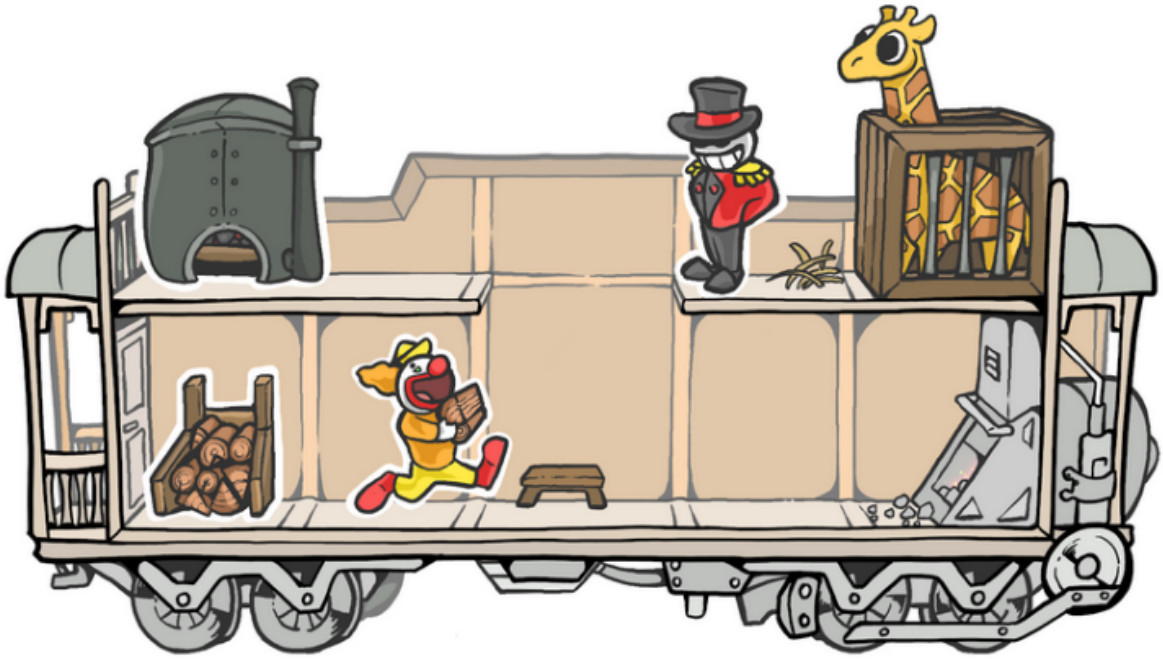
Game Programming Laboratory

Silly Gilly

"The Show Must Go On"



Mikael Stellio - Producer, Programmer, Designer
Felix Hasler - Technical Artist, Designer
Elias Huwyler - Software Architect, Programmer, Designer
Silvia Nauer - Sound Architect, Programmer, Designer
Sean Bone - Physics Programmer, Designer

Contents

1

Formal Project Proposal

1.1 Game Description

1.1.1 Overview

The players are members of a traveling circus company, which moves from town to town by train in order to perform.

The game takes place on the train, in a 2-dimensional side-view cutaway always showing the full train on the screen (see Figure 1.1). The wagons have multiple floors that can be stacked vertically throughout the game and each floor has slots that can contain machines. Up to four players need to keep the train running by moving between machines on the train and carrying resources to keep said machines working. They need to construct circus tents to perform when they reach settlements in order to gain more resources, machines and extensions to the train.

The game ends when the players are no longer able to keep up and run out of resources – the train stops, and the circus company is attacked by bandits, or just starves in the deserted wasteland that stretches out between settlements.

1.1.2 Background Story

In a world where sadness rules undisputed, a lonely circus named Silly Gilly embarks on a journey to spread joy and happiness over the Sadlands and bring hope to the decaying civilization, but their mission reveals itself to be more complicated and risky than expected!

The weak try to survive in the remaining towns or villages, while the evil roam the arid landscape organized in roaring convoys, with one objective in mind: to steal what they want from

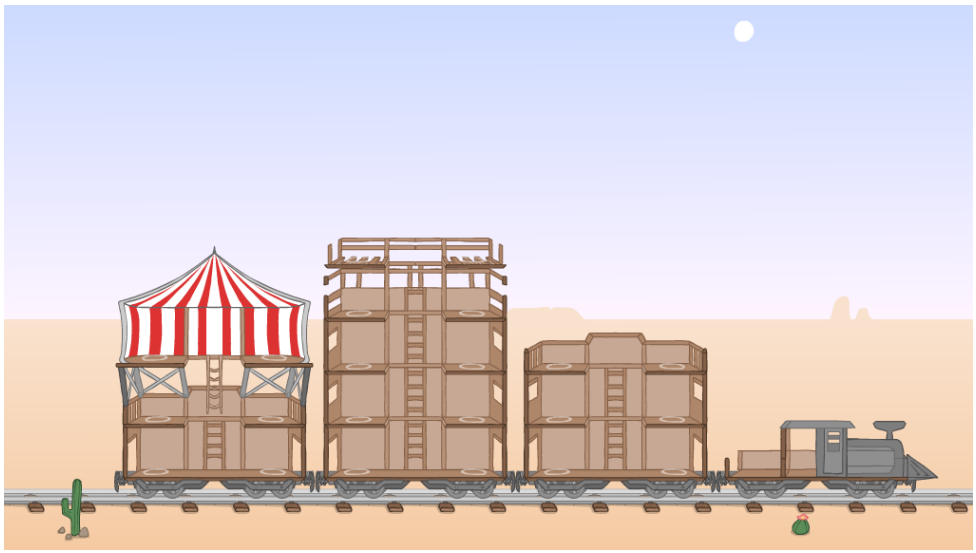


Figure 1.1: *Possible train structure*

everyone who comes across their path. But there is always a bigger fish! Nature is neither generous nor forgiving in the Sadlands, rather it is aggressive and full of nasty surprises.

Is it even possible to teach happiness to a world whose daily motto is: "The ones who stop get scrapped for the benefit of the fastest"?

Will the circus train ever stop? You decide!

1.1.3 Design Decisions

We began our design process by laying down some fundamental requirements we wanted our game to fulfill:

- **Challenging and fast-paced:** We want our game to be intense, constantly keeping players on the edge of their seats. The tension is broken up by alternating phases of the game to give players a break.
- **Easily extendable:** Given the tight schedule and the nature of this project, we want to design our game in such a way that it can be broken down into modular parts, and extended gradually whilst being playable at each stage. This approach impacts game design, code and artwork.
- **Cooperative multiplayer:** We want our game to be multiplayer, and instigate a cooperative style of play.
- **Modern circus theme:** Our game will be themed around a modern circus company.
- **Cartoony and noisy:** We want our game to "look and feel" cartoony and noisy. This affects the visual art style, animations and sound design, but also gameplay aspects.

We opted for a 2D platforming game: this fits nicely with our chosen theme and background – the circus train already mostly restricts gameplay to two dimensions of motion. The train is

composed of a locomotive plus four wagons with a fixed position (see Figure 1.1). Each of the floors of a wagon has two machine slots, and floors can be stacked vertically in order to obtain different configurations of the train and additional space. Players can also move machines between slots, but single-use wrenches are required to do so, limiting the degree to which the players can reconfigure the train.

The players have to move within the train, moving items from one machine to another whilst keeping track of what needs doing and avoiding losing speed, since the only game ending condition is met when the train stops, independently from what causes it. For instance: In order to keep the train moving, the locomotive must be constantly refilled with fuel. Coal is more effective than wood, however, coal must first be produced by feeding wood into a coal kiln (see Figure 1.2).

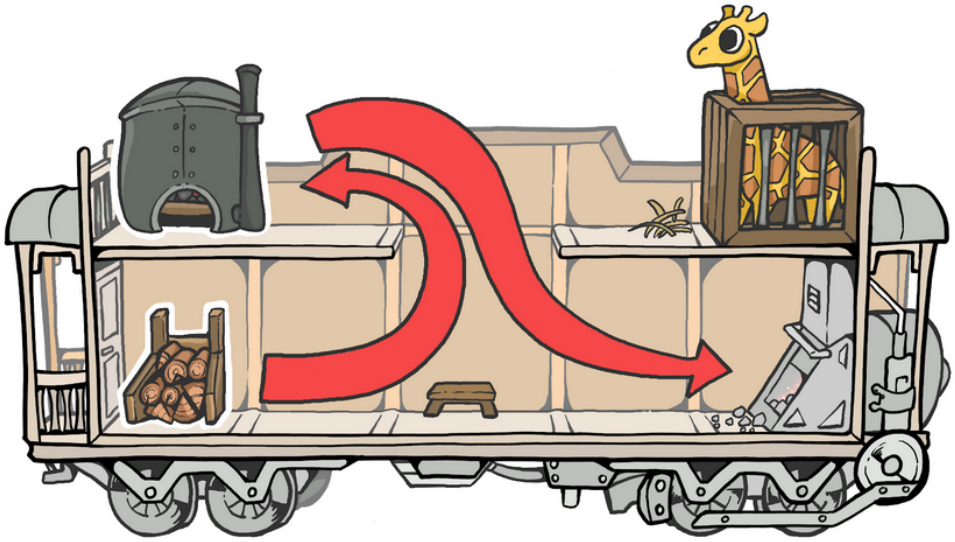


Figure 1.2: *Example management cycle: Coal kiln*

In order to buy resources (wood, water, animal food, etc), the players require money. This can be earned by setting up tents to perform in when they pass through settlements. Tents can be erected on top of each train wagon by feeding wood and cloth into a special machine. When the players pass through a town, they will be rewarded with money for each fully constructed tent, with greater rewards for tents that are populated with animals, character players, etc. The downside to constructing tents is that they slow down the train – build your tents too soon, and you risk running out of money (and therefore resources) before you reach your destination. Lastly, it is important to point out that tents can be seen as a special type of machine that produces money, but is active only when the train is passing through a city.

After receiving monetary rewards for their performance, players will also receive wrenches (to rearrange machines in the train), extra wagon floors (for more machine slots), as well as new animals and machines.

