Exploring Game Design
Pitfalls through patterns

Experiences when making our first game

Gotland University
HT 2012
Project Report
Author: Jens Hedenskog
Supervisor: Troels Linde
Examinator: Steven Bachelder
Synopsis

The purpose of this report is to analyze what went wrong with the adventure game project called Fairytale, I started together with 4 of my fellow university students at Gotland University, spring 2007. My ambition with this report is to enlighten problems in game design that arose during the game development process in order to prevent others from making the same mistakes.


The results of the report show that redesigning already finished game features means a lot of troubles in relation to its dependency on other game elements. The key abilities of the main character were vaguely defined since the beginning of the project which caused problems with earlier designed levels whenever a new item was introduced. The terrain of the prior levels didn’t match the abilities of the new items, which forced changes to be made. The biggest mistake with this project was that finished game elements never were considered final.

My role in the project was the solo game programmer and co-designer. I shared the designing tasks together with Annika Fogelgren who also was the producer of our team. Albertina Sparrhult, Emma Johansson and Marie Viberg were our core graphic artists. Together, we created the Fairytale game.
# Table of Contents

Synopsis....................................................................................................................................................... 2

Table of Contents........................................................................................................................................ 3

Acknowledgements...................................................................................................................................... 4

Introduction................................................................................................................................................ 5

Main Body – Chapter I: Setting up the Game World................................................................................... 6

Main Body – Chapter II: Movement & Levels............................................................................................ 8

Main Body – Chapter III: The Blueberry Quest......................................................................................... 10

Main Body – Chapter IV: Player Feedback................................................................................................. 13

Main Body – Chapter V: Activating the Player......................................................................................... 15

Main Body – Chapter VI: Tools & Abilities............................................................................................... 17

Main Body – Chapter VII: Resources & Rewards...................................................................................... 20

Main Body – Chapter VIII: Enemies & Bosses......................................................................................... 23

Main Body – Chapter IX: Introducing the Narrative................................................................................ 25

Main Body – Chapter X: Redesigning & Killing Darlings......................................................................... 27

Conclusions.................................................................................................................................................. 29

References ................................................................................................................................................... 30

Bibliography ............................................................................................................................................... 31

Appendix .................................................................................................................................................... 32
Acknowledgements

I would like to thank my supervisor Troels Linde for his precise and accurate feedback. My mother’s encouragement during the period was invaluable to me. Thanks mom for always supporting me.

I also want to express my sincere gratitude towards Hans Svensson who helped me in many ways for the completion of the report.
Introduction

This report is written for me as a reflection, but also for other young game developers who are planning to start out a fresh big game project.

I will go through important phases of our game development process and analyze the problems we had by the format: Objectives, Challenges, Experience & Best Practice.

Objectives will explain what we wanted to do and achieve during the phase. Challenges will explain how we approached the problem and the different challenges we faced. In Experience I will write about my own experience and what we knew and didn’t know at that time. In Best Practice I will share whatever solutions I recommend others to tackle the problem with.

To figure out what went wrong in the development process and why, I will divide the problems we faced into different game design patterns. By doing this it becomes easier to overlook them and find solutions in other games using the same patterns.

Finally I will finish the report by summarizing my conclusions into straight forward tips and tricks I hope will be of use to other game students and young developers tackling with their own first production.
Chapter I: Setting up the Game World

Objectives: Define the basic components of the game. What perspective should we have? How many players? What avatar should the player control, and in what way?

Challenges: We set the format of our game to PC and to be played by only using the keyboard. We ignored the mouse because we wanted to project an old school feeling with simple controls, similar to the videogames of late 80’s and early 90’s.

In this report I will cover the difficulties and successes we faced during the Fairytale game project. I will explain the game according to game design patterns as defined by Björk, S. and Holopainen, J. 2005. I cite these pattern definitions within frames.

“Game World: The environment in which the gameplay or parts of the gameplay takes place is determined by the spatial relationships of the game elements.”

“Single-Player Games: Games where there is only one player in a game instance.”

“Third-Person Views: Players are shown the game world with a focus on a game element under the players’ control.”

The decision was natural for us. We wanted a single player adventure game, because we wanted to give each player the opportunity to play in their own pace. Multiplayer style usually increases the stress factor, so it didn’t fit well with our goal to emulate the feeling of having a fairytale book being read for you like when you were a young child. Third person view gives a good perspective of the Avatar and the surrounding environment and creatures.

“Alternative Reality: The game is described as taking place in an alternative reality in order to justify and motivate game elements, possible actions, and rules that contradict the ordinary laws of nature or the usual rules of social conduct.”
Choosing an *Alternative Reality* permitted us to defy common sense and nature laws which only would have crippled our limitless imaginary world.

However, the following patterns presented more of a challenge to us as we couldn’t make up our minds how we wanted their shape.

“**Avatars: Avatar is a game element, which is tightly connected to the player's success and failure in the game. In many cases, the Avatar is the only means through which a player can affect the game world.**”

“**Identification: The characters or parts of the game with which players identify.**”

The inhabitants of our *Game World* weren’t humans, they were fairytale creatures called Umps. The Umps lives in close relation to the forest and plants, as they are born from special flowers.

