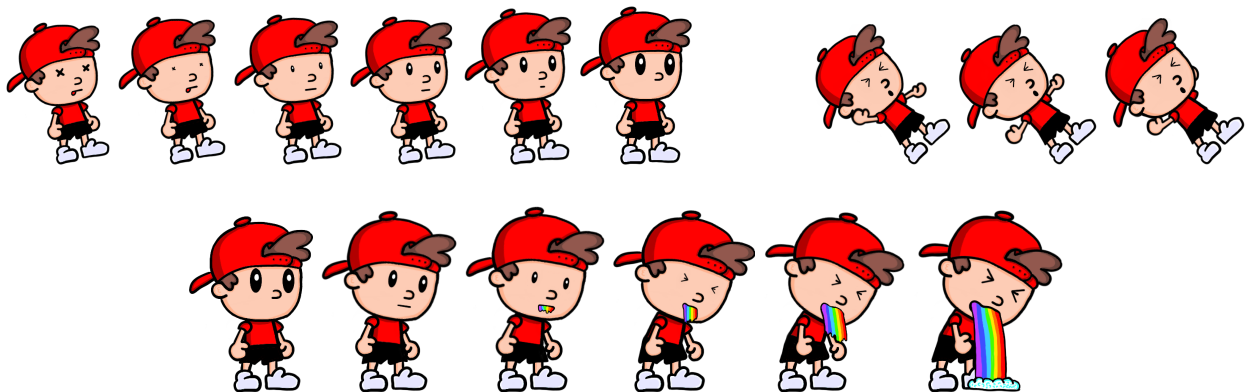
Task progress

Our goal for the alpha release was to produce a finished working version of our game that implements the main gameplay and some extra features. In order to reach this goal, we first wanted to add some UI elements like menus, and character animations, also we added some form of interaction for the characters so they could choose their players and pick the mode of the game. Finally we added the multiplayer functionality so the game could be played as 2 versus 2.

Animations

New character animations

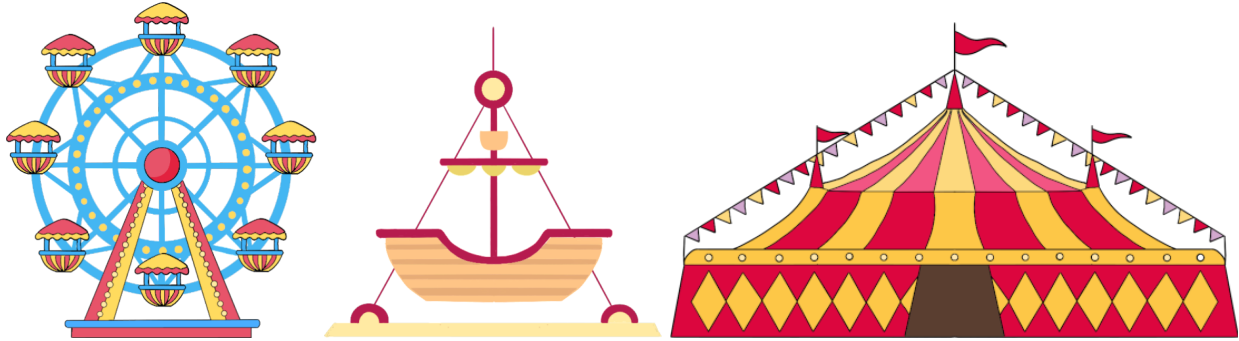
We added a few new character animations: falling, getting hit and vomiting. Below are some of the frames for the main characters.



We also created 2 brand new characters and designed the same animations for them. The animations for all the characters are very similar, with the exception that one set of characters wears a cap and the others don't. This way the players know which team they are on and their characters are easy to tell apart. The new players are shown further in the document.

Background changed

For the background we wanted to add some elements that would represent the environment of an amusement park, since for the previous demo we went with a very simple background. For this milestone we wanted to make sure that the players knew or could see that they were in an amusement park. To accomplish this we created amusement park rides with their own animations to set them in the background. Some examples: a ferris wheel that turns in circles, a circus tent with moving flags or a ship that sways back and forth.



The background now looks more vibrant and it makes the player feel as if they were on a roller coaster in an amusement park. In the following image, it can be seen how the background looks now. In the future we would like to make the animations more fluid and add more frames to make them better.

Candy animations and effects

The candy collides with players, bases and with other collectibles. We added a particle system to make the candy more visible for the players. Whenever candy is hit by a player, a fireworks particle is activated with other post processing effects. When it is hit by other objects, an explosion particle is activated and the candy disappears. We added more types of movement to the candy like it would follow along the track while doing a sine wave motion in the y axis. Similarly there is a cosine wave motion, a loop motion, etc.



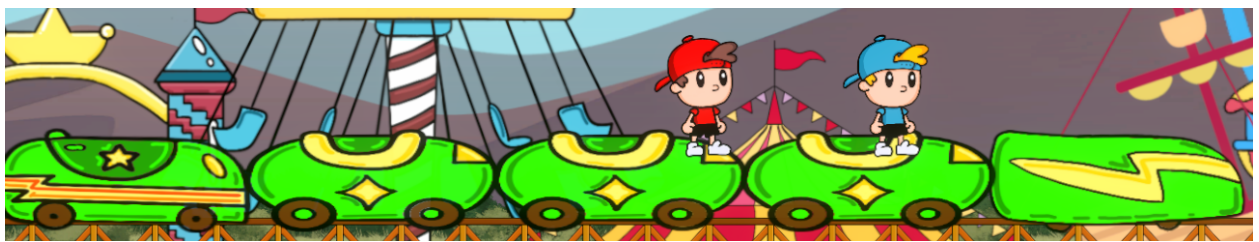
Balloons popping

For the bases which will be explained in more depth later, we added a set of balloons that symbolize how much time the bases will be available for the characters. There will be a number of balloons that will slowly pop and when the last balloon pops the base will fall out of the screen. For this we created the animations for the two teams' bases and their balloons popping.



Roller coaster sprite

For the previous milestone we didn't have the final icon for the roller coaster yet. But for this milestone we were able to create three different icons for the roller coaster to use as the front wagon, the ones in the middle and the last wagon. We also made two different styles (orange and green) so it could be randomized at the beginning of the game and the roller coaster was a different color for each time.



Visual effects

Mainly particle systems, and post processing effects like depth of field, lens distortion, chromatic aberration, etc. We also use timeScale to slow down for a few seconds to give a cinematic effect. 3rd party assets from Unity Asset Store were used for the particle systems, whereas some were made by us.

Music and sound effects

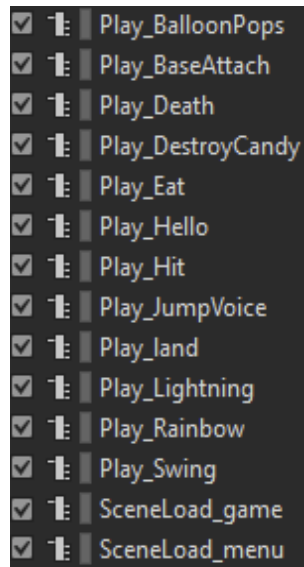
Music

In this milestone we finished the main gameplay background music. It starts with an instrumental part, which we will use in a loop for the controls tutorial at the beginning of the game. At the first chorus the song changes to a kind of EDM or electro dance pop style, and the later verses stay electronic with a very similar melody as the first instrumental part. This part will be looped for the remainder of each round. The whole song is about 3 minutes long, which should be enough considering the rounds with a time limit will last 5 minutes according to plans.