In order to develop the game such that it fulfills our requirements at every stage whilst also being a playable game no matter how far we get in development, we decided to break gameplay down into three semi-independent phases, which alternate and repeat:

1. **Travel phase:** The train travels from one city to the next, and the players have to manage feeding animals, keeping machines running, etc. whilst not running out of money and preventing the train from stopping. Moreover the players should be aware of how much time they have before reaching the next settlement, based on their speed, in order to build tents in time. Low speed gives more time to assemble tents, but more resources will be used to keep the circus running and vice versa for high speed.
2. **Performance phase:** During this phase the train is passing through a settlement and the assembled tents start to produce money depending on the animals or players they contain, and in the future players might be required to perform some simple task (e.g. button mashing) to obtain more rewards. In this phase the locomotive does not need fuel to keep going, thus the train speed will remain constant so that the players have time to focus on earning the different rewards by managing the tents.
3. **Reward and reconfiguration phase:** Based on their performance, the players receive monetary rewards, as well as extensions to the train in the form of new machines, animals and wagon floors. They could also receive a certain number of wrenches, which allow them to move machines within the train. In this short phase the train just came out of a city and the players have some time to dismantle tents to avoid losing speed, and re-organize themselves before a new travel phase filled with events and stuff to manage begins. In this phase the management is still at the same level as in the performance phase.

Every phase has a pre-defined distance (communicated to the player via the UI) that the train has to travel before entering a new phase, thus depending on the speed of the train (which is managed by the players), each phase could take more or less time. Generally, travel phases are the longest, while performance and reconfiguration phases take less time, as in real life passing through a city takes less time than going from city to city.

A fundamental design decision we took is that the game should play out as an uninterrupted train ride – the transitions between different gameplay phases is smooth (no scene changes), and each takes place on the train. However, since the game needs to be playable without every part being fully implemented, this will not be the case from the very beginning:

1. **Functional minimum:** To begin with, only the travel phase will be implemented. Upon reaching a city, the game is interrupted, and players receive a monetary reward based on the number of tents they have and how they were populated (animals and player characters). Then, a new level is loaded, with a different train configuration (which is designed by the developers), and the game continues until the next town. This makes for a playable game, albeit not our desired target.
2. **Reconfiguration phase:** Once we are happy with the travel phase, we will add the reconfiguration phase. At the beginning of each level, players get to choose extensions to the train (cargo containers, machines, animals, etc.) and receive wrenches to reconfigure existing machines. Levels are no longer rigidly designed by the developers, and the duration of the game is no longer limited by the number of levels implemented, but rather by the players' ability to keep up with management.
3. **Desired target:** Finally, the whole experience will be stitched together into a single continuous train ride by adding a simple performance phase (still no button mashing etc).

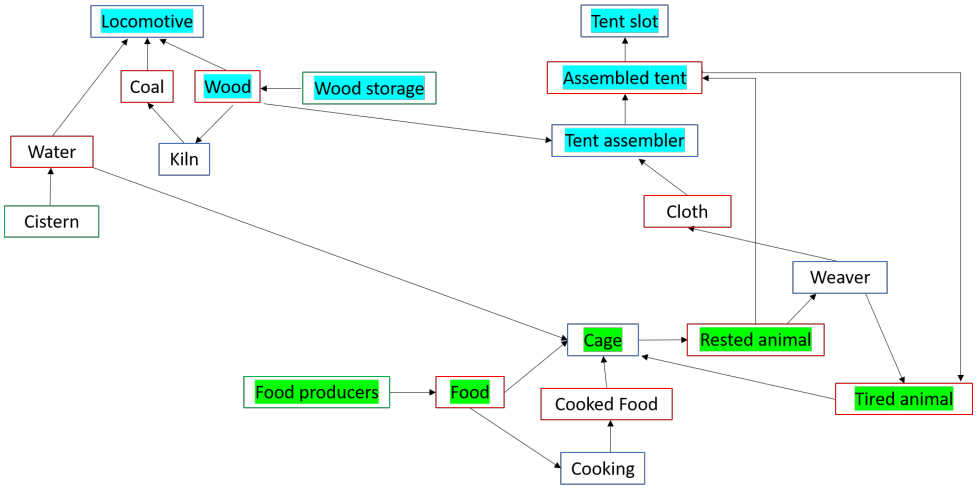


Figure 1.3: *The gameplay elements for the management phase*

1.2 "Big Idea" Bullseye

"Silly Gilly - The Show Must Go On" is a fast-paced cooperative resource management game. The primary conceptual idea of the game is the cooperative management of the resources on the convoy during the travel between towns. This management task is fast-paced: animals become hungry, the locomotive needs coal and external events occur which must be dealt with.
The technical achievement is that every run from scratch of the game must be different and the difficulty of the game must increase gradually between trips.
See Fig. 1.4 for a graphical representation of the “Big Idea” Bullseye.

Figure 1.4: *Big Idea Bullseye*

1.3 Technical Achievement

Every run is different, and difficulty increases gradually between trips.

One of the main goals when creating level-based games is to not just increase the difficulty but also provide unique challenges with each run. Forcing players to change their strategy over time keeps the game engaging and the players entertained. In our case this is done by hitting the player with more conflict or restricting their ability to customise their train.

Our concept includes a wide range of parameters with which the main gameplay loop can be diversified, obvious changes being the addition of resources, machines and environmental events with each successful run. Any change can have a devastating effect, even with less obvious measures such as automatically moving important machines to a different part of the train.

While using manually defined levels in the low target phase of the project we want to observe the effects certain parameter changes have on player strategy. Later we want to write a generative model that sets these parameters automatically while also (on average) increasing the difficulty. This rogue-like mechanic adds great replayability to our game if done right, but balancing it in such a way that it produces runs that are both interesting but not impossible is not an easy task.

1.4 Development Schedule

1.4.1 Layered Task Breakdown

Functional Minimum

- Travel phase with hardcoded levels
- Monetary rewards based on constructed tents
- Filler graphics, no animations
- Locomotive with wood and wood producer
- Tent construction with only wood (no cloth)
- Collisions, interactions, basic physics, player movement

Low Target

- Basic animations
- Basic graphics (moving background, parallax, characters, ...)
- Animals (giraffes and tigers), feeding
- Named towns
- Distinguishable player characters

- Coal and Kiln
- User interface (speed/temperature index, ...)

Desired Target

- Sheep and cloth
- Environmental interference events
- Threats beyond resource depletion
- Performance phase
- Build/rebuild mechanic
- Background music
- Basic foley and stock sounds
- Complete graphics (no more fillers)
- Continuous gameplay, but with hardcoded/finite levels

High Target

- Expanded sound design
- Full animations
- Extra machines, animals and resources
- Infinite automatic levels
- Varied customizable player characters

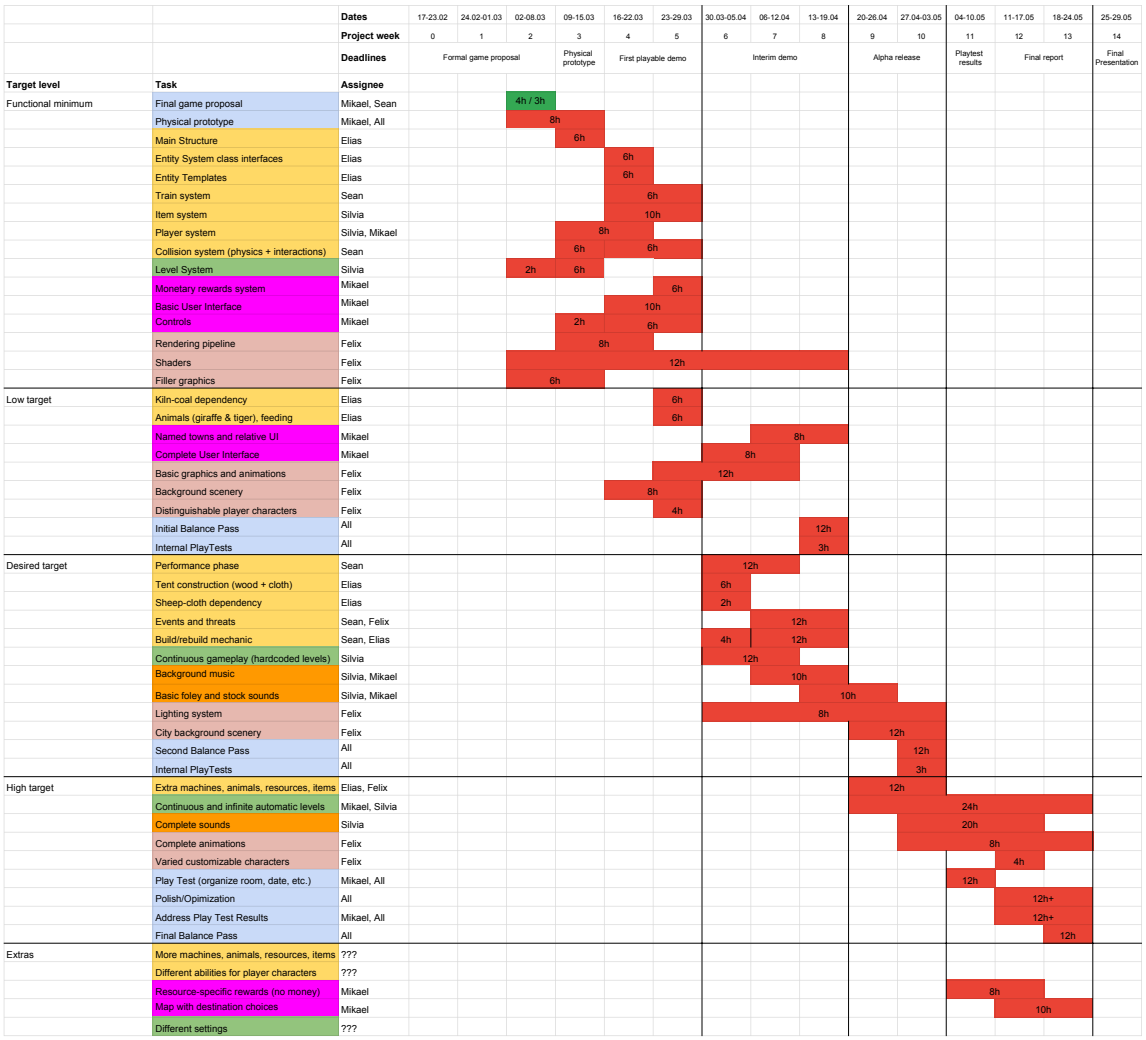
Extras

- Map with destination choices
- Resource-specific rewards instead of money
- Different abilities for player characters
- Different settings
- More stuff!!