The game starts out in the Ump Village where the player is in control of one of these young Umps.

We decided quite fast who the main *Avatar* should be and its appearance. It wasn’t of any importance whether it was a boy or a girl. We left the *Identification* up to the player.

In the beginning of the development process we referred to the *Avatar* as “The Ump”. Later on in the process, the *Game World* had not only gone through significant enhancements of graphics, but also included narrative aspects (which I will describe further in another section), making the *Avatar* in need of updates as well.

After some thought, we decided the *Avatar* to be a girl. This because serious female character designs currently is a minority in the game industry. We also felt responsibility as game developers regarding the *Identification*, especially because we were making a children’s game.

The *Avatar* was in need of a new name. We had called her by many different names before but as time passed, we finally agreed on naming her “Lyra”. This based on its cheerful sound, but also the musical instrument lyre. The lyre was introduced into the game as a key item along with the narrative. Because of the great importance it gained in the *Game World* we made the *Avatar’s* name reflect that.
Experience: We had many ideas how the umps would look like and also who the player should be. I don’t believe there was anything wrong about this process. The Game World went through an enhancement period and so did the Avatar. It had to take the time it took. In the end, we were satisfied with the results.

Best practice: The Avatars are the vicarious subjects of games. If their design is poor, the rest of the game will also suffer from it, because all interaction with the Game World is often only permitted through utilizing the Avatars. Another important aspect is that we identify ourselves with them to some degree. If something happened to the Avatar we often say it happened to ourselves. Look at the target audience and think carefully about who they would want to be playing as.

Chapter II: Movement and Levels

Objectives: Define the basic Movement of the Avatar and base the Levels on this.

“Movement: The action of moving game elements in the Game World.”

Challenges: The Avatar could stand still, run, jump and crawl. We made use of this at some places when constructing the terrain. For example, inside the well, the player found herself in a water cave, but couldn’t proceed deeper inside unless she would crawl through a narrow passage. This kind of realized the Inaccessible Areas pattern.

“Inaccessible Areas: Inaccessible Areas are parts of the Game World the player can perceive but cannot currently enter, such as areas behind locked doors or sufficiently high ledges.”

I believe the Inaccessible Areas is one of the most powerful patterns, because it can be used to block the player, where she shouldn’t be, in a way that makes sense to the player. Another way to block the player would be Invisible Walls. Even though we hate them, I have to confess we put up some Invisible Walls.
Invisible Walls: Invisible Walls are impassible obstacles that limit the players' movement, but not vision, to areas that appear to be part of the game world.

_Invisible Walls_ blocks the player successfully, but it doesn’t make sense. Why can’t I go further? It looks like I can, but some invisible force is preventing me. A better way to block the player, especially in platform games, is to experiment with jump height.

We diligently made use of the jumping height to create _Inaccessible Areas_.

The player would receive a jumping shoes item early in the game. Equipped, it would almost double the jumping height. We made so that the first two _Levels_ would require this ability in order to be completed.

In the Ump Village and its vicinity we gave the player different quests she had to complete in order to proceed further in the game. Some of these quests can be perceived as _Levels_.

_“Levels: A level is a part of the game in which all player actions take place until a certain goal has been reached or an end condition has been fulfilled.”_ 

One of those was the Squirrels Quest. The player had to climb up to the top of a great tree in order to save a squirrel, which had eaten too many nuts so it couldn’t get down by itself. The tree wasn’t climbed by jumping on branches, but by jumping on falling leaves.

The falling of the leaves followed a pattern which made the _Level_ impossible to complete without equipping the jumping shoes, the distance between some of the leaves simply became too great otherwise.

Up to this point we had 3 quests and some areas strictly depending on the height enabled by the jumping shoes. Even so we realized we couldn’t go on like this. The height of the jump was just far too great. It would require every background in the future to be 50% higher than otherwise necessary, which is a lot counted in production time.

We took the hard decision to change the jumping height, and had to deal with the consequences that followed. After a transition period, we chose to completely remove the jumping shoes from the game, and made the default jump height slightly higher instead. I will describe this further together with our other items in the _Tools_ chapter.
The Movement allowed within the Game World had changed, which forced us to look over what currently existed and how it would be affected.

The pattern which the leaves fell in the Squirrels Quest had to be redone, as well as most of the Blueberry Quest. I will cover the Blueberry Quest in a separate chapter because of its significant importance to us being our first Level.

Experience: This was our team’s first big game project. We were very dedicated, enthusiastic and sadly a bit naïve. Who would have thought our game elements were so tightly connected with each other that a small change at one place would greatly affect another. It was hard to get an overview of the consequence each change implicated.

Best Practice: Have a clear shared vision of how the end product should function. Define the Movement limitations early on and always take it into account when designing the Levels. “Never” change basic game elements with high dependency.

Chapter III: The Blueberry Quest

Objectives: Creating a Level which stands out from the normal game play, keeping the player interested by its variation but also engaged in completing the quest.

Challenges: This was our first quest, maybe that’s why it was so important to us because it kind of set the standard of what a Level was in our game.