We also started working on the main menu music. This song will be much shorter and with a consistent electro pop style. This song is in its very early phases, but we suspect about half a minute will be long enough.

For composing the music we used FL Studio, with which one of the team members already had experience. Even though composing the main gameplay music took long, we plan that finishing the main menu's theme will take a much shorter amount of time, given the experience we got while working on the gameplay music.

Sound Design



The sound is designed to be fun and happy like the artstyle. The Rollercoasters sound depends on the velocity of the cars. There is a pulling sound, when the minimum speed is achieved and cheering at the first drop after each pulling up.

Before the new visual effects were applied to the candy the eating sound was someone biting an apple. We changed that to bubbles bursting and bubbling, to reflect the visual effect.

On respawn the visual effect will be a lightning strike. The sound for this is already in the game.

The rainbow sound is already finished, but it doesn't start properly when run from the code. It is inspired by this [video](#).

One sound that is still missing is footsteps.

Bases

The original design of the bases proved to be too rigid for such a vibrant environment. They had a static world position, which also made accessing and leaving the bases way too hard, since the players barely had any time to spend on the base's surface if they wanted to reach the roller coaster.

We came up with a new design, where the bases are hung on balloons. Using balloons allowed the idea that the bases can follow the roller coaster for a short amount of time. The base floats in with a number of balloons attached to it. The balloons pop after some time one after the other, and intuitively fall down after all the balloons popped. This makes reading the amount of time the players can spend on the bases really easy, and also adds to the overall vibe and atmosphere of the game. Players can now respawn at their respective bases after they fall off the rollercoaster as well.



Menus

For designing the UI of the menus we decided to work with the new UI system of Unity called UI Toolkit. The UI Toolkit ditches the old GameObject-based UI of uGUI (Unity's old UI system), where elements of the UI are put on a Canvas Object. Instead, it organizes UI layouts into so-called .uxml files, which are simple .xml files and follow the principles of web UI design, specifically the FlexBox structure. Attributes of layout elements can be collected in .uss files, which work similarly to .css files. The user needs to define handles to the different layout elements. The names of the handle only have semantic meaning to the user, meaning a button handle can be put on a label or a panel too.

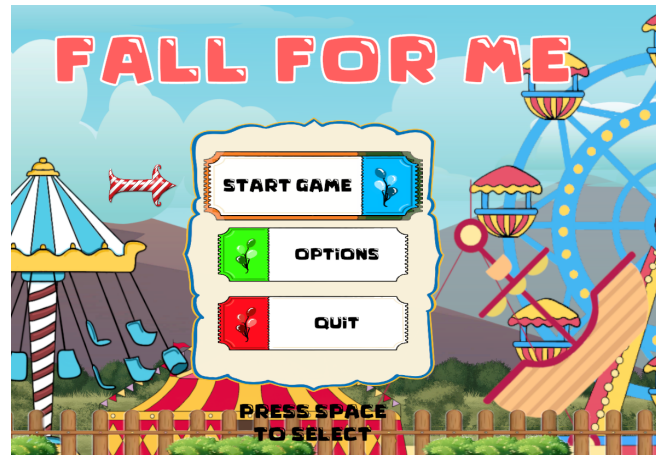
Separating the layout and the logic of the UI is considered good practice in every front-end project, as it enables building and testing the two areas in parallel. Not to mention that the programmer doesn't have to redo the whole UI after each change made by the artist. However, composing a web-based UI has its limitations, as flex attributes affect one another. This means that simple UIs can be achieved with relative ease, but more complex structures can be and were a head-scratcher for us. The bigger problem however was the user intractability of this new system. Even though the UI Toolkit has been out for more than 2 years now, tutorials on the subject, especially controlling with multiple gamepads, are not of the highest quality, and the documentation of the Toolkit can be very confusing. Since the layout and the logic is separated, working with the layout elements is a bit harder on the logic side. Also, the new UI Toolkit has a built-in input handling system, but it is still not the most usable for gamepad inputs.

Since we were losing a lot of time on the subject, we decided to make the new menus for this milestone with the old uGUI system. The gameplay UI still uses the new UI Toolkit. In all of the menus, the background is blurred out but still playing the gameplay's background. This adds a welcome vibrancy to the menus, and makes starting the game more seamless.

Main menu

The main menu has a very simple layout, consisting of 3 buttons - Start, Options and Quit. Intuitively, the options button leads to the options menu, while the quit button exits the application. At the moment we do not have any settings in the options menu and we plan to add

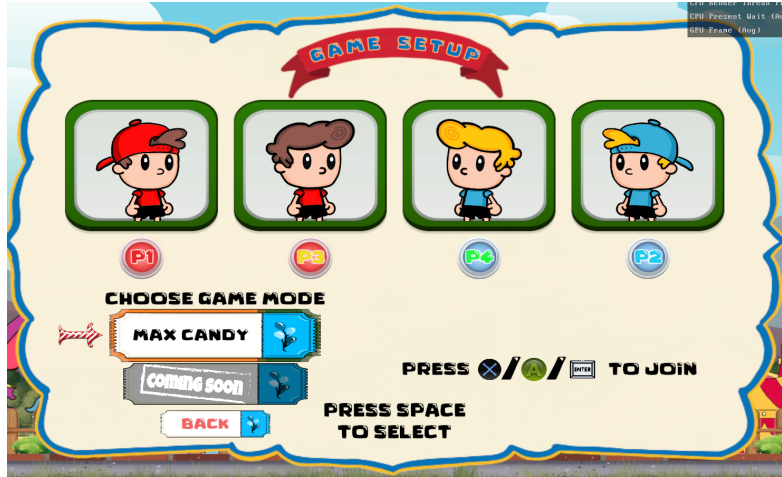
this during the next milestone so the players can modify the settings in the game. The start button leads to choosing the characters and the game setup screen.



Game setup menu

The players can get to this menu by clicking the Start button in the Main menu. On this screen the players can choose their character and the gamemode of the game. The characters only have cosmetic differences, and each handles in the exact same way. We currently plan two different game modes: a time-limited mode which runs until 5 minutes, and the team who collected and deposited the most candies in this interval, wins; and a collector mode, where the teams aim to collect a number of candies, and who first deposits this amount, wins. The time mode has not been implemented yet but we plan to have it for the next milestone and have it during the play testing for the users to try it out.

For this menu we planned to include a cursor for each player, similar as to that in the Super Smash Bros. character selection screen. On connecting a controller a cursor for the new player would be spawned, and disconnecting the controller would remove the appropriate cursor. The player could hover their cursor over their selected character, and click to select it. This solution would efficiently show the number of active players, and would make navigating this menu more fun. However, this is where we decided to roll back to the old uGUI, since achieving this behavior with the UI Toolkit proved to be unnecessarily overcomplicated. While controlling a cursor in the uGUI is flawlessly working, spawning in the cursor is still unstable, and therefore we removed it from this release. The current solution uses only a navigation flow between the elements of the UI, where the players can select the different choices similar to the Main menu.



Pause menu

To access this menu players can click Esc (keyboard) or select (controller) during the game and pause the gameplay. This gives the players control to take a break from the game. In the pause menu there are three buttons Resume, How to play and Exit. If a player clicks on resume, the game will resume from where it left off when the players paused. If the player clicks on How to play, the controls for the game will show to inform the players how to play. If the Exit button is selected the gameplay will end and the main menu will show like at the start of the game.