1.4.2 Task List

For a task list and the relative responsible person see Fig. 1.5.

1.4.3 Timeline



		Dates	17-23.02	24.02-01.03	02-08.03	09-15.03	16-22.03	23-29.03	30.03-05.04	06-12.04	13-19.04	20-26.04	27.04-03.05	04-10.05	11-17.05	18-24.05	25-29.05
		Project week	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
		Deadlines	Formal game proposal			Physical prototype	First playable demo		Interim demo			Alpha release		Playtest results	Final report		Final Presentation
Target level	**Task**	**Assignee**															
Functional minimum	Final game proposal	Mikael, Sean			4h / 3h												
	Physical prototype	Mikael, All			8h												
	Main Structure	Elias				6h											
	Entity System class interfaces	Elias					6h										
	Entity Templates	Elias					6h										
	Train system	Sean						6h									
	Item system	Silvia						10h									
	Player system	Silvia, Mikael				8h											
	Collision system (physics + interactions)	Sean				6h		6h									
	Level System	Silvia			2h	6h											
	Monetary rewards system	Mikael						6h									
	Basic User Interface	Mikael						10h									
	Controls	Mikael				2h		6h									
	Rendering pipeline	Felix				8h											
	Shaders	Felix						12h									
	Filler graphics	Felix			6h												
Low target	Kiln-coal dependency	Elias						6h									
	Animals (giraffe & tiger), feeding	Elias						6h									
	Named towns and relative UI	Mikael								8h							
	Complete User Interface	Mikael							8h								
	Basic graphics and animations	Felix							12h								
	Background scenery	Felix					8h										
	Distinguishable player characters	Felix						4h									
	Initial Balance Pass	All									12h						
	Internal PlayTests	All									3h						
Desired target	Performance phase	Sean							12h								
	Tent construction (wood + cloth)	Elias							6h								
	Sheep-cloth dependency	Elias							2h								
	Events and threats	Sean, Felix								12h							
	Build/rebuild mechanic	Sean, Elias							4h	12h							
	Continuous gameplay (hardcoded levels)	Silvia							12h								
	Background music	Silvia, Mikael								10h							
	Basic foley and stock sounds	Silvia, Mikael									10h						
	Lighting system	Felix							8h								
	City background scenery	Felix										12h					
	Second Balance Pass	All											12h				
	Internal PlayTests	All											3h				
High target	Extra machines, animals, resources, items	Elias, Felix										12h					
	Continuous and infinite automatic levels	Mikael, Silvia												24h			
	Complete sounds	Silvia											20h				
	Complete animations	Felix												8h			
	Varied customizable characters	Felix													4h		
	Play Test (organize room, date, etc.)	Mikael, All												12h			
	Polish/Opimization	All													12h+		
	Address Play Test Results	Mikael, All													12h+		
	Final Balance Pass	All														12h	
Extras	More machines, animals, resources, items	???															
	Different abilities for player characters	???															
	Resource-specific rewards (no money)	Mikael												8h			
	Map with destination choices	Mikael													10h		
	Different settings	???															

Figure 1.5: *Gantt chart*

1.5 Assessment

The game is aimed at 3-4 players looking for a cartoony fast-paced cooperative resource management game. We are aiming to have a colorful cartoon look for our game which should be consistent over the whole game experience. The players should have a different experience every run of the game, akin to a rouge-like. There should always be the sentiment of "Ahh well we failed, let's try again and get further in the next run!". So the overall balance of the game and the difficulty ramp up are important aspects of the overall success of the game idea.

2

Prototype

2.1 Prototype Setup

Our physical prototype simulates the final video game with a turn-based board game where four players move on a grid to keep the train going for as many turns as they can.

At the beginning of the game the play area consists of only the locomotive (where the engine is located) and two wagons. For the first turn, the players are distributed equally among the four bottom floors. They start with 10 coins to sustain the train during the first section and they can place a free wood producer in a machine slot of the second floor of the second wagon of their choice. Also, the Game Master (GM) sets the initial speed of the train to 6, the length of the first travel phase to 20 and the length of the first performance phase to 2d4 (see Figure 2.1 for a possible initial setup).

Figure 2.1: *Initial setup of the game*

Each turn is divided in three different phases:

- Player actions
- Train movement
- End of turn

The first phase is where players actually play the game, while the second and third one are the GM's responsibility.

2.1.1 Turn Description

Each turn, during the player actions, each player can move up to 4 tiles in accordance with the following rules:

- A player cannot move diagonally, but only horizontally or vertically.
- A player cannot end their movement or dash action on a tile with no floor.
- A player cannot move through walls, floors or roofs.
- A player can pass through and end their movement/dash action on a tile occupied by other players.

Moreover, each player can perform a single action from the following list:

- **Dash:** Move 2 additional tiles.
- **Interact:** Use a machine on the tile you are on and resolve the corresponding machine effects, e.g. put wood in the kiln and get coal back.
- **Throw:** Pass an item you are holding to a player with no item within 2 tiles (determined with respect to the movement rules).

Also, once per turn a player can pass an item to a player on their same tile as a free action. If the other player is also holding an item, they can use the same free action to swap items (this is not valid for normal throw actions).

Figure 2.2: *Players*

It is important to specify that the players take their turns simultaneously and can interleave movement and actions as they like, e.g. Alice moves 2 tiles, then Bob uses his action to use the kiln and moves 4 tiles to reach the same tile of Alice in order to use his free swap action to pass a piece of coal to Alice, then she moves her remaining 2 tiles and uses the dash action to move 2 additional tiles towards the engine.

To avoid stalling, the GM should make sure that players don't lose too much time discussing what to do (1'30" per turn seems to be a fine choice). After all the players have finished their

turn the GM proceeds by moving the train on the distance counter by the current speed value and decreases the current speed by one (note that the speed has a maximum value of twelve). Then, in the third phase, the GM performs the following actions in the specified order depending on the location of the train.

If the train is in a city:

- Gain coins from the assembled tents, with a maximum total gain of 10 coins per performance phase. Add 10 coins to this monetary cap every 10 cities to avoid getting limited by it after a while.

If the train is on the railroad:

- If the train exited the city this turn, the players can buy an item of choice for 2 coins (at least one player not holding an item is required) or a machine of choice for 5 coins (at least one machine slot is required). As an exception the players can directly buy a cage already containing an animal for 7 coins.
- If the train exited the city this turn with an assembled tent on top of each wagon, the players can buy a new floor for 10 coins. This floor can be placed on top of an existing wagon or can be placed as the first floor of a new wagon, with a maximum of four wagons (remember to move the roof of an existing wagon one floor up or add a roof to a newly created wagon).
- If the train exited the city this turn, every tired animal not in an assembled tent is lost.
- If the train exited the city this turn, remove all the assembled tents. Each animal contained in an assembled tent slot must be put in an empty cage, if there is no empty cage the animal is lost.
- Resolve events, e.g. the GM rolls dice at the end of each turn to determine if and which wheel breaks. Each broken wheel decreases the speed by one additional point at the end of each turn. Broken wheels can be repaired by a player using an interact action on the tile above it.

After the third phase a new turn starts and the players can move and take their actions as before. After exiting a city the GM determines the length of the new travel phase by adding 5 to the previous travel phase length and sets the length of the new performance phase to 2d4 (throw two four-sided dice).

The game goes on until the GM has to decrease the speed below zero, at that point the score of the players is equal to the number of cities completely passed and they can start over to try beating their record again.

2.1.2 Items

In this section all the items will be described.

- **Wood:** The primary resource of the game, used to fuel the train engine, build tents and produce coal.

- **Coal:** An improved fuel option over wood to gain speed.
- **Rested/Tired giraffe:** Herbivore animal used in performances. It cannot be put into a cage on the same floor of a carnivore animal.
- **Rested/Tired tiger:** Carnivore animal used in performances. It cannot be put into a cage on the same floor of an herbivore animal.
- **Rested/Tired sheep:** Herbivore animal used to produce cloth and cannot be used in performances. It cannot be put into a cage on the same floor of a carnivore animal.
- **Meat:** Used to feed carnivore animals.
- **Veggies:** Used to feed herbivore animals.
- **Cloth:** Used to improve the effectiveness of an assembled tent.
- **Wrench:** Can mark two machine slots (two occupied slots or one occupied and one empty slot) as wrenched. The wrench is lost when the second mark is applied and the contents of the two wrenched machine slots are switched.

Figure 2.3: *Items*

Note that each player can hold a single item at a time and nothing can be put "on the ground" outside of a machine slot. Item drops are planned for the final video game, this is an approximation to reduce confusion on the board.

2.1.3 Machines

In this section all the machines and their interaction effects will be explained.

- **Engine:** One of the most important machines, since it determines the speed of the train and the game lose condition. For interactions it takes as input either a unit of wood or coal and immediately increases the current speed by 2 or 4 respectively. The engine is always located on the rightmost tile of the locomotive.
- **X producer:** If used, one coin is spent and a unit of X is given to the user. To use this machine the user cannot hold anything.
- **Kiln:** If used, it takes as input a unit of wood and immediately returns a unit of coal to the user.
- **Cage:** It can hold an animal (put/take animals with an interaction). Additionally, if the animal in the cage is tired, this machine can be used to convert said animal to the rested state by taking the correct food item as input. The food is lost and the animal is now rested and still in the cage.