The player gets the quest from the Bakery Lady near the bakery in the village. Every year she bakes a cake for the yearly big festival, which happens to be this evening.

However, someone had eaten all the blueberries she had prepared, so she needs help picking blueberries to be ready in time.

Now it’s up to the player to save the festival!

Our initial vision of this quest was a slight stress feeling to raise adrenaline. We also wanted to convey a feeling of freedom by giving the player the chance to jump very high while reaching for the berries.

The Collecting and Delivery patterns made up the basic parts of this quest.

“Collecting: The action of collecting game elements from the game world.”

“Delivery: Delivery consists of moving a certain game element to another specified game element or place within the game space.”
To add that additional dimension of slight stress we chose to add a *Time Limit*.

“Time Limits: The Time Limit for completing an action, reaching a goal, staying in a certain mode of play, or finishing a game session has a limit based on either game time or real time.”

We implemented this as a clock (2 digit numbers) at the top-center of the screen, which started ticking down once the quest had started. Two other patterns were added as well:

“Memorizing: Games where players gain benefit by remembering facts about the game or game state.”

“Character Development: The improvement of characters' skills or knowledge.”

By completing the *Delivery* of the berries, the player would level up feelings thus the *Character Development*. Feelings is a unique feature for our game which I will describe this further in Chapter VII. We made the choice to place the berries at the exact same locations every session. This realized the *Memorizing* pattern because the player could remember where the berries were last time and therefore get a good chance to improve her result next try.

But most unique were the *Movement Limitations*. In order to get that free feeling of jumping very high, we changed the rules and gave the *Avatar* a temporary *Tool*; we called it the Charge Jump.

“The charge jump was a special ability only available by equipping the jumping shoes item inside the blueberry forest. The *Avatar* would then be able to charge up jump power on the spot by pressing and holding the jump-button. The longer time, the more jump power was charged. Upon
releasing the button, the Avatar would shoot high up into the sky before falling down again. This made it possible to pick the blueberries growing at the top of the bush.

It sounded good and we had fun testing it internally.

However, when we let our target audience try it out we received mixed results...

Relatively few people even noticed the Time Limiter we had set up, so they didn’t understand why the game would say “Times up!!” after a while. We figured this was because we didn’t add any special sounds or any introducing animation for it. Also we had never used that spot on the screen before so that might have been confusing too.

The Time Limiter was just the tip of an iceberg. No one understood the sudden change in Movement Limitations enabled by the charge jumping Tool. Normally, the jump button made the Avatar jump a short distance, but now suddenly the Avatar started behaving very differently, and all this without an explanation.

We had to refine this quest somehow in order to make it “playable”. We analyzed the Time Limiter, trying to find a substitute or other solution. We discussed it back and forth and what we finally came up with was a giant bug.

We called him Mr. Munchy. He functioned as an Enemy, which kind of ignored the Avatar, his only goal was to pick the blueberries one by one in his own pace.

This called for the Attention Swapping pattern, where the player had to not only look for blueberries, but also try picking them before the Enemy.

“Attention Swapping: Players have to move their attention between different parts of the game.”

Interacting with him would just make the Avatar bounce away. Also, the charge jump was replaced by platforms to jump on. That solution was accepted right away, because platforms are such a common element in games.

Later on, we removed the jumping shoes item from the game. This affected the overall jumping height because we reduced its power. This forced us to lower all the platforms in the Blueberry Forest, or else they would be unreachable. In other words, we had to redesign the blueberry Level two times before we were satisfied.
Experiences: Redesigning became, whether we liked it or not, a recurring part of our development process. The main reason for this was that we were learning by doing. We had to make certain mistakes in order to learn from them.

Also our graphic artist had grown a lot since the start, so our first graphic artifacts had become a bit dull compared to the latest. There were no discussion over this; they had to be redrawn according to our graphic artists.

Best Practice: The Movement Limitations should never be changed, if changed by for example equipping a specific Tool, then that Tool needs a proper introduction with a training ground for that. If not, the player will have no chance of knowing what it does or figuring out where to use it. I will describe Tools further in chapter VI.

To save production time, I suggest using mockup versions of the graphical artifacts, and play test a lot before finalizing things, or lots of work might go to waste. One note though is that experience never goes to waste.

Chapter IV: Player Feedback

Objectives: Let our target audience play test our game without help to see what is understood, and what actions in need of further guidance.

Challenges: Finding our target audience and proper location could have been problems, but our school Gotland University participated in several exhibitions in Stockholm and Visby in which we could show our Fairytale. We were kind of lucky too that most of the visitors were in our age range.

What we noticed was that players seldom knew where to go, and even got stuck at some places. One solution for this problem was to further improve the Clue pattern which already existed in the game.

“Clues: Clues are game elements that give the players information about how the goals of the game can be reached.”
Our Non-Playing Characters, like the Bakery Lady, had some hints and Clues to the player what to do, where to go and how to get there. It didn’t seem to be sufficient though so we gave more characters guiding dialogues. We also added a blinking X mark on the map item, with animated dots leading to the spot.

Only utilizing Clues as guidance wasn’t enough, what we really needed here was an important pattern in game design; the Consistent Reality Logic. This was closely related to our graphics.