Tutorial menu

For playtesting we added a simple tutorial for the players to get the hang of the game and its controls. The tutorial menu will show when the player is playing the game for the first time. If the gameplay is restarted it will not be shown anymore.

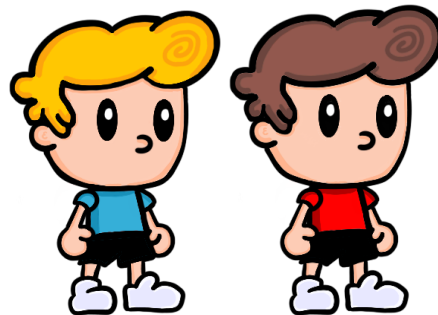
The menu is very simple and it only consists of a few pages with the basics of the game, how to play, how to win and the controls among others.



Multiplayer Functionality

New players

We created 2 new players for the two teams, with the same design except for the caps. The new characters have their own candy counter in the gameplay UI but they have the same controls as the other characters. We did the animations fairly similarly so that the players could switch back and forth from the different characters easily.



How it works

Players can join in the game setup menu. They need to press a button to join, 1 player per device. They can choose any of the characters they want (if its not already chosen). The players get carried over to the gameplay scene after that. Each device is bound to the player and that player character only accepts input from that device and ignores all others.

Winning conditions

As previously mentioned we plan to add two different winning conditions to the game: either reaching a specific amount of points first, or getting the most points in a given time. For this release we only implemented the former: the players need to collect and deposit 50 candies in total, at which point they win the game.

Implementing a time-based gameplay would require some more logic, such as designing and implementing a clock in the main gameplay UI, and tackling scenarios where both teams have

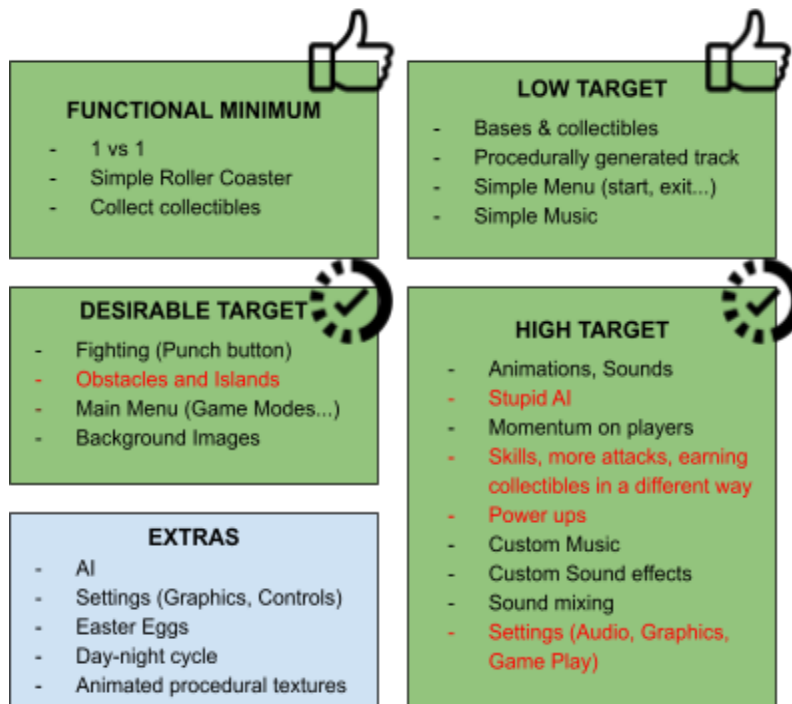
the same amount of points when the timer runs out. Because of this, we decided to include this condition in a later release.



After having it implemented, the players can choose between the different game modes in the Game Setup menu. In the future, we also may include other game modes, or a custom mode where the player can enter their own time- or point-limit.

Layers

In comparison with the previous milestone we have been able to achieve the low target in its entirety plus doing most of the desirable target. The only thing we were not able to achieve was the obstacles. We plan to add them for playtesting so the users can test them and give us their feedback. We were also able to achieve some of the goals from the high target and plan to achieve some more which is explained in future work.



Problems

Meta files in gitignore

To avoid the recurring merge conflicts in hard-to-decipher scene files we introduced more prefabs. One of our team members had included Unity meta files in gitignore. Meta files store the hash ID's of all objects in Unity and as a result the new prefabs weren't synced well over git.

Problems between using both old and new UI

Old UI has a vast amount of documentation compared to the new UI but the new UI offers much better development with html and css like framework. It has native support for a lot of features that make scaling for different aspect ratios a walk in the park. But due to the lack of much documentation we faced a lot of problems with it and eventually decided to fall back to the old UI system.

There were some problems with mixing both of the UI types since the text fonts and some of the images were messed up when using both at the same time. We had to fix this problem multiple times which became very tedious.

Choosing characters and going into gameplay

We had some trouble carrying over the spawned player to another scene without copying all the classes in it. This was necessary when choosing the characters in the setup menu and creating the gameplay scene altogether.

Integrating Wwise into Unity

There was an unexpected Error, when opening Unity after integrating Wwise. This Error also happened, when retrying the integration with a fresh Unity project. It seems to be connected to recent changes in unity because there were no discussions about it on the internet. We were able to solve the problem on our own though.

Future work

As we mentioned we plan to add a few ideas that we were not able to include in this release, for example the game mode of using time as a winning condition and maybe obstacles for the players. The animations will also be worked on to make them more fluid.

We want to add some more effects for the characters so they really stand out. We also have as part of our high target making an AI so players can play the game as 1 or as 3 and play with or against the AI. Another thing we wanted to add is for the player to be able to modify the settings of the game.

If we have enough time we would love to include an easter egg or maybe a day and night cycle for the gameplay.

Work log

Week	Goal	Tasks	Assigned to	Hours expected	Actual hours
Week 8: 29.05 - 04.06	Desirable Target mechanics	Fighting	Ankur, Bende	5-6	6
		Obstacles	Ankur, Andrea	10-15	Not done
		Add sound effects	Mischa	8	8
Week 9: 05.06 - 11.06	Main menu	Artwork and different game modes	Andrea	5	10
	Sound	Add Music	Mischa, Bende	2	4
Week 10: 12.06 - 18.06	Background artwork	Finish other artwork missing	Andrea	10	15
	Sound	Mix Sound	Mischa	3	3
	Milestone 3 documentation	Presentation	All	5	4
		Document	All	4	4

Milestone 4: Playtesting Fall For Me!

Team Cicisoft

Andrea Solanas de Vicente

Ankur Deria

Michael Dey

Bendegúz Timár

Task progress

In this chapter, we will describe the results from our playtesting sessions. We will explain the results we obtained, as well as the type of testers we had and how the sessions were organized. Finally we will show the questions and answers we received and what changes we made based on the results.

Testers and sessions

Testers:

Names	Scale of 1 to 5 (1 they don't play video games and 5 they play a lot of video games)
Male 24	5
Male 24	5
Female 23	3
Male 26	4
Male 23	5
Male 24	1
Female 22	1
Male 24	5
Female 24	1
Male 25	1
Male 24	1
Female 23	2

We had a broad variety of testers, most of them were close friends of members of the group. Some of them play video games regularly and others have no knowledge of video games or don't play as often. This was great for us because we had different points of views about the game.

For the sessions we created a google form with questions about the different areas and parts of the game, so the testers could answer them after playing the game. This way we have all the answers together and we can compare what each of the testers said after testing the game.