- **Weaver:** If used, it takes a rested sheep and immediately returns a unit of cloth to the user. The used sheep is now tired and must be put into an empty cage immediately, if there is no empty cage the sheep is lost.
- **Tent assembler:** This is a special type of machine always present on each roof and represented by a little red and white flag. This machine takes its inputs from the left- and rightmost tiles of the relative roof. If an interact action is used on one of those two special tiles, the specific tile takes either a unit of wood or a unit of cloth and cannot take any other inputs. The roof becomes an assembled tent as soon as a unit of wood is put in one of the tent assembler tiles, and then it becomes an improved assembled tent as soon as a unit of cloth is put in the other tent assembler tile. Note that the special tent assembler tiles remain and are still usable even if the roof is replaced with an assembled tent, but, since the assembled tent has special performer slots in the same location, a player using the interact action must specify which effect he/she wants to resolve.
- **Assembled tent:** This is another special type of machine that can be obtained through the tent assembler. Like the tent assembler, this machine can take up to two inputs from the left- and rightmost tiles of its floor (the top of the wagon). These performer slots take either a single rested giraffe or tiger, or a single player as input. The animals can be put in a slot by a player by using an interact action on the tile while holding the rested animal that can perform. On the other hand a player can stay on the performer slot to be taken as an input. Note that multiple players in the same performer slot do not change the amount of coins rewarded. At the end of each turn, a performer slot containing a rested animal immediately returns 2 coins and changes the state of the animal to tired, while a slot with at least one player standing in it immediately returns 2 coins. If both slots contain a player only 3 coins are rewarded in total (for one tent). An assembled tent that has been improved with cloth returns double the normal amount of coins. E.g. a train has two assembled tents on top of its wagons, one containing a player in each of its performer slots and the other containing a player and a rested animal. After a turn in a city this configuration returns 3 coins for the first tent and 4 coins for the second one, but after a second turn in the city the second tent returns only 2 coins, since the animal is now tired.

Figure 2.4: *Machines*

Note that machines can only be placed in machine slots. Each floor (not the roofs) has two machine slots located at the left- and rightmost tiles (remember that each floor has five tiles on the horizontal axis and one on the vertical axis).

Figure 2.5: *Special machines*

2.2 Playing Experience

At first we tried different variants of the game to find the most balanced. Then we tried to change some of the rules to reflect different variants of the same mechanics in order to find the best implementation for the final video game.

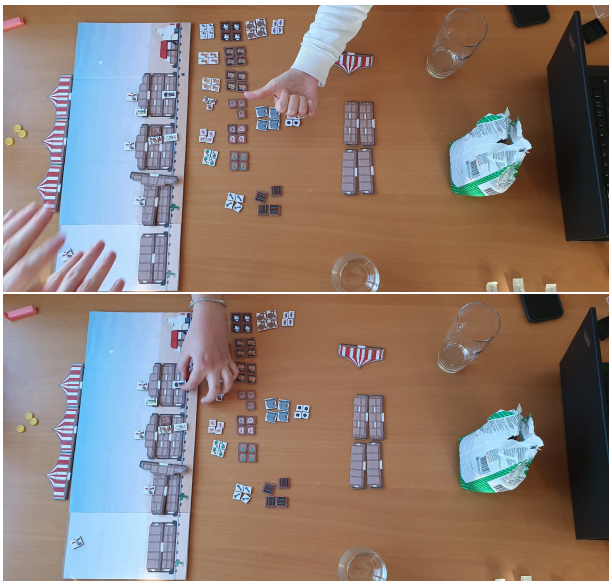


Figure 2.6: *Having fun!*

We had a lot of fun playing, losing and testing for a whole day. The final version of the prototype seems balanced and works quite well, it could be well enjoyed with friends after the end of this project as a proper board game.

2.3 Findings and Conclusion

As aforementioned we tested different variants, thus we also discovered some flaws and balance issues in our initial plan. The following points summarize our main conclusions from playing the physical prototype:

- Keeping the speed of the train constant during the performance phase causes a lot of edge cases that could easily earn a lot of coins to the players just by going really slow. It is better to have decreasing speed even during the performance. To still allow players to concentrate on the performance no events take place in a city.
- Starting with four wagons reduces the need to buy new floors, since from the beginning the train offers sixteen machine slots that are already more than enough for most games. We decided to reduce the starting number of wagons to two and possibly the maximum number of wagons to three. As a side effect, since the whole train is always shown on screen, less wagons allow us to scale up the play area to show more detail and give a better overview.
- To avoid having the players gain a big amount of coins in one city we decided to set a monetary cap for each performance phase, so that every city can give only a maximum amount of coins. This is justified by the fact that cities are not infinitely rich.
- Random events, like breaking wheels, are very important to keep the game interesting and unpredictable, and they add a lot of fun too!
- Using hardcoded mechanics for testing is very useful, e.g. a linear increase of the length of each travel phase is a good method for ramping up the difficulty.
- In the first level the initial speed should be enough to arrive at the first town without putting fuel in the engine. This gives the players time to organize themselves.
- Possible variant: Instead of continuously decreasing the train's speed it could be kept constant, but with a continuously decreasing fuel level instead. That makes it so the train does not slow down but it stops when there is no more fuel.

3

Interim Report

3.1 Progress

3.1.1 Current State of the Game

The current version of the game includes the main logic of the game, most of the final graphics, UI, hardcoded levels and a first basic sound system. Some more details are explained in the following.

After character selection the game starts in the first level where the players are tasked with building tents and keeping the train moving. If the tents are built, during the performance phase (characterized by a different background music and circus graphics) it is possible to earn money for the circus, see Fig. 3.2. This money is used to keep the train operational during the next phases. As the levels progress, the number of tasks for the players increase: Let giraffes and tigers perform during the performance phase, produce cloth with sheep, upgrade tents with cloth (see Fig. 3.1) and feed the animals. Furthermore, during all the levels, external events are triggered at random. As of the interim demo the only external event is the breaking of one or more train wheels (see Fig. 3.1). This event increases train speed degradation and can be resolved by keeping the interact button pressed on the fire for a given time.

With the interim demo almost all graphics systems are now fully established: Train graphics with static light generation, background scenery, a tile-based system for dynamic objects, animated particle effects, menu graphics and post-processing. Especially for the menus there are still a lot of temporary graphics (buttons, backgrounds) that have yet to be added for the alpha build.

The User Interface consists of a HUD and menu system: Players can pause, end or restart the game and change settings about screen and sound. In the pause screen there is a small tutorial

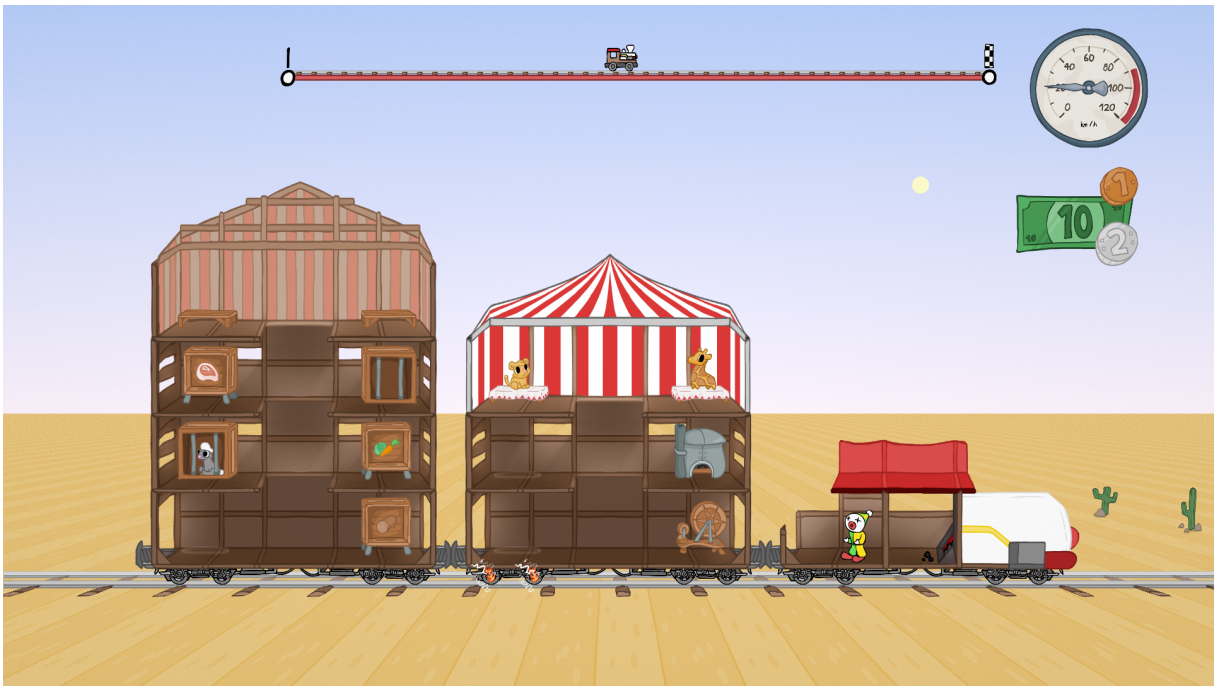


Figure 3.1: *Travel phase. The wagon on the left has a lower-tier wooden tent and the wagon on the right has an upgraded cloth tent. The two back wheels of the first wagon are broken (external event).*

about the game controls. Lastly, there is a game over screen when the train reaches zero speed.

The sound system works as follows: Everything happening in-game has its own sound effect combined with a visual effect (e.g. wheel explosion) and there is background music. The current music assets are not done by us and aren't our final choice, same for some sound effects that still need tuning.

3.1.2 Layered Task Update

The first two layers (functional minimum and low target) are completely finished, the third layer (desired target) is in progress:

Desired Target

- Sheep and cloth: DONE
- Environmental interference, events, threats: PARTIALLY DONE
- Performance phase: PARTIALLY DONE
- Build/rebuild mechanic: POSTPONED
- Background music: PARTIALLY DONE
- Basic foley and stock sounds: PARTIALLY DONE

Figure 3.2: *Performance phase. The graphics of the distance bar is different and if tents are built money is spawned.*

Figure 3.3: *Pause screen. The pause screen contains a small tutorial about the gamepad controls.*

- Complete graphics (no more fillers): PARTIALLY DONE
- Continuous gameplay, but with hardcoded/finite levels: DONE

3.2 Challenges

3.2.1 Monogame Extended ECS

Turns out the warning from the Gamelab team to "be careful about using external libraries" also applies to libraries directly recommended by the Monogame website.