"Consistent Reality Logic: Consistent Reality Logic governs that the game elements, the player actions and their consequences, and the game events are consistent.

For a game to be consistent means, first, that there are no contradictions or irregularities in the functioning of the game. For example, if the player can blow up a crate, the player should also be able to blow up all other similar crates. Another, more fundamental, layer of consistency concerns the degree to which our intuitive and natural ways of being in the real world are transformed into the metaphors used in the game itself. This means that all games have an internal logic that mimics reality or at least relates to how we understand reality through categories and relations."

Simple things, like; what makes a door a door? Sure, it’s often a rectangle shaped form at the bottom of a building, by interacting with it (opening the door), you can enter the building (assuming it wasn’t locked).

My point here is that we take certain things for granted, and apply properties to objects we’ve never been in contact with before; it’s our minds making assumptions all on its own. Well yeah we’ve chosen an Alternative Reality world so we could make the door skyrocket up into the air and explode, but that would have been pointless.

Let’s get back to the Consistent Reality Logic. If an object, looking like “something” acts in a certain way, then another object that looks similar should also act in “the same way”. So how do we distinguish doors, paths and platforms from the non-interactive walls and background forest?
We achieved this by using small graphic tricks like difference in color tone and shadowing techniques. It suddenly became very clear whether the *Avatar* could jump on the branch or would fall right through it (if it was part of the background).

If a character or animal had dialogues, we added a chat icon above their heads, telling the player that this object has a conversation stored to be heard.

The buttons used to play the game were also a puzzle. We solved it by printing out a piece of paper telling what each button did. But that solution was only temporary. We had to teach the player how to play the game inside the game, and doing that as clear as possible.

We added arrow icons pointing towards each door, smoothly sliding in under them. This hopefully told the player that this door could be entered by pressing the Up-key on the keyboard. To exit a house we used an animated down arrow.

**Experience:** None of us had any prior exhibition experience. Why would we? Luckily we had the chance to exhibit many times, so after time we grew more experienced and so did the decoration for our booth. Letting kids try out the game may be the most precise and thankful test group there is. Not mainly because they happen to be our target audience, but also because kids immediately let you know whether they are interested or bored, always speaking their minds out loud.

Feedback inside the game was something we didn’t really prioritize until after the demo testing. We were the only people playing it, so we knew how to play the game and what to do. We worked so close and concentrated on the game that we became blind to its flaws, we didn’t realize the importance of feedback.

**Best practice:** The ideal is a game that is self explanatory, where the player easily can progress forward by herself, without having to ask an exhibitionist for help or look up things in the game manual.

Feedback should be as simple as possible and always related to the *Consistent Reality Logic.* The worst thing a game can do is to fool the player where she thinks she has solved a problem, but the game logic produces an undesired outcome (often nothing).

I often hear the parallel “make it so a child would understand”. But according to my experience children are very quick to recognize patterns, so “make it so your parents would understand” would be more realistic.

**Chapter V: Activating the Player**

**Objectives:** We had defined our *Game World*, added some *Levels* and painted beautiful backgrounds for the *Avatar* to explore. But what was the player supposed to do while
moving across the wide areas we had created? We needed to activate the player somehow in order to not become bored.

**Challenges:** We could have made more use of platforms and the jumping *Movement*, but our game wasn’t a whole through platform game, but instead an adventure game which focused on exploring worlds and their creatures. We had some areas which made use of the platform element but we didn’t want it to become the main theme.

Probably the most common game element utilized to solve a situation like ours would be to deploy *Enemies* blocking the *Avatars* path.

> “Enemies: Enemies are avatars and units that hinder the players trying to complete the goals.”

The problem with *Enemies* was that we wanted to avoid unnecessary violence within the game. So we decided to solve this without them, at least for the time being.

Instead, because we humans love to find and collect stuff, we added fruits and berries along the way to be picked up.

> “Pick-Ups: Pick-Ups are game elements that exist in the game world and can be collected by players, usually by moving an avatar or Units in contact with the Pick-Up.”

> “Collecting: The action of collecting game elements from the game world.”

Great, the *Avatar* was able to collect different *Pick-Ups* like blueberries, apples and grapes, even rare sparkle fairies could be caught. But at some point, the player would undoubtedly start to wonder for what purpose they all existed, and how they could be used. This became another issue we had to tackle with.

**Experience:** None of us had any prior experience in making bigger games, so the development process was more or less trial and error in every aspect. We had to make mistakes in order to learn from them.
We had great ideas, painted them and put them into the game. But unfortunately we were unable to analyze how each artifact would affect the overall game play. This gave us flat backgrounds without level design thinking about Maneuvering the Avatar and without proper location planning where to put Pick-Ups or Enemies.

“Maneuvering: Controlling the movement of game elements in real-time games.”

**Best Practice:** Think closely about how you want the game to be played. What should the player do and what challenges will be required to overcome in order to proceed. Look at similar games, how did they solve the problems? What did they do well, and what didn’t work out? Why?

**Chapter VI: Tools & Abilities**

**Objectives:** Take a closer look on our Tools and items, reduce similar functions and make more distinct and unique solutions to give every item its own purpose and meaning.