The sessions were organized individually or in pairs. One member of the group would meet up with 1 or 2 testers and play the game. Sometimes the members of the group would play with the tester and in other cases the testers would play with each other. In the case where one member of the group would play with the tester, we would let the tester figure out the game first and play easily in the beginning. This is because we have had a lot of time practicing and playing the game and it gave us an unfair advantage over the testers. During the sessions, we would let the testers play all the different areas of the game and play as often as they wanted. We did not answer any questions while they were playing but would write down some of the comments they would make. Also, if we were to see something go wrong we would write it down to fix it afterwards. After the sessions we would conduct a small interview out loud just discussing the things that the testers would mention. Finally we would ask them to fill out the aforementioned form.

We tested the game both with the keyboard controls and using our personal gamepads. Since the game can be played using both, we wanted to make sure all of the different ways to play worked.



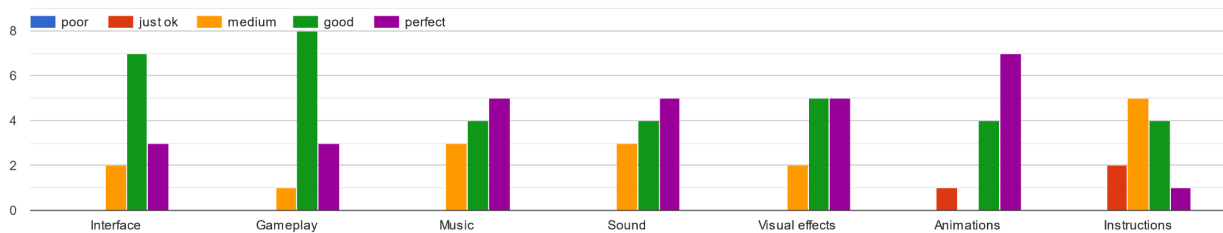
Questions

Here is the list of questions we had in the form that the testers would fill in after testing out game:

- What was your first impression?
- Was the game's premise exciting?
- How did that impression change as you played / What surprised you?
- What did you like least?
- What did you like most?
- Did the game drag at any point?
- If you could change just one thing, what would it be?
- Describe the objective of the game.
- Was the objective clear at all times?
- What types of choices did you make during the game?
- What was the most important decision you made?
- What was your strategy for trying to win?
- Did you find any loopholes in the system?
- How would you describe the fighting?
- Rate these elements (of game)

- How did the controls feel? Did they make sense?
- Were the procedures and rules easy to understand?
- Could you find the information you needed on the interface?
- Was there anything about the interface you would change?
- Did anything feel clunky or awkward?
- Are there any controls or interface features you would like to see added?
- Please describe sound effects that were too loud / not loud enough.
- Which Sound effects were missing?
- Was any sound effect off eg. it didn't fit the visuals or style?
- Overall, how much fun did you have?
- How much do you want to play again?
- How likely is it that you would purchase the game?
- If you were to give this game as a gift, who would you give it to?
- What elements of the game attracted your attention?
- What new features would you add to the game

Rate these elements



Results

Based on the answers we collected we came to the following results.

The game is praised for its cute design and straightforward controls. It offers a nice and fun experience with smooth gameplay, although some players feel the spawn times are too high. Overall, players are amazed by its interactivity and find it to be a very fun and interesting game.

Opinions regarding whether the game surprised the players or not varied. Some players felt that the game stayed the same and met their expectations. However, others were pleasantly surprised by certain aspects; namely depositing the collectibles was a part favored by most. The interaction and gameplay were described as simple and enjoyable, with fun animation and fighting.

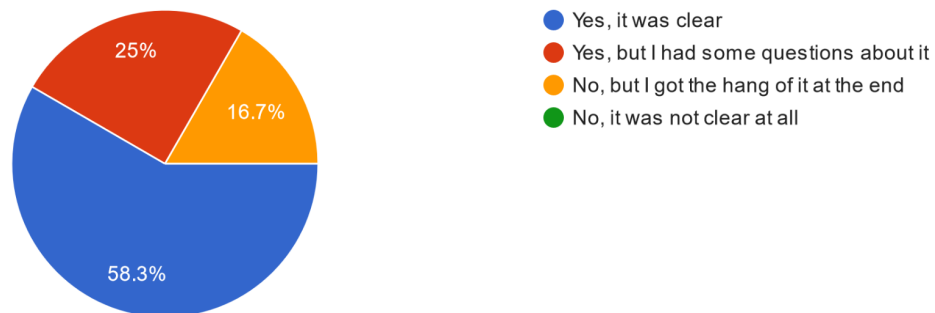
The aspects that the players liked the least about the game can be summarized as follows: The duration of the game was considered too long which affected the gameplay experience. Some instructions were difficult to read due to blending with the background, causing confusion. Additionally, players found the spawn times and the amount of candy to be unsatisfactory. The presentation of all instructions at once at the beginning was overwhelming for some players. The controls were also a source of dissatisfaction, specifically some players experienced lag

before jumping. It was also mentioned that the game was in 2D. Finally, some players struggled with getting onto the platform in front of the train, leading to frustration.

Players provided various suggestions for changes they would make to the game. These include decreasing the required candy amount to win (around 20 or 25) and improving the character's time response for smoother actions. They also suggested using balloon assets for spawning instead of platforms. Additionally, adjusting the spawn time based on team disadvantage and introducing more candy and NPC opponents were proposed. Some players recommended incorporating a walkthrough of instructions within the game, with a "back" button for convenience.

Was the objective clear at all times?

12 responses



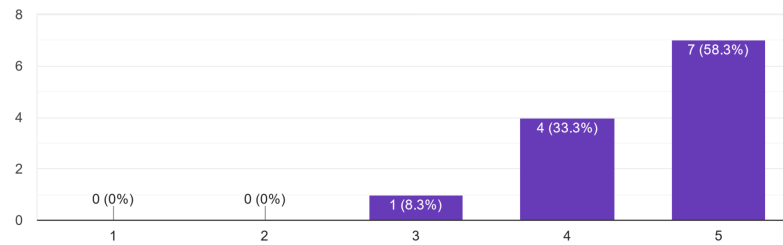
When asked about the clarity of the objective of the game, most players were able to understand it at first while others discovered the idea as they played the game. Players adopted various strategies during the game. Some focused on knocking off their opponents to create a window of opportunity to collect and save candy without interference; or to decrease their enemies' points. Others prioritized jumping frequently to avoid conflicts and stay out of trouble. This showed us that the aim of the game was well understood and people were able to adopt different strategies to win.

The controls were easy to understand and made sense for the players. Also the interface was easy to figure out and most of the players found the information they needed relatively quickly.

The opinions about the music and sound effects in the game were generally positive. Some players found the sound effects to be fine and enjoyable, particularly liking the screams of the falling children. The volume of the music and sound effects was generally considered perfect, with players stating that they were in a good volume and equally balanced. However, one player mentioned that the music stopped at some point, which was addressed and taken into account.

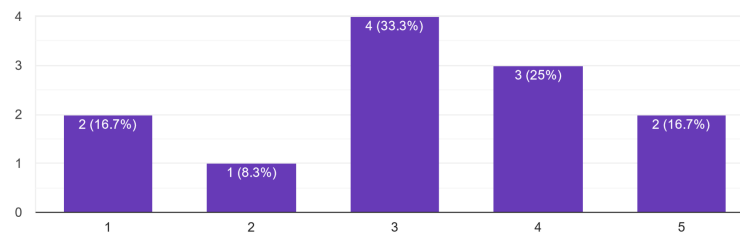
Overall, how much fun did you have?