For this project we used the Monogame Extended framework mostly for its entity component system. Unfortunately it only supports 32 component types per scene which mid-way through the project prompted a major rework on graphics and other systems to stay within that limit (away from the ECS pattern towards a visitor pattern).

That limitation came with the component system using a 32-bit mask for filtering. This limitation wasn't documented and strictly goes against the point of using an ECS pattern (which depends on the ability to add a lot of component types for data storage). On a good note, the replacement systems we came up with turned out to be more stable and easier to maintain.

3.2.2 Fullscreen System

For the whole duration of the project we had problems with window size and full-screen mode. While we could count on the Xbox One™having a fixed resolution of 1920x1080, thanks to the COVID-19 outbreak we had to deal with this issue so other Gamelab students can test our game locally on their PC.

Resize and resolution problems are a known Monogame UWP issue that haven't been properly fixed yet, forcing us to change XNA screen options manually. So far our best and most stable solution involves letter-boxing so the game elements don't need to be moved.

3.2.3 Audio and Graphical Feedback

Our game concept calls for a lot of sound and graphical cues to keep the players informed about what is happening on their train. The most challenging aspects included train speed, amount of money, external events, distance to the next city and current state of the performance. We noticed that just having a speed-o-meter (or a distance meter) is by far not enough information, instead we require a lot of movement and sound (to for example convey urgency when the speed gets too low). We ended up with an unexpectedly high amount of different sounds, particle effects and screen space manipulations to convey the correct feeling (and there is still a lot of work to be done).

3.2.4 Entity Removal

In the same vein as with the component limit the way the ECS system handles entity deletion isn't fully documented: It's done in a lazy execution way at the end of each turn based on a list of entities to be deleted. This list has no safeguards and can hold duplicates but the deletion routine crashes when trying to delete the same entity twice. While at the same time we can't check if an entity is already scheduled for deletion as the list is a private member of the entity system and null checks are pointless due to the lazy execution. Furthermore, some of our systems work on lists of entities generated by previous systems, so those need to keep an internal blacklist to make sure no entity is processed too often. It took us several iterations and a lot of seemingly random crashes to track-down all the edge cases and occurrences of entities getting double deleted.

3.3 Future Work

3.3.1 Menu Graphics and Performance Phase

Good menu graphics, if done right, give a good first impression for the game. We plan on adding an animated background and vivid interactive buttons to set the mood before the game has even started. Preferably this task coincides with a little intro sequence for the game trailer.

The performance phase currently only differs from the travel phase by the graphics of the distance meter, background music and money spawning. In the future the background needs to show a city landscape and several light/post-processing effects to convey the feel of a running performance.

3.3.2 Balancing Depending on Player Count

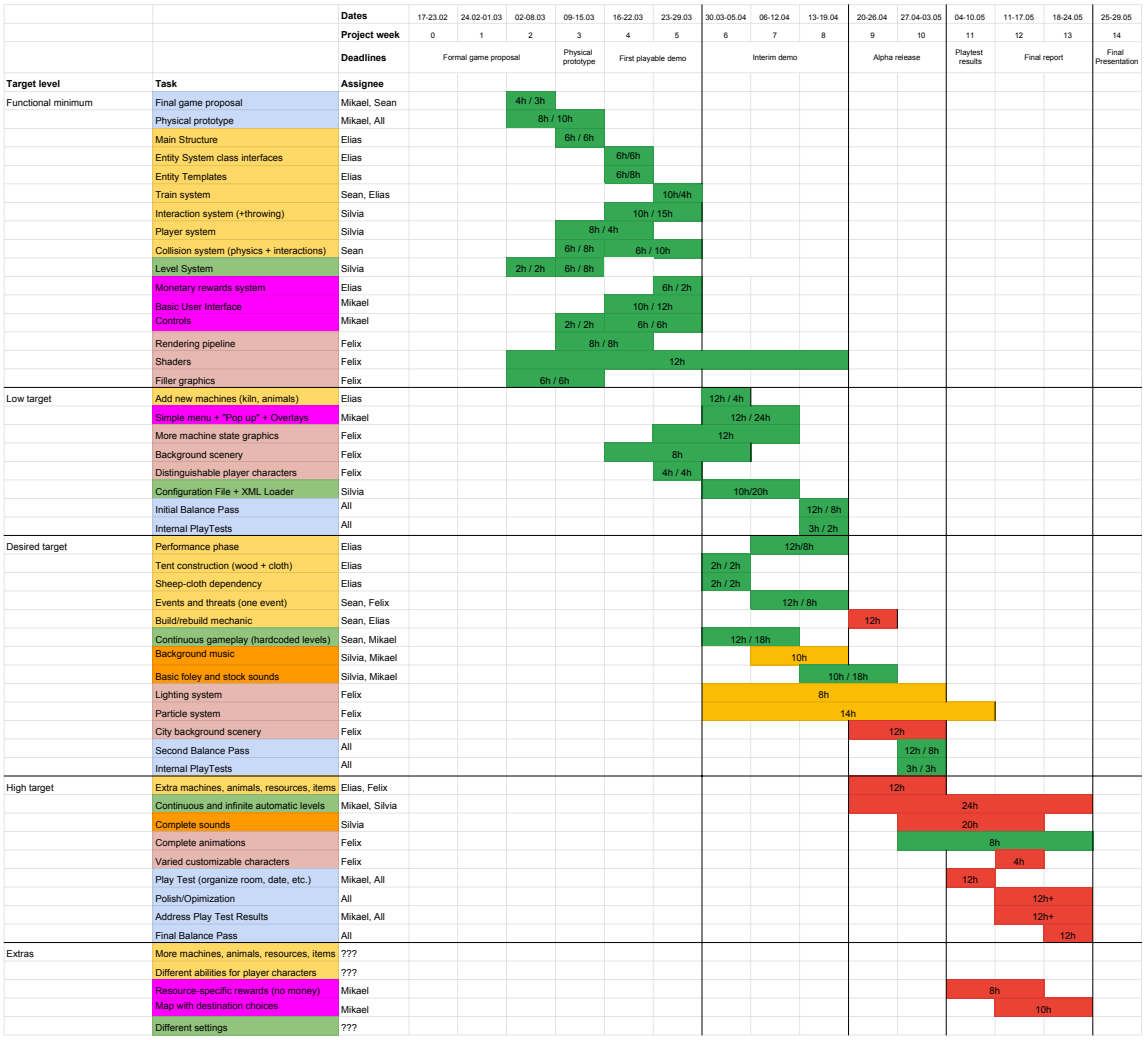
More players necessarily make maintaining the train easier. Balancing based on the player count is required to keep the game interesting. This balancing has to be done on a per-attribute basis (speed degradation, reward amount, etc.) and requires meticulous testing to get right. For now the game is well balanced for two non-experienced players and a draft balancing for one player has been done too, so that anyone can play our game alone if needed, due to the current COVID-19 situation. A correct balancing for three or four player is not guaranteed at the moment, since we lack the number of people for testing.

3.3.3 Complete Sound System

The main next task for the sound system are background music tracks: Our goal is to produce these tracks ourselves. The idea is to take some known jingle and adapt them to our theme by mixing circus motifs with train-inspired percussion.

3.3.4 Procedural Levels

For the moment levels are hardcoded. The goal is to generate them automatically in order to have a continuous, infinite game. The infinite automatic levels, together with a correct balancing of the game, are tightly bound to our technical achievement of increasing the difficulty with the players' progress and always keeping the game interesting between runs. This will be a key point to make progress on in the next weeks.



		Dates	17-23.02	24.02-01.03	02-08.03	09-15.03	16-22.03	23-29.03	30.03-05.04	06-12.04	13-19.04	20-26.04	27.04-03.05	04-10.05	11-17.05	18-24.05	25-29.05
		Project week	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
		Deadlines	Formal game proposal			Physical prototype	First playable demo		Interim demo			Alpha release		Playtest results	Final report		Final Presentation
Target level	**Task**	**Assignee**															
Functional minimum	Final game proposal	Mikael, Sean			4h / 3h												
	Physical prototype	Mikael, All			8h / 10h												
	Main Structure	Elias				6h / 6h											
	Entity System class interfaces	Elias					6h/6h										
	Entity Templates	Elias					6h/8h										
	Train system	Sean, Elias						10h/4h									
	Interaction system (+throwing)	Silvia					10h / 15h										
	Player system	Silvia				8h / 4h											
	Collision system (physics + interactions)	Sean				6h / 8h	6h / 10h										
	Level System	Silvia			2h / 2h	6h / 8h											
	Monetary rewards system	Elias						6h / 2h									
	Basic User Interface	Mikael					10h / 12h										
	Controls	Mikael				2h / 2h	6h / 6h										
	Rendering pipeline	Felix				8h / 8h											
	Shaders	Felix						12h									
	Filler graphics	Felix			6h / 6h												
Low target	Add new machines (kiln, animals)	Elias							12h / 4h								
	Simple menu + "Pop up" + Overlays	Mikael							12h / 24h								
	More machine state graphics	Felix						12h									
	Background scenery	Felix					8h										
	Distinguishable player characters	Felix						4h / 4h									
	Configuration File + XML Loader	Silvia							10h/20h								
	Initial Balance Pass	All									12h / 8h						
	Internal PlayTests	All									3h / 2h						
Desired target	Performance phase	Elias								12h/8h							
	Tent construction (wood + cloth)	Elias							2h / 2h								
	Sheep-cloth dependency	Elias							2h / 2h								
	Events and threats (one event)	Sean, Felix								12h / 8h							
	Build/rebuild mechanic	Sean, Elias										12h					
	Continuous gameplay (hardcoded levels)	Sean, Mikael							12h / 18h								
	Background music	Silvia, Mikael								10h							
	Basic foley and stock sounds	Silvia, Mikael									10h / 18h						
	Lighting system	Felix							8h								
	Particle system	Felix							14h								
	City background scenery	Felix										12h					
	Second Balance Pass	All											12h / 8h				
	Internal PlayTests	All											3h / 3h				
High target	Extra machines, animals, resources, items	Elias, Felix										12h					
	Continuous and infinite automatic levels	Mikael, Silvia											24h				
	Complete sounds	Silvia											20h				
	Complete animations	Felix											8h				
	Varied customizable characters	Felix													4h		
	Play Test (organize room, date, etc.)	Mikael, All												12h			
	Polish/Opimization	All													12h+		
	Address Play Test Results	Mikael, All													12h+		
	Final Balance Pass	All														12h	
Extras	More machines, animals, resources, items	???															
	Different abilities for player characters	???															
	Resource-specific rewards (no money)	Mikael												8h			
	Map with destination choices	Mikael													10h		
	Different settings	???															