“Tools: Tools are game elements that enable the players’ Avatars and Units to perform actions otherwise unavailable to them.”

**Challenges:** The way we had created items up to this point was that anyone of us in the team could come up with suggestions for an item. Nothing wrong with that, but what we did was that we passed almost every idea, without any evaluation how it would affect the already existing game elements. The graphic artists started working on the new artifact right away, and we soon had a new item in the game, often without a game mechanic purpose.

We didn’t take items, or in this case Tools aspects and needs into account when passing most of the ideas. Our main focus resided in the Fairytale reality, what is cute and funny, instead of their practical use within the Alternative Reality.

Ideas for items are splendid during the startup phase, but later on when half of the world is done, there are too many parameters to take into account. There is the Right Level of Difficulty, the Smooth Learning Curve and Consistent Reality Logic to consider.
“Right Level of Difficulty: That the level of difficulty experienced by the player is the one intended by the game design.”

“Smooth Learning Curves: Games designed to provide players with the possibility of smoothly progressing from novice to master.”

As I mentioned in Chapter III: The Blueberry Quest, assigning spots in the game world as training grounds for new Tools is a great way to realize the Smooth Learning Curves. There, the player can try out the new Tool in peace without being threatened.

However, we never utilized this, probably because we never prioritized Tools. It’s a pity, because if we had, we wouldn’t have had the problem with activation of the player. The terrain would be based on the Avatars Movement possibilities and the abilities gained by utilizing her Tools.

We figured something had to be done, and so we analyzed all the Tools within the game to figure out which to keep, which to change and which to discard.

We sorted the items depending on their function, dividing them into different Tools. What we came up with was a big mind map.
This gave us a clear view of all the items, making it easy to spot their mechanic functions within the game. We could also see which were redundant with similar scopes of use.

The Jumping Shoes were amongst the most troublesome item. My vision for it was to make use of the Inaccessible Areas pattern. The player can see a ledge, but not reach it yet with current Tools. So when finally getting access to that new Tool it’s up to the player to remember the locations where that ability was needed.

“Inaccessible Areas: Inaccessible Areas are parts of the Game World the player can perceive but cannot currently enter, such as areas behind locked doors or sufficiently high ledges.”

However, the Jumping Shoes were easily obtained as the first item in the game because it was required for completing the first Level. This more or less nullified its impact on the game, because there was never a time where the player was missing the Tool, neither looking forward to a solution, because the problem hadn’t been introduced yet.

After changes to the Blueberry Quest, we removed the Charge Jump Tool, but kept the item Jumping Shoes for a while. We thought that by equipping it from the inventory the player could jump higher. But after some testing we came to the conclusion that it was too tedious to open the inventory every time for just this simple feature. Instead, we chose to completely remove the item, and make the default jump of the Avatar the same height as the Jumping Shoes would have given.

What we noticed by sorting out the items was that some didn’t have a purpose in the game at all. This regarded mostly the food, like Cupcakes, Bread and Bag of Candies to mention a few. So how did we solve this? We started working with Resources which I will cover in the next chapter.

**Experience:** Giving items a clear purpose was quite the challenge because there was so much to take into account, not only from inside the game, but also how costly the change would become production time wise.

We had a solution for the rope where it could be used as a liana to jump over cliffs or where the tip would be launched straight up in the air, attaching itself to something enabling the rope to be climbed, thus reaching higher grounds and accessing new areas. All good ideas, but they would require new code and animations which would take up far too much time of our already busy schedule.
Best Practice: Looking at this in retrospect is a bit sad and frustrating. It was so easy for us to come up with ideas for new content to put into the game, but that turned out to be our weakness.

We many times had an idea for a new cool and funny item. Then we started painting it the way we wanted it to look. When finished, we looked for a place in the game to put it amongst the current existing quests and Levels. Or last resort; come up with new quests based on the item in order to give it a meaning.

This way of working should be avoided at all cost because it messes up the development process. As I’ve said before, key items and Tools has to be defined early on in order to have it in mind when designing Levels and providing a Smooth Learning Curve. Each of these key Tools should have their own unique ability, setting up for possible combinations within the Levels enabling interactive problem solving for the player.

Chapter VII: Resources & Rewards

Objectives: Find out what kind of Resources our game had, and how they could be manipulated. Also analyzing what kind of rewards we could offer the player.

Challenges: Before getting further into Resources, I present its pattern definition:

“Resources: Game elements that are used by players to enable actions in a game.”

What I mean by Resource manipulation is that by replenishing or exhausting them in different ways sets up for specific functions which could be translated into different Tools. For example: Picking up a heart (in many games) will replenish 1 unit of the Avatars life bar, this makes that heart a Resource. The player could instead pick up a health potion, which upon using it will restore 5 units of life. That would become a Tool.

What made our game so hard to develop was that it looked like a normal Role-Playing game like any else, but the inner game mechanics were very different. We had no life bar to refill, no extra Lives to add, (because the player couldn’t die), no weapons and no Enemies fight. So what elements did we have to play with?