12 responses



How likely is it that you would purchase the game?

12 responses



Overall all the players had high levels of fun throughout the game which made us very happy since it was our aim. A lot of the players showed interest in wanting to play again.

Players provided suggestions for additions they would like to see in the game which were very important for us.

- Making combat more exciting and dynamic by expanding the punching mechanic or introducing other combat abilities such as kicking
- Adding balloons for spawning instead of platforms, adding a visual element to the game.
- The introduction of different game modes, such as easy, medium, and hard,
- Additional maps
- Increasing the overall speed of the game

At the end, players left very kind and sweet messages congratulating us on our work. These were all very appreciated.

Changes made

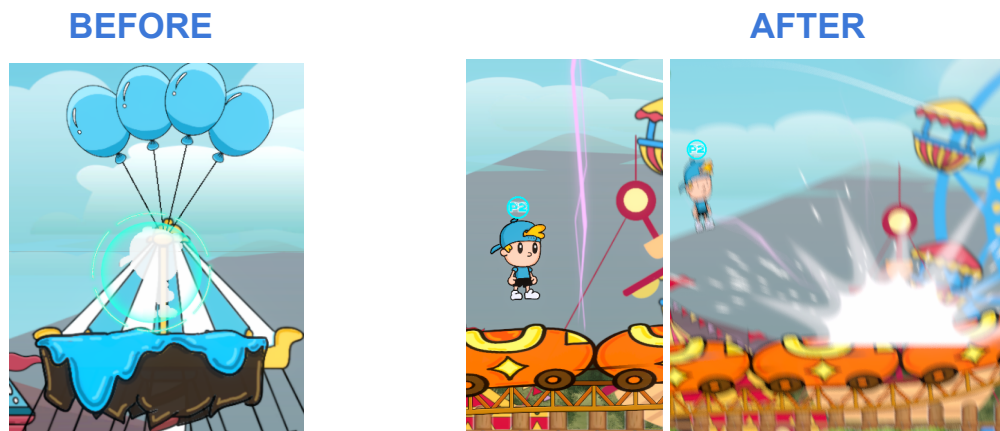
As mentioned before, the players let us know a few aspects or actions of the game that they would change or some problems that they had while testing the game. We took them all into account and implemented them as the play testing phase went on. Some of the changes that people mentioned and our implemented solutions are the following.

Time of playing

At the beginning of the playtesting we set the win condition to 50 candies. This meant that the game would not end until one of the teams would obtain 50 candies. Some players complained that this was a very long time of playing and recommended we decrease the number and the time of gameplay. We lowered the number to 25 reducing the time. Furthermore, adding the second mode of playing with a timer was warmly welcomed by the testers. The timer was set for 5 minutes which proved to be enough time to play and still enjoy the game.

Spawning

One of the most mentioned things among the players was that the respawn mechanic was too long and made the game quite boring when the players would die. They suggested multiple solutions and we decided to fix this issue in the following way. Before, the players would fall off the roller coaster and be respawned at the next base of their team color. Now the players directly spawn in the center of the roller coaster 2-5 seconds after dying. They still lose their candy if they fall but this way they come back to the game sooner and can play more. The spawning mechanic got more flashy as well and now acts as an attack as players near the lightning strike get pushed back.



Buttons to go back in the tutorial menu

We did not notice this problem until a few players mentioned it while playing. The tutorial menu only had a button to go to the next page but no button to go back in case the players wanted to check what it said again. This was not good since it meant that players who accidentally skipped the tutorial pages couldn't go back and undo their mistake. We added a second button so the players could choose between going to the next page or going to the previous page.

Another interesting aspect was the users' behavior towards the tutorial itself. We noticed that players very often skip the tutorial slides and decide to test out the game without the knowledge. They expect to learn the controls while they play or just assume that people that have previously played the game will explain the rules to them. This however led to some drawbacks for them, since we were not allowed to explain anything; resulting in forceful read-throughs of the tutorial.

In future releases, we plan to solve this problem by making the tutorial more light-weight or more interactive.

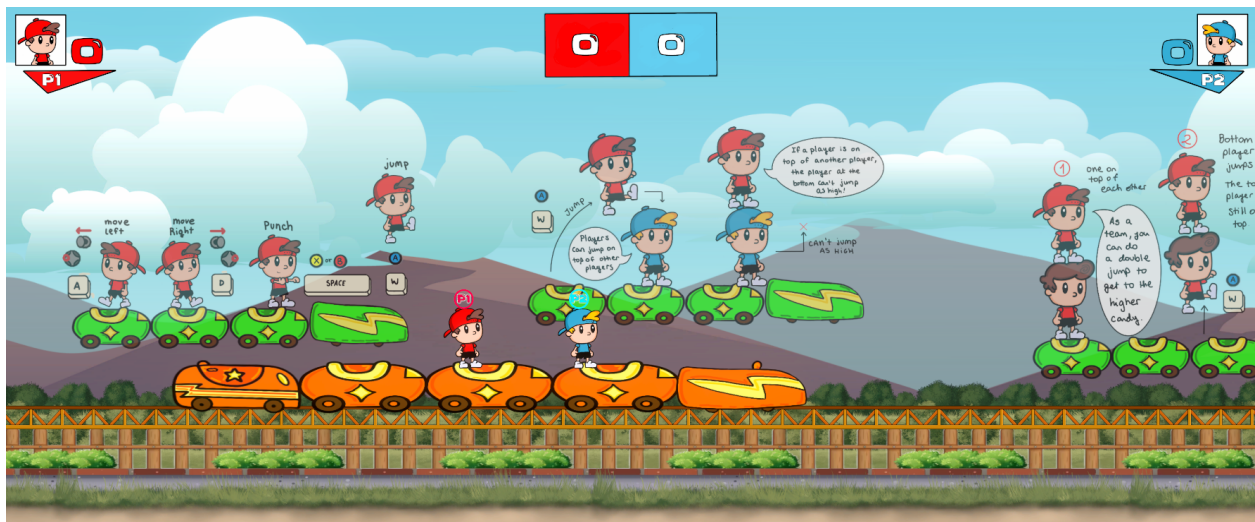
BEFORE

AFTER



Tutorial area at beginning of game

To make sure everyone understands the base mechanics and the aim of the game, we added a punishment-free zone at the beginning of the rounds. There, the players can duke it out from the beginning without worrying about falling off the roller coaster. Both sides of the rollercoaster are blocked by invisible walls, so that players don't fall off. It must be mentioned that the invisible walls will have some texture in near-future releases. There are instructions in the background moving from right to left, starting from basic controls to advanced team-play and punch mechanics. The tutorial area is for about 1 minute and the actual game starts on the base drop of the music. This is when candies start spawning, the track starts going up and down and bases start to spawn as well.



Button descriptions for menus

Another very important problem was that the players did not know how to click on the buttons since the controls of the menus were not explained, only the controls of the gameplay. We decided to add some text to show to the players what buttons needed to be pressed.