Figure 3.4: *Updated Gantt chart (assigned time / actual time)*

4

Alpha Release

4.1 Progress

4.1.1 Current State of the Game

The alpha version of the game includes a difficulty system that ensures a linear increase in difficulty while playing, an in-game tutorial based on pop-ups, randomly and semi-randomly generated levels and a new shop system, together with improved sounds, songs and visual art.

After character selection the players can choose their preferred difficulty. The game selects by default the difficulty recommended for the number of players connected, but a higher or lower difficulty is always available. While playing, tutorial pop-ups will tell less experienced players what to do during the first phases of the game (see Fig. 4.1).

Furthermore, who played the interim demo will notice a new speed-meter and associated speed logic, that should clarify the speed value that are not recommended. Low speed means the players are about to lose or they are going too slow, thus will spend more coins to get to the next city, while on the other hand going too fast increases the chance of spawning new events. In the current version the players have the choice to buy new machines from the shop area on top of the locomotive (see Fig 4.2) and place them where they prefer, with the limitation of only a fraction of randomly selected free machine slots being available.

This alpha release also includes new machines and items, like the well (yes, a well on a train) and bucket that provide the players with water, that can be used to stop speed decrease for some seconds. Another new item is the wrench, that enables the players to move machines in different slots to improve the efficiency of the train.

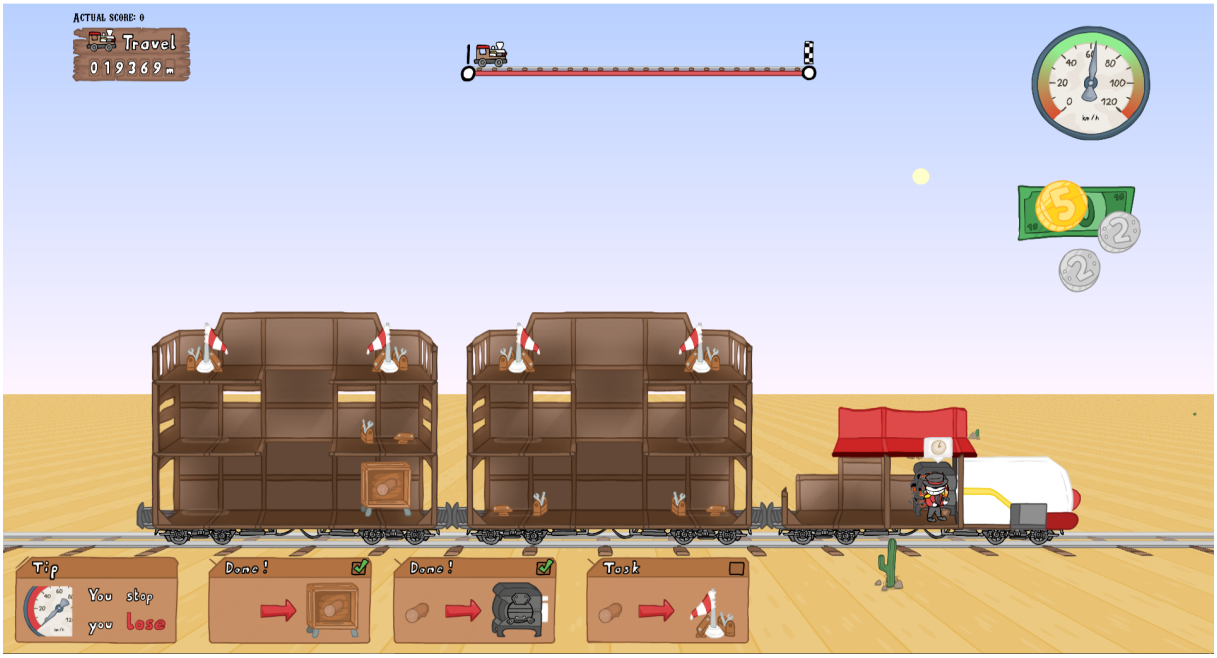


Figure 4.1: *Tutorial pop-ups. Pop-ups for the tutorial appear on the bottom of the screen.*

4.1.2 Layered Task Update

Aside from some sound and graphical aspects, after the interim release we completed the third layer and started working on the high target.

High Target

- Expanded sound design: DONE
- Full animations: DONE
- Extra machines, animals and resources: PARTIALLY DONE
- Infinite automatic levels: DONE (COULD BE IMPROVED)
- Varied customizable player characters: NOT DONE

4.2 Challenges

4.2.1 Monogame Extended NuclexGui

We decided to use an existing GUI implementation for our game, to avoid taking to much work time from the gameplay mechanics. After trying different implementations (Monogame Extended GUI, GeonBit GUI) and discovering that none of those supported gamepad controls, we decided to use an implementation of the Nuclex GUI Framework for XNA ported to Monogame

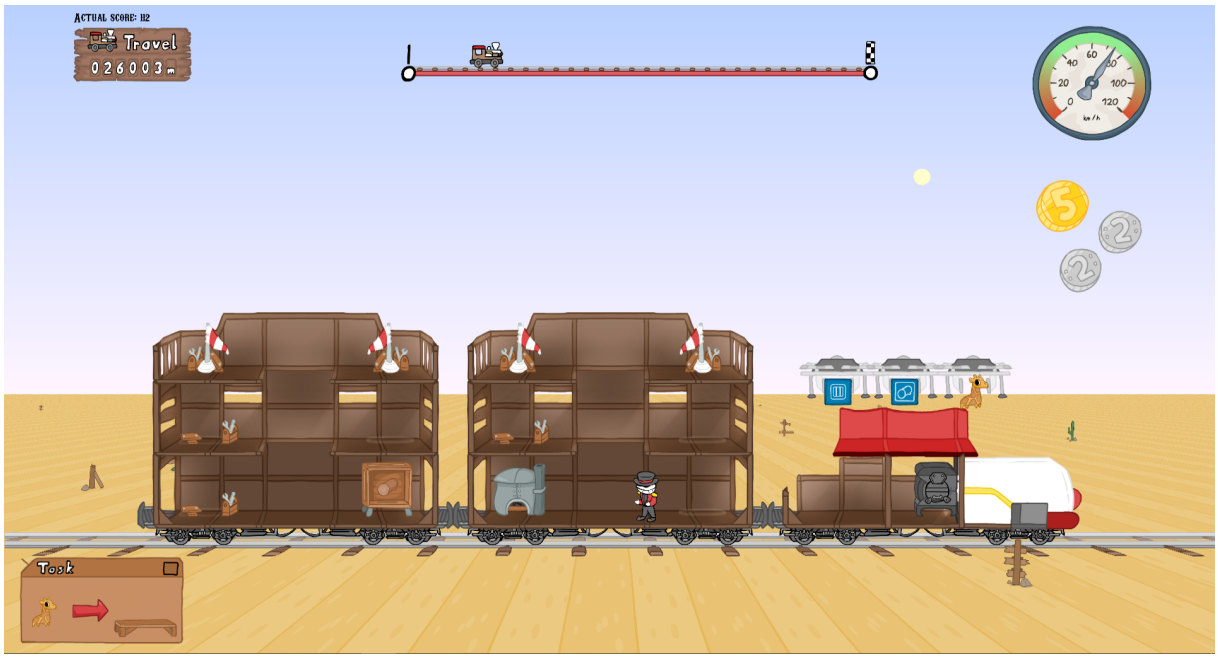


Figure 4.2: *Shop area on top of the locomotive. Three drones provide machines or items which can be bought by the players.*

(Monogame Extended NuclexGui) for our GUI, since it seemed to support the gamepad. Sadly, after testing the implementation with multiple controllers it turned out that only one single controller is supported, thus as workaround we had to implement the support of the other three controllers ourselves. This implementation works fine with buttons, text, and other simple controls, but it starts causing problems with more complicated controls, e.g. sliders. The main problem of Monogame Extended GUI and NuclexGui is the really poor documentation and lack of a complete set of examples, so we had to manually look at the source code to understand how some of the logic worked. In retrospect, implementing our own GUI would have taken more time, but probably give us a more stable system to work with at the current state of the game.

4.2.2 Shop System

We had differing opinions whether we want to do an in-game shop or use a new UI screen for the shop. The overwhelming majority of the team wants to keep the game as menu free as possible so we agreed on an in-game solution. For the in-game solution we needed a place on the train which always stays free and accessible, that's why we decided to use the roof of the locomotive for our shop area. The items available in the shop are decided, by a biased random system. The first slot from the left always gives a new machine not yet on the train, the middle slot is a completely random machine from the pool of unlocked machines, the right slot gives either a newly unlocked item or if no new items are available a random one from the list of unlocked items. The unlocked items and machines are based on the level-system to provide some control on which machines and items are available during the course of the game.

4.2.3 Balancing

Aside from the playtesting problems we had and still currently have due to the COVID-19 situation, balancing the game became even more difficult after introducing randomized levels and the new more complex performance and speed mechanic. The game difficulties Baby, Easy and Medium are currently well balanced for one player and two players, but no guarantee is given yet on the higher difficulties and on how they play for three or four players. The current difficulties are the following:

- Baby: for one inexperienced player
- Easy: for one experienced player or two inexperienced players
- Medium: for two experienced players or three inexperienced players
- Hard: for three experienced players or four inexperienced players
- Extreme: for three experienced players or four inexperienced players (or you can play alone if you want to try hard, but really hard)

Note that in the future we will probably change the names of the difficulties to something a bit funnier, but for this version we want to avoid confusing the players.