“Lives: Lives can be defined as the number of chances a player has within a game session before it is terminated.”
We had developed a unique element for the game which we referred to as feelings. Initially we weren’t sure how we wanted it to function, but for our first demo it worked as a Progress Indicator which reflected how far the player had proceeded in the game.

“Progress Indicators: The player is given information about his current progress towards a closure in addition to the configuration of game elements involved.”

The feelings were divided into 5 different categories; Happiness, Patience, Courage, Empathy and Love. These feeling meters worked like experience bars which would gradually be filled up as the player completed different quests.

Depending on the quests’ appearance, the player would level up a specific feeling. For example; solving the Cave Troll’s riddles would level up Courage, while listening to the Old Tree’s stories would level up Patience.

We stored these feeling Resources inside the interface so the player easily could view them at all times.

Our feeling meters had the characteristics of the Container pattern:

“Container: Container is a game element that can store other game elements.”

When all quests in the game had been completed, all the meters would be filled up. It kind of represented the Avatar’s character growth during her adventures.

However, we came to change this feature later on when we added narrative aspects to the game (Chapter IX). The feeling Resources would be required in order to play the mystical instrument called the Fantasio.

This instrument became a key Tool together with the new narrative, and in order to play it, the player had to utilize her feelings. The feeling Resources changed form and became Renewable Resources.

“Renewable Resources: A type of resource of which more instances can be generated during game play.”

The feelings would gradually deplete when playing the instrument, and be refilled slowly over time once the player stopped playing.
This became the perfect Resource for our currently “useless” items. By letting the Avatar eat a blueberry pie, the Happiness meter would instantly be refilled, and the player was once again able to play “happy songs”.

The feeling meter had gained a clear purpose, and we could combine the feeling Resources with playing music, making items such as blueberry pies or cupcakes have the function of refilling the Container.

The problem with our non-function Collectables, the fruits and berries, was solved by making them a separate Resource by themselves, money. So in our Alternative Reality the creatures used fruits as currency when buying new things, like trading goods.

The player could for example exchange 20 blueberries for a blueberry pie in the village bakery, or collect enough apples to trade them for a lamp in the shop.

Experience: We looked at what we had and how it could be combined with new ideas, which gave birth to new features, like the feeling Container. The combinations made it possible to solve old problems like our Collectables and some of the Tools functions.

One of the hardest issues to tackle with was how to reward the player. The Legend of Zelda-series offers the player heart pieces, increased max amount capacity of different Containers such as bomb bags or arrow quivers, or sword and shield upgrades. General role-playing games have shops where new weapons and armors can be bought or found in hidden chests.

We had nothing similar. After some thinking, we came up with a couple of Tools which could work as rewards. We made boss keys, which were required in order to fight against the boss, and ribbons in different colors, which were used as passes to travel between different worlds in the game. However, both of these ideas were discarded later on mainly because their development time would become too costly.

The player could collect rare sparkle fairies, which was like a reward, but we never implemented a real purpose for them. The plan was to use them later when opening special feeling doors hidden throughout the game, one for each feeling. The doors would only open when the player had leveled up max amount of that feeling. Inside, the player would receive a wonderful treasure. Sounds exciting, but in fact we were only chasing our own tails again by adding new things without a clear purpose.

Best Practice: Figure out what kind of Resources the game have at its disposal, and then construct different Tools, items and rewards based on them.

For a reward to be rewarding, it needs to have a beneficial function to the player. If not by manipulating current Resources in a favorable way, then by providing new options regarding Maneuvering the Avatar, reaching earlier Inaccessible Areas or new ways to overcome an Enemy for example.
Chapter VIII: Enemies & Bosses

Objectives: We finally agreed to implement Enemies into the game, and the problem with activating the player seemed no longer to be an issue.

Challenges: After some thought, we changed our minds regarding the Enemies part. It didn’t hurt to have a few Enemies blocking the way, or forcing the player to outsmart them. Adding Enemies in itself doesn’t require violence in order to Overcome them.

“Overcome: This is the goal of the player to defeat an opposing force in a test, or a series of tests, involving attributes or performance of low-level actions.”

Inside the deep forest we added a couple of angry squirrels which would throw pinecones at anyone approaching them.

The player had to pass them in order to get deeper into the forest, but as soon as the Avatar came close, they would start hurling pinecones, which upon impact pushed the player away, and would likely push her down from the branches.

The only way to Overcome them was to play a harmonizing tune on the Fantasio, which would make them calm down, and eventually fall asleep.

Another successful Enemy we added was a couple of mushrooms, patrolling the forest ground. They kind of worked like Tools because they were used as stepping stones in order to reach the tree branches. Jumping on them would make the Avatar bounce high up in the air. This proved to become a fresh Maneuvering exercise for the player.

“Obstacles: Obstacles are game elements that hinder the players from taking the shortest route between two places.”

In order to block the player (in a good way) from just walking straight through the forest, we added Obstacles, in the form of winds, which pushed the Avatar backwards. This forced the player to take the route through the treetops and therefore having to Overcome the squirrels.

Inside the well, we added Obstacles in the form of rocks with sad faces on. A small and a big rock were placed apart from each other. When playing the Fantasio, their faces
would become happy, and they would become attracted, moving towards each other, making stairs the *Avatar* could climb and then continue on her quest.