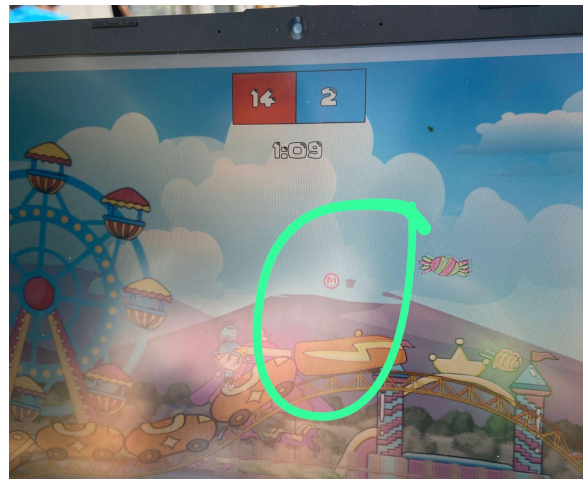
Decrease time of knockback and consecutive punches

At the beginning of the playtesting, most of the players complained that when they were being punched they had no way to defend themselves. The knockback time was too long and it meant the opposite team was able to knock them out easily if they got into an unlucky cycle. We decided to decrease the stun-time so that the players wouldn't get knocked out as easily.

Bugs

Player disappears when spawned

We had some players discover this bug, where once the player was respawned in the base the sprite and animations wouldn't show making the character "invisible". The player component was still in the game but for some reason the sprites and animations didn't work. We were very surprised by this because we were unable to replicate this bug and would only see it when the players would play test.



Work log

Week	Goal	Tasks	Assigned to	Hours expected	Actual hours
Week 11: 19.06 - 25.06	Playtest	Prepare playtest sessions	All	10	5
		Arrange sessions	All	20	15
Week 12: 26.06 - 02.07	Add improvements	Feedback from playtesting and improvements	All	10-15	30
	Milestone 4 documentation	Presentation	All	5	5
		Document	All	4	4

Milestone 5: Final release

Fall For Me!

Team Cicisoft

Andrea Solanas de Vicente

Ankur Deria

Michael Dey

Bendegúz Timár

Task progress

Our goal for the final release was to produce the last finished working version of our game that implements the entirety of the game with all the elements from the previous milestones and some changes added after the playtesting. In order to reach this goal, we modified aspects of the game, we upgraded some UI elements (menus, tutorials...) and also added some new sound effects.

Game summary

Fall for me is a competitive multiplayer fighting game, in which two or more players (up until 4) battle on top of a moving roller coaster to see who can catch the most candies. All players will play on the same device with multiple controllers. All the players have the same type of character and can use the same actions to fight with each other, but the main goal of the game is to have the most amount of candy.

The setting is an amusement park in the background while the players are placed on top of the roller coaster in front of the screen. The roller coaster moves to the right while the players must try to stay on top and catch all the candy they can. They have to try and beat the opposite team by pushing them off the roller coaster. Players will also have to watch out for obstacles and try to stay on top of the roller coaster themselves.

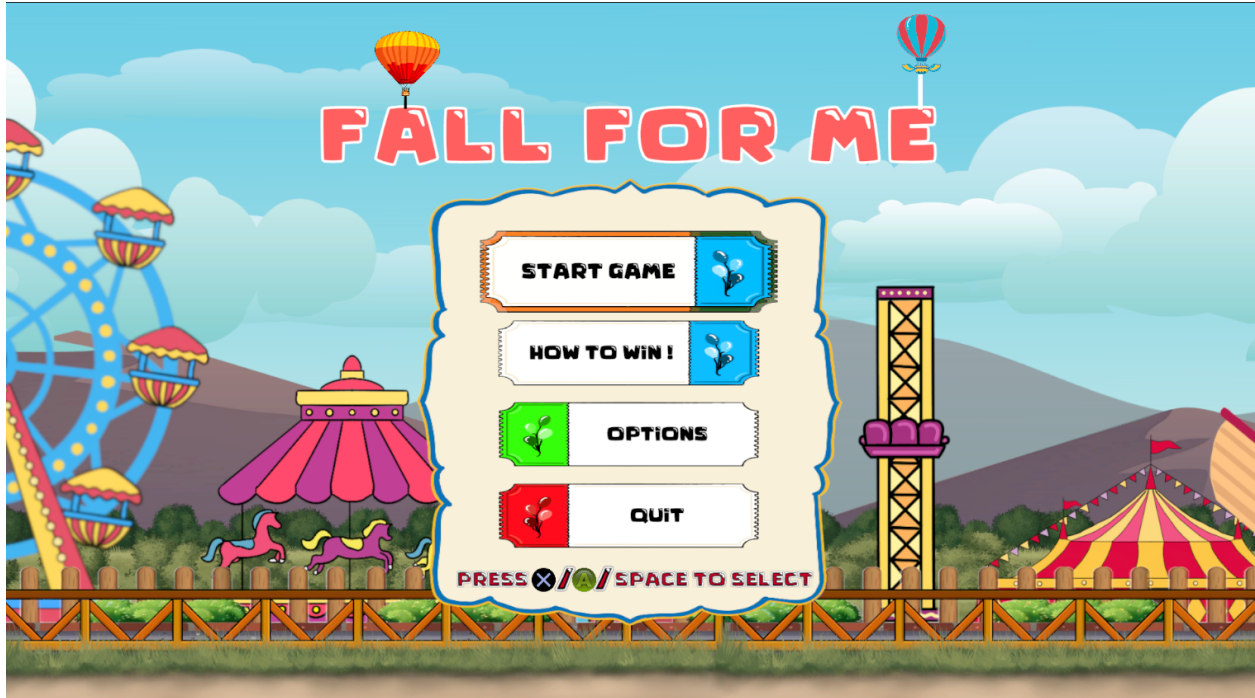
Our game merges a fun and competitive setting of an amusement park and fighting style videogames. Since its multiplayer the game offers a great chance for players to play with their friends or against them.

Final game overview

In this chapter, first provide a summary of your final results including screenshots from your final game.