4.2.4 Performance graphics

Drawing a town and audience requires a lot of assets but is crucial to give the player the proper feeling for when the performance phase starts. To minimise the load (and for not having to create a house generator) we opted to only show houses "rushing by" in the foreground and fill most of the space with randomly sampled audience members. The audience members aren't animated but display a slight head-bobbing to further reduce workload. This technique works very well when the train is going fast since most artifacts disappear with motion, and it still looks reasonable for slow speeds.

4.2.5 Background music

The alpha version of the game contains background music also for the travel phase, which is made by us. Background musics for menu and performance phase are not yet composed by us, but in the final version of the game they will definitely be written by us (a first version is already done but not included in the alpha release). The main challenge for background musics was to compose them, since only one member of the team had experience with that. To write music we used a tool named BoscaCeoil and after that we mixed music with Cubase.

Another challenge with music is that in our game we have two different musics based on the phase of the game (one for travel phase and one for performance phase) and so, we have to switch between one music and the other. We did it simply using System.Windows.Media. MediaPlayer but it turned out not to work properly with different songs. The solution for the Alpha release is playing these two musics as Microsoft.Xna.Framework.Audio.SoundEffect but in the

future we will do some further investigation in order to solve the problem.

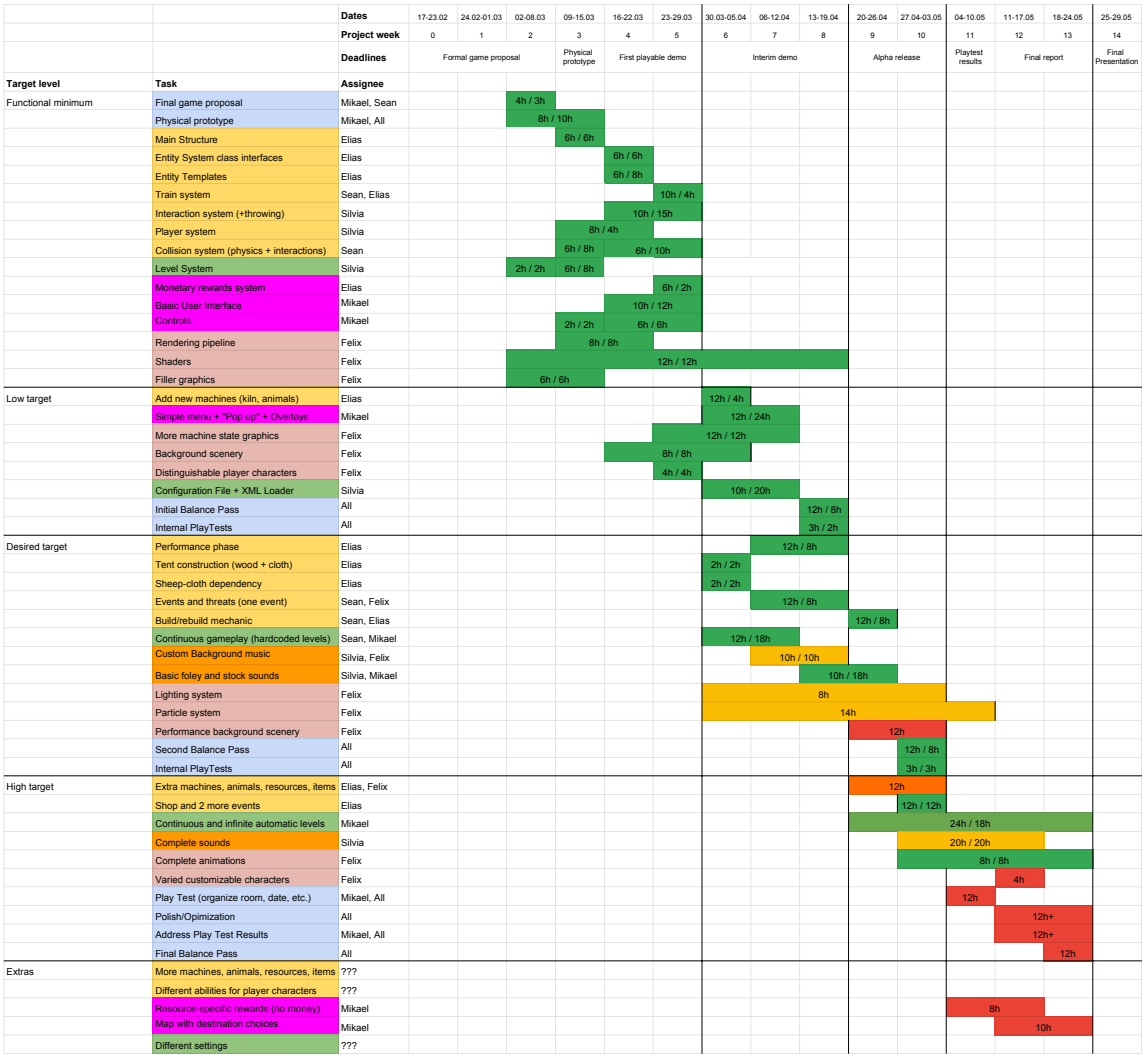
4.3 Future Work

4.3.1 Polishing and Optimization

The main task for the last weeks of the project is going to be the polishing and optimization of the game, to ensure the best player experience and give a more complete impression of our game. A big part of this is going to be on the visuals and sound effects side, but also a finer balancing and adding more variety to the gameplay (machine upgrades, player power-ups, etc.) play a role.

4.3.2 Extras

Even if the high target is not completely finished yet, in the next weeks we would still like to add some more animals (omnivore animal, e.g. bear), items (machine upgrades, player power-ups, etc.) and events. A scoreboard and highscore system could also make for a great addition.



		Dates	17-23.02	24.02-01.03	02-08.03	09-15.03	16-22.03	23-29.03	30.03-05.04	06-12.04	13-19.04	20-26.04	27.04-03.05	04-10.05	11-17.05	18-24.05	25-29.05
		Project week	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
		Deadlines	Formal game proposal			Physical prototype	First playable demo		Interim demo			Alpha release		Playtest results	Final report		Final Presentation
Target level	**Task**	**Assignee**															
Functional minimum	Final game proposal	Mikael, Sean			4h / 3h												
	Physical prototype	Mikael, All			8h / 10h												
	Main Structure	Elias				6h / 6h											
	Entity System class interfaces	Elias					6h / 6h										
	Entity Templates	Elias					6h / 8h										
	Train system	Sean, Elias						10h / 4h									
	Interaction system (+throwing)	Silvia					10h / 15h										
	Player system	Silvia				8h / 4h											
	Collision system (physics + interactions)	Sean				6h / 8h	6h / 10h										
	Level System	Silvia			2h / 2h	6h / 8h											
	Monetary rewards system	Elias						6h / 2h									
	Basic User Interface	Mikael					10h / 12h										
	Controls	Mikael				2h / 2h	6h / 6h										
	Rendering pipeline	Felix				8h / 8h											
	Shaders	Felix			12h / 12h												
	Filler graphics	Felix			6h / 6h												
Low target	Add new machines (kiln, animals)	Elias							12h / 4h								
	Simple menu + "Pop up" + Overlays	Mikael							12h / 24h								
	More machine state graphics	Felix						12h / 12h									
	Background scenery	Felix					8h / 8h										
	Distinguishable player characters	Felix						4h / 4h									
	Configuration File + XML Loader	Silvia							10h / 20h								
	Initial Balance Pass	All									12h / 8h						
	Internal PlayTests	All									3h / 2h						
Desired target	Performance phase	Elias								12h / 8h							
	Tent construction (wood + cloth)	Elias							2h / 2h								
	Sheep-cloth dependency	Elias							2h / 2h								
	Events and threats (one event)	Sean, Felix								12h / 8h							
	Build/rebuild mechanic	Sean, Elias										12h / 8h					
	Continuous gameplay (hardcoded levels)	Sean, Mikael							12h / 18h								
	Custom Background music	Silvia, Felix								10h / 10h							
	Basic foley and stock sounds	Silvia, Mikael									10h / 18h						
	Lighting system	Felix							8h								
	Particle system	Felix							14h								
	Performance background scenery	Felix										12h					
	Second Balance Pass	All											12h / 8h				
	Internal PlayTests	All											3h / 3h				
High target	Extra machines, animals, resources, items	Elias, Felix										12h					
	Shop and 2 more events	Elias											12h / 12h				
	Continuous and infinite automatic levels	Mikael										24h / 18h					
	Complete sounds	Silvia											20h / 20h				
	Complete animations	Felix											8h / 8h				
	Varied customizable characters	Felix													4h		
	Play Test (organize room, date, etc.)	Mikael, All												12h			
	Polish/Opimization	All													12h+		
	Address Play Test Results	Mikael, All													12h+		
	Final Balance Pass	All														12h	
Extras	More machines, animals, resources, items	???															
	Different abilities for player characters	???															
	Resource-specific rewards (no money)	Mikael												8h			
	Map with destination choices	Mikael													10h		
	Different settings	???															

Figure 4.3: *Updated Gantt chart (assigned time / actual time)*

5

Playtest

5.1 Playtesting Session

We decided to send our alpha demo together with a questionnaire to our friends and other "gamelab-ers". Sadly, due to the COVID-19 situation, we knew that few could probably play our game as intended: On the Xbox One, with a controller and with a group of 3-4 people. Most of the feedback we got were from single players with a keyboard.

This way of doing playtesting has its pros and cons: you could potentially reach a lot of people, but without organizing an event where testers could come and play in person many people might not have much time to play our game, especially in the last few weeks of the semester; on top of that, most can't play local multiplayer games like ours due to being stuck at home.

5.2 Questions and Comments

Even though most of the testers played alone we still managed to get some very useful feedback. Listed below the questions we asked and a summary of given answers.

What was your first impression of Silly Gilly when opening the game?

Most testers found the graphics and sounds very welcoming, but lacked clear explanations about the game goal, controls and game mechanics when diving into the game for the first time.

What is your impression now after having played the game?

Aside from some frustrating aspects like having to wait to lose when reaching zero coins and

tutorials not being very visible, the testers gave a positive feedback to the general game idea and design.