I think this new thinking, how to plan the environment became an inspiration for the second world of Fairytale, the Sky World. There we added different looking clouds with different abilities. One would just disperse almost right away, another was a twin cloud which played tricks on the player by separating immediately making the *Avatar* fall down, and then slowly return, a third would temporary boost the *Avatars* jump height.

The adding of *Enemies* would of course make our minds wander off to bigger *Enemies*, such as *Boss Monsters*.

“*Boss Monsters*: A more powerful enemy the players have to overcome to reach certain goals in the game.”

We decided to finish with a *Boss Monster* at the end of each world of Fairytale. We set this as *Consistent Reality Logic* of the game flow.

Our *Boss Monsters* were a great success because of their intimidating appearance and engaging game play.

The first boss was a big mantis, which would hunt the *Avatar* all the way up a giant beanstalk. The player had to dodge its attacks by crawling, and then jump on its head, using it like a stepping stone, to get higher up to next platform.

The mantis would get angrier and angrier moving faster and faster. Until either he pushed down the *Avatar* or the player would reach the upper Sky World.

The second *Boss Monster* was a giant octopus. The battle was fought high up in the sky on a flying ship, the weather suddenly turned bad, a storm was coming, and out of cloud a giant octopus appeared. It spitted ink and hurled giant tentacles at the player. If the player would be hit by the ink, the *Avatar* would get all black and slowed for some time, making it harder to dodge the tentacles.

Same as earlier, the player couldn’t die. The only way to lose was if she would be pushed off the ship and fall down. Then the game would just resend the *Avatar* to the area prior the boss area. The penalties of the game weren’t that harsh.

The way to *Overcome* the octopus was to evade its attacks, then after a while it would start attacking the ships mast. It was now up to the *Avatar* to roll over a barrel to a
lever and jump it to hurl a barrel at him while he was busy. After a certain amount of hits, all the tentacles would have patches over them and the Boss Monster would retreat, and the player could proceed to the next world.

**Experience:** Making Enemies and Boss Monsters were lots of fun, and they filled the hole of action and challenge in terms of Overcoming an opponent we lacked before. We remade the deep forest, and made it higher, making it possible to add Enemies there. It wasn’t so costly production wise and the enhancements in game play experience made it worth it.

**Best Practice:** A good idea about how to help the player figure out how to Overcome a Boss Monster is to teach the player the skills or Tools needed to win even before fighting the actual boss. We could add smaller versions of the boss in the form of Enemies, which would mimic the Boss Monster’s Movement pattern for example. If the player figures out how to Overcome those smaller Enemies then she could reuse the same tactics versus the similar, but bigger, Boss Monster.

The goal should be to teach the player how to use the Tools needed and how to Maneuver the Avatar correctly through the Levels prior the Boss Monster, giving the player a fair chance to Overcome the boss by thinking for herself. If the player feels smart figuring out how to beat the boss, we have succeeded as game developers.

**Chapter IX: Introducing Narrative**

**Objectives:** The game had grown so much from just being a small village with some mini games to a world begging to get explored. We wanted to add further dept to the game by adding a story behind what had happened and what soon was about to happen.

**Challenges:** The narrative aspect made us put more effort into the dialogues giving every villager a specific personality. This wasn’t too painful to implement. Where the player should start turned instead out to be a bigger issue.

In the first demo, the ump named Lyra started out at her home with her parents side by side. After some time the father was removed and later on even her mother as well. Instead she had more like a godmother, Aska, the village elder who had been taking care of her together with her friend (and secret love?) Loki.

Aska had raised them by the fine arts of music. Every year the umps had a big celebration with lots of food and music performances. The festival was about to happen tonight.
“Cut Scenes: Sequences of storytelling where players cannot act within the game.”

Most of the current quests were able fit into the new story, but certain _Cut Scenes_ needed to be added as well as a night version of the Shepherds Field. At first we refused to have _Cut Scenes_ because we wanted the player to be in control of the _Avatar_ at all times. Adding a _Cut Scene_ would cause an _Ultra-Powerful Event_ where the player would lose control of the _Avatar_ until it had finished, which was something we wanted to avoid. Still, we changed our minds regarding this and finally agreed to add a few _Cut Scenes_ for the better of the narrative.

“The player would now start in Aska’s house instead. But that place didn’t even exist in the village yet, so we decided to add a completely new area at the outskirts of the village where Aska could live. This location took the spot where the old Windglade was (the location of the Squirrels Quest _Level_), which forced us to relocate a couple of areas.

We decided to put the Blueberry Forest next to Aska’s house in order to guide the player to the correct place (being the first _Level_) without getting lost on the way.

After we relocated one area, we noticed many of the others might as well be changed too for better guidance of the player. We connected them somewhat chronologically which helped the player to smoothly progress through the different _Levels_ of the village. When done, it actually fit better with the new game flow.

The Sky World on the other hand was much easier to change because it was still in the concept drawing phase, which made it cheap and painless to add small changes to the backgrounds in favor of the narrative.