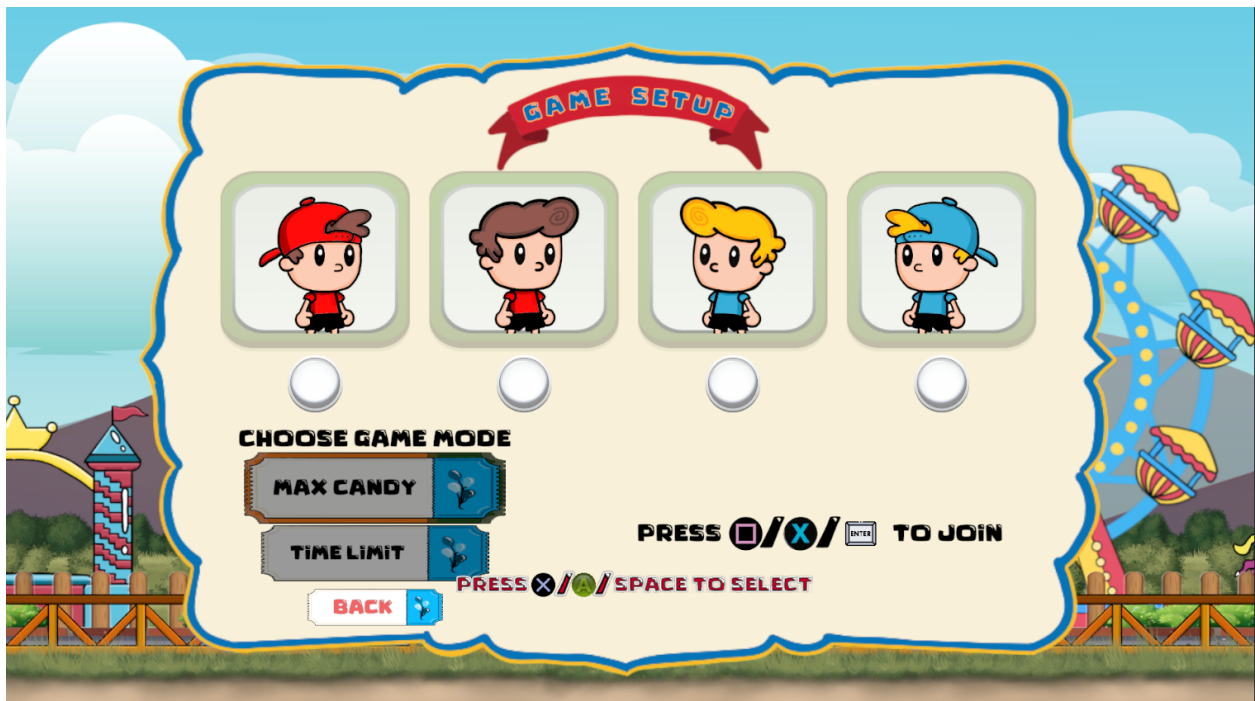
User Interface

We added an introduction screen where the title of the game comes at the same rhythm as the music.



In the main menu players can choose to start the game, learn how to play, check out the options for the game, and also quit the game.

If the players choose to start the game they will be directed to the player choosing screen where they will be able to choose what character they want to play as well as what game mode to play.



Players can always go back to the main menu in case they misclicked or just would like to go back to the main menu.

If players choose to learn how to play the game, a series of tutorial pages will inform the players of the rules of the game as well as some tricks to win. The players can move through these pages like a book and go back and forth between them as they please, to really learn and understand the rules of the game. Here are 2 of the 6 different tutorial pages that are shown to the characters.



If players choose to see the options of the settings of the game they will be directed to the settings page, where they will be able to modify the volume of the music, effects and the overall sound of the game itself. They will also be able to prevent the chill area from appearing at the beginning of the game (this will be explained later).



Once in the gameplay the players can decide to pause the game as they please. We have the pause menu where the players can take a break from the game. There they can choose to resume the game, check the controls and exit the game.



Main game

The main game consists of a moving screen where the rollercoaster is the main object moving. The background is a series of amusement park rides to integrate the players into the game and make them feel like they are in an amusement park. The players are positioned on top of the roller coaster at the beginning of the game and from there they can start playing.



There are two modes of game play: first team to get to 25 wins, or 5 minutes on the clock and whoever has the most candies wins.



There can be 2 players or 4 players. If 2 players are playing it's 1 versus 1, but if 4 players are playing it's 2 versus 2.

Changes from Alpha Release

Here we show the multiple changes and upgrades we have done to the game since the alpha release milestone. Most of these changes were added after the feedback we obtained from the playtesting which was very beneficial to us.

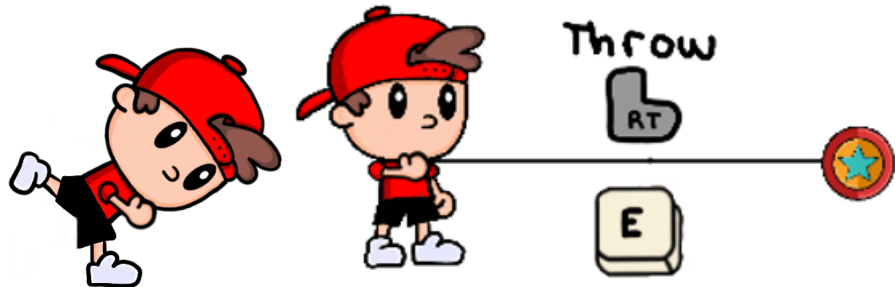
Changes made to the characters

We have added two more attacks to the characters. Before the characters could punch as a fighting mechanic, but now they can also kick backwards and throw a Yo-yo to drag players closer to them.

BEFORE



AFTER



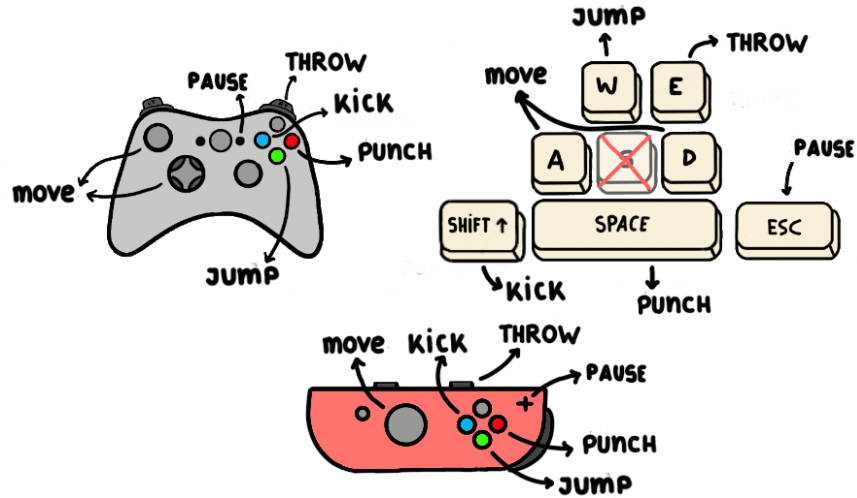
We have also made it possible for players to pass some of their candy to their teammate (if there are 4 players). They just have to punch their partner and they will get half of the candy they are carrying. This motivates players to try and work together with their teammate.

In the character choosing menu, before you could only select the two main characters if you were playing 1vs1 but now players can select any player they want to play, giving them more variety every time they play. That being said, they still have to be on opposite teams in case there are only 2 players.

As a last add on, we added an attack combo to make the punching better. If a player punches the same player 3 times in a row, the last punch will have a higher force than the previous ones.

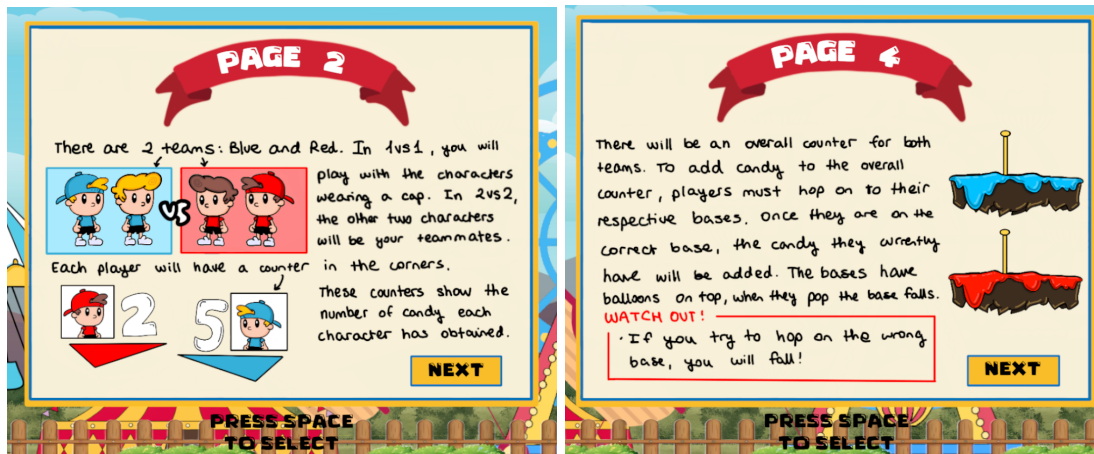
Changes made to the User Interface

We added a new type of controllers for the game: Nintendo Switch controllers. This was we have more options for players to play the game. Since we added this we had to modify the controls showing in the game for the players.