How many rounds have you played the game for?

On average, testers played 5-6 rounds each, mostly in singleplayer, with exceptional minima of 1 and maxima of 15 rounds.

How long did you play Silly Gilly and what made you quit the game?

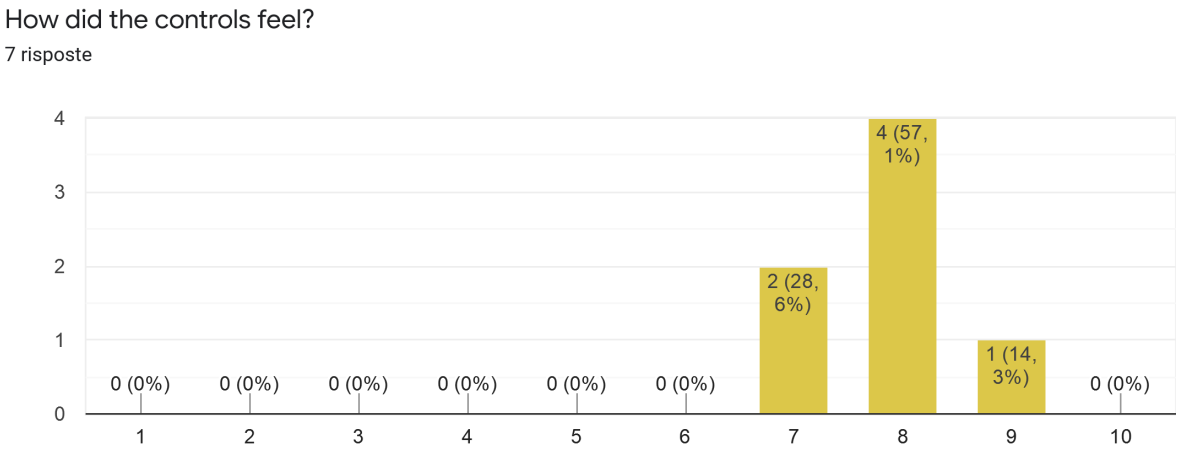
On average, testers played for 45 minutes to an hour. We observed two different categories of players. Some stopped playing due to getting frustrated by not being able to improve their performance gains and others due to the game becoming too stressful. Some testers learned quickly how to play the game, got very good at it and either felt that they saw everything the game had to offer or stopped playing due to lack of time on their side. Someone also left the game due to not being able to play with others, but specified that they certainly would have played more together with friends.

Did you play with a keyboard or with a controller?

As expected more than 50 percent of testers played with a keyboard. This was our main fear, since (as for most platformers) keyboard controls didn't feel very comfortable to us either.

How did the controls feel?

Even though most testers played with the keyboard we got a quite positive feedback for the feel of the controls.

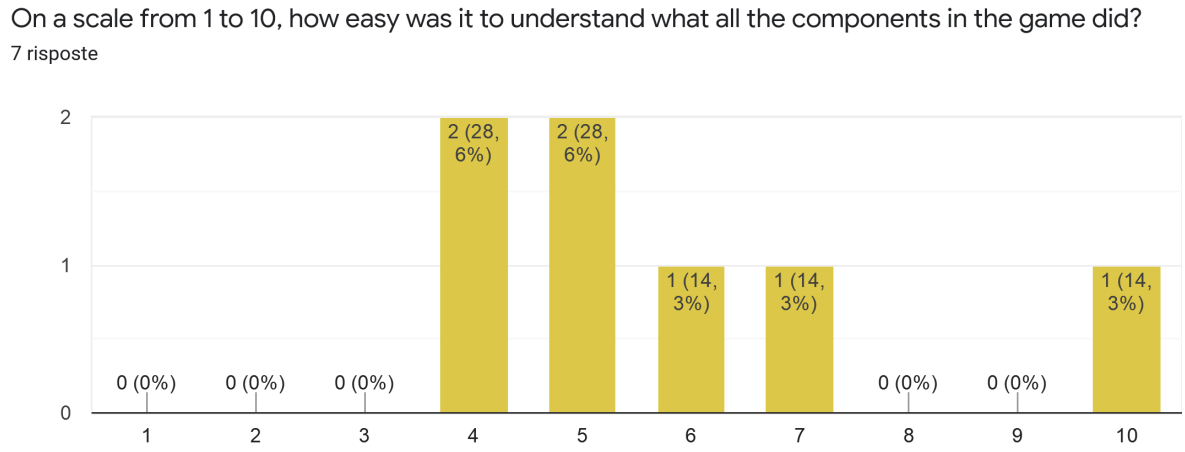


Was the objective of the game clear enough? What was, in your experience, the objective of the game?

At first most testers were confused about the main goal of the game, but after playing some rounds everyone managed to grasp it quite easily: Survive the longest while continuously upgrading your train.

On a scale from 1 to 10, how easy was it to understand what all the components in the game did?

Overall, the game mechanics were not very well explained and the tutorial was hardly visible for most.



How did you like the sound effects?

Everyone liked most of the sounds. Testers found that they really give a sense of speed for the train and found them helpful. Unexpectedly, some people specifically liked the giraffe sound.

Was there any aspect of the game you found frustrating?

In general, aside from the aforementioned lack of visible and clear explanations, testers had difficulties with jumping using the keyboard and found that having to wait for the train to stop after finishing all their money got frustrating and boring especially over multiple rounds.

What do you feel was missing from the game?

In general, most testers asked for clearer explanations and the option to rebind controls (especially for keyboard users). Someone also asked for a more variety, adding more phases and more items in the shop.

If you could change aspects of the game, what would it be?

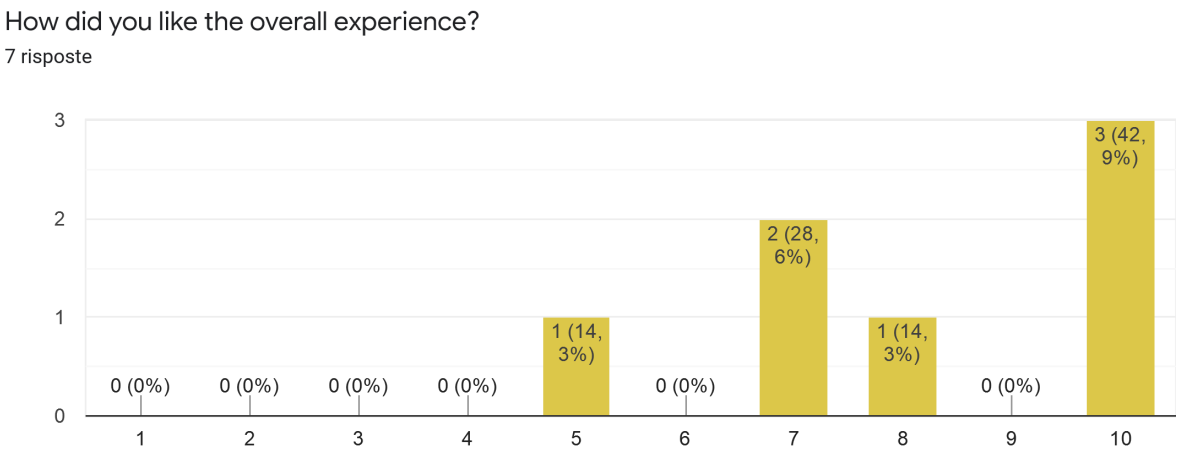
Aside improving the tutorials as already mentioned multiple times, testers wanted to change sprites, colors or sounds, while others asked for more animals, playable characters and changing specific control bindings. Someone proposed to remove cages due to the feeding mechanic being counter-intuitive, and to make it more obvious how the money gain mechanic works.

If you were to give this game as a gift, what kind of person would you give it to?

In general, testers would gift our game to people that play games like Overcooked or Unrailed, or to a group of friends to be able to play together. Most of the testers would gift it to teenagers and young adults.

How did you like the overall experience?

Overall we got fairly positive feedback, which is great!



General comments

Aside from positive comments, as expected someone wrote that he/she would have probably liked the game way more while playing with friends, since playing alone was not captivating enough.

It is important to note that we included online logging into our alpha demo, in order to obtain further details on all the runs played by the testers. At this point we received more than 70 log-files, meaning that our game was already played a lot. Most runs were very short, probably due to players still trying to learn the game mechanics, but over time we started getting much longer runs, meaning that at least some players grasped the game flow. The fact that a lot of rounds were played, while we did not obtain a lot of filled in questionnaires, could mean that this strategy we were forced to adopt due to the current situation is not the best.

5.3 Design Revisions

Even though we are still getting more feedback from people that did not have much time for our game during last week, the main things we have to improve seems to be clear. We obviously need to improve the tutorials and give clearer explanations on the game mechanics, thus we will probably decouple the tutorial mode from the actual game so that players have time to learn everything and then play it properly. On the same line, we will also add explanations for keyboard controls, since most players currently end up playing without a controller.

To improve player accessibility, for the next release we are also planning to distribute an executable to avoid having players to install the certificate. The installation process has turned off quite a few people from testing our game already.

Also, we got some comments on the "feed animals in cages" routine not being intuitive, so to avoid removing a very important mechanic from the game we decided to solve the problem by changing the cage graphics to something similar to a feeding pen.

From a mechanics point of view we are going to change the logic for fuelling the engine: At the current state, putting multiple pieces of wood or coal rapidly would waste some of their effects compared to waiting for the effect of each piece to finish before putting in a new one. To meet expectations we are going to allow players to put multiple pieces of wood or coal without losing anything.

In a similar way, having empty tents produce money without player interaction is counter intuitive. We are thinking of allowing players to perform as well, so there is an alternative way to gain a good amount of coins without having animals (mostly in the early game).

Lastly, to reduce player frustration we give them the option to skip to the game over screen if they reached zero coins and can not make it to the next performance, so that players playing at lower difficulties, where the speed decreases slowly, don’t have to wait a whole minute to see their score.

6

Conclusion

6.1 Final Results

6.2 Experience

6.3 Personal Impressions

Could be nice to do this per developer, so everyone can tell his opinion without having one person that writes others' ideas, since they could be misinterpreted