**Experience:** Narrative, how the story is told in terms of scene, setting, music, sounds, dialogue and characters is indeed very interesting. I had no prior experience in putting narrative aspects into game design so this was very meaningful to me. Our storywriter was also new to putting story into games so we both learned a lot from this experience.

**Best Practice:** Set the story together with the design at an early stage like during the game concept phase in order to avoid feature conflicts and redesigning later on in the process. But in our case, we had no chance to do this because we never knew the game would become this big when we started out.
Chapter X: Redesigning & Killing Darlings

Objectives: Together with the narrative lots of changes followed. Some forced us to completely remove design and graphic artifacts while others required new solutions in design followed by graphics and code changes. Everything couldn’t be combined, and cuts had to be made.

Challenges: The challenges with narrative were to synergize it with the design, making all the content fit together. Sadly we were forced to remove some ideas from our design plans. One of those was the insect which used ribbons to travel around the world. We lost a great reward item, but what we gained was an even more powerful replacement as it was the mantis Boss Monster and the beanstalk.

In order to reach the beanstalk the player had to pass through the Toad Castle, and without the mythical instrument Fantasio, she would be unable to pass the castle guards. To make this Fantasio she had to collect 3 important parts. Those were combined with already existing quests. We already had a troll quest where the troll would ask the player three riddles, if she solved them all, he would now offer her one of his hair locks, which would do fine as strings for the Fantasio. We kept the quest untouched, but changed its reward.

A big issue was forming though in the team regarding the confusion of what should happen where, as design constantly was changing. New features and items were added while others were removed.

We had introduced a new key Tool, the instrument, which needed to be used frequently throughout the game. The order of the quests had to be updated fitting the new narrative and we had to analyze which item would be received where and what path should be blocked until a certain event had occurred as well.

We had to think in new ways to be able to connect the dots correctly, between what we had and what was required in terms of narrative.

The Troll of riddles was one of those quests. We had changed the reward to a key item, and by doing so, we had to deny access to this area in order to prevent the player from coming here before even receiving the quest of making the Fantasio. We solved it by placing a log on the road together with two lazy lumberjacks. They would block the path to the Deep Forest until the festival had ended.

The Squirrel’s Quest was also changed. Instead of saving a squirrel, we changed it to an umbrella which had been blown away and got stuck at the Great Tree’s top. This
umbrella became an important Tool because it enabled the Avatar to cross water by riding it like a boat. We restricted the falling of the leaves until after the festival. We then could use water paths to create Inaccessible Areas which could get accessed only when the player had received the umbrella Tool.

Experience: Redesigning, finding unexpected combinations and thinking outside the box of earlier solutions were things we had done before so it went better this time. Still it took a lot of energy to redo things over again and the time kept ticking away.

Some ideas were painful to let go of while others didn’t matter as much.

Best Practice: If you have to kill darlings, it’s impossible to keep them all. Begin by making a priority list for yourself, and let others have their way when it doesn’t matter that much to you. By giving up less important ones you are able to fight for those which matters most to you.

If you are considering a redesign of a feature, then you are more or less forced to split up the project into smaller parts which are easy to overview and check their dependencies on each other. The lower dependency, the less painful a redesign would be. However there are many hidden game elements to take into account, like the Smooth Learning Curve for example, which make redesigning considered a last resort.

Our biggest mistake was that we never considered anything completely finished.
Conclusion

We learned important lessons the hard way through trial and error. I here present some of the conclusions I came to realize from my experiences of the Fairytale project.

Let some things take the time it takes. The better foundation the *Game World* resides on, the less cracks there will be later on in the game development process.

The *Avatars* the player is in control of are very important, because we relate to the *Game World* through them. If something happened to the *Avatar* we often say it happened to ourselves. We *Identify* with them to some degree. If the *Avatar* is poorly made, then the rest of the game becomes poor as well. Further reading about the game’s fundamental design can be found at Chapter I.

Make sure that each *Level* has a purpose and that they follow a similar theme without changing the rules of the *Game World* without a logic explanation.

Decide early on if the game should have a story or not, and research different means of storytelling in movies and games today, so you can set the standard for your project. Once set, make sure to have it in mind when designing the *Levels* of the game. More about game story in Chapter IX and read *Level* related content in Chapters II - V.

Think closely about the goals of the game. What *Obstacles* hinders the player from success? What can the player do? And how does that help in *Overcoming* the problems faced? Where should the challenge lie?

What are the sub goals the player has to fulfill? Which rewards should be offered the player and where and when will they be received? The players *Tools* are strictly bound to the *Levels*, so make sure there will be no conflicts in their designs. Chapter VIII is dedicated to Enemies & Bosses, *Obstacles* and how to *Overcome* them. The game’s *Tools* and *Rewards* are analyzed in Chapter VI respective VII.

It’s also important to decide how to guide the player through the game. For what actions will it be necessary to add graphical help feedback, and what will the player understand without explanation? Read further about player feedback in Chapter IV.

Beware of changing design because everything in a game is tightly connected with each other, making a change at one place might cause problems elsewhere. The last Chapter X, covers the problems faced when changes of design becomes overwhelming.

A final advice is learning to accept that something is finished, and thus able to move forward.

Jenifer Niles
Bibliography