During the testing we noticed people were not reading all of the tutorial pages since they were too long and had too much text. We upgraded them to feel easier to read and more visual for the players.

BEFORE



AFTER



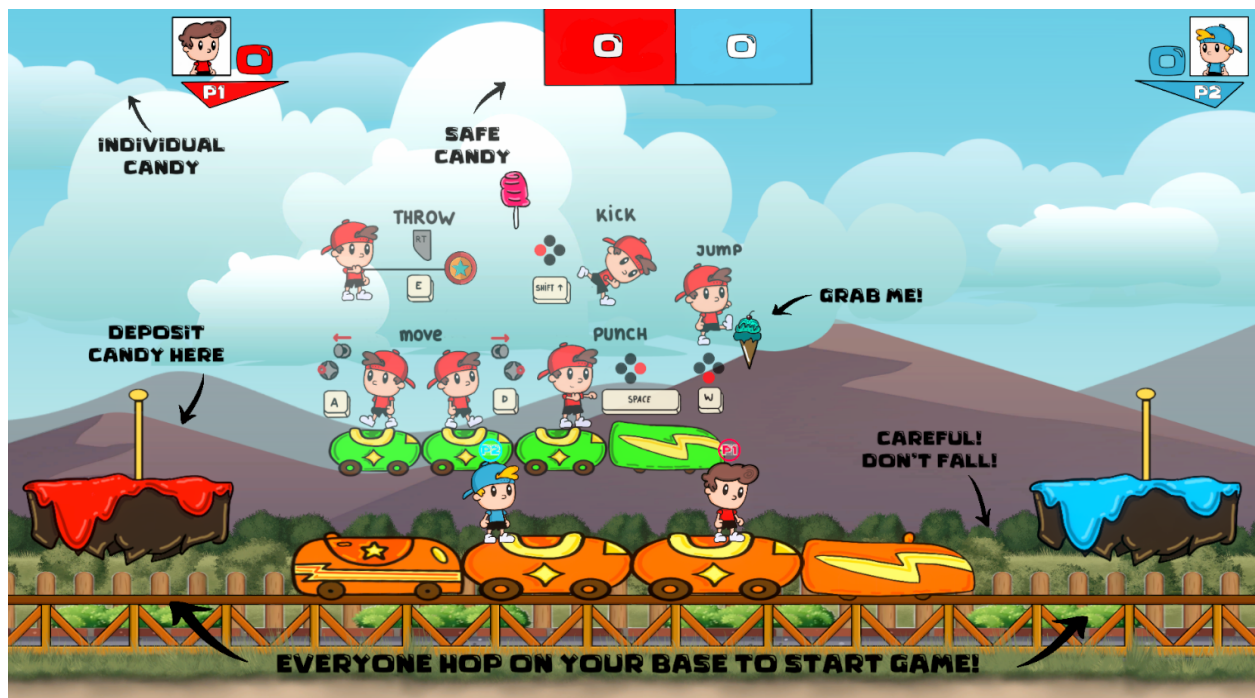
As mentioned above, we have an options menu where the characters can modify the sound settings and choose if the chill area is activated or not.

Changes made to the gameplay

To make the game more interesting we added an obstacle for the players to try and dodge. We created a bird that swoops in from above and drags whoever it finds in its way. This way the characters are kept on their toes even if the rest of the players have died.



As well as changing the tutorial pages, we added a chill zone/area at the beginning of the games where players can practice the controls and learn how the game works before actually starting the game. If the more advanced players don't want to deal with it, they can always take it out in the options menu.



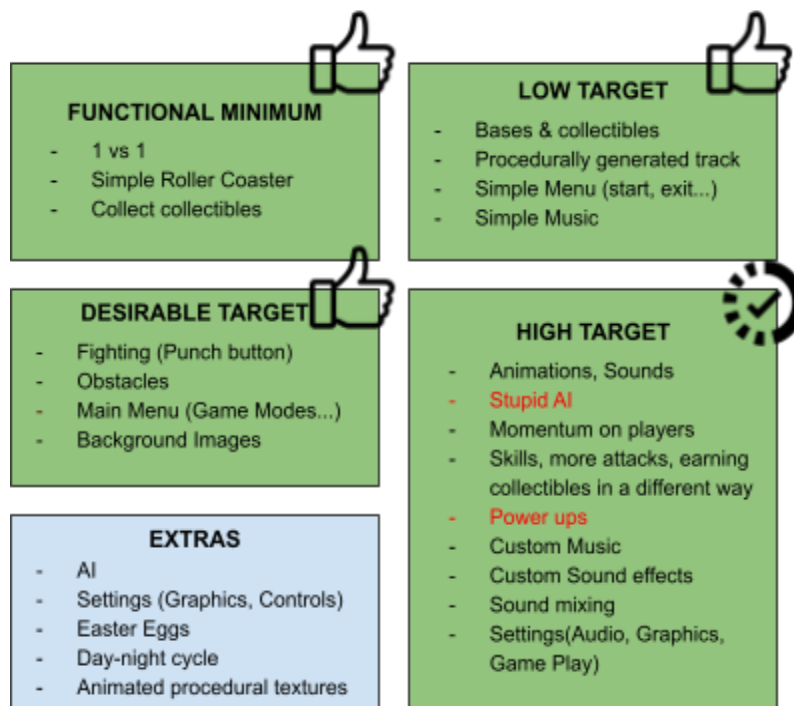
Another complaint from the playtesting was that the respawn was too long so we changed it to now the players respawn 3-4 seconds after they die in the middle of the rollercoaster. This way the players get more playing time and it's more entertaining.

We added a second way of playing, with a clock that counts down and whoever has the most candies at the end wins. This was already implemented during the testing to see which mode the testers preferred.

Changes made to the music

We have added a few new sound effects for all the new items added to the gameplay like the obstacles or the new character mechanics. We also created new music for the chill area/main menu.

Here is the same image we have been using in previous milestones to show the advancements we have made. We decided not to do any power ups since we thought the new attacks were enough for the players to enjoy the game more.



Experience during class

Seeing our game from the beginning to now, it really shows how far it has come. We were able to meet almost all of our goals and targets on time. In some milestones we were very ahead of what we had planned, but in others we fell a bit behind. In the end everything came together.

The design of the game is the same from the beginning but we have adapted some of the requisites of the game to the feedback we have obtained and also our own personal impression of the game. For example, the respawning is different to how we first envisioned it, and we have more attacks than we thought of in the beginning.

In terms of the development schedule as we said, we were able to keep up with it most of the time, with very minor exceptions.

All of the different elements of the project helped us one way or another. The schedule helped us stay on track with what we wanted to do. The interim demo was the first time we could show our game and have a first prototype. The testing was great for us to obtain some feedback from people. This was in our opinion very important and it helped us a lot before the final submission.

The course met our expectations and we are very happy with how our game turned out. We are planning to spend more time developing it even after this class since we think it has the potential to be better and a lot of fun.

Week	Goal	Tasks	Assigned to	Hours expected	Actual hours
Week 13: 03.07 - 09.07	Milestone 5 documentation	Document	All	6	6
		Demo presentation	All	5	3
Week 14: 10.07 - 16.07	Final video	Prepare video and editing	All	10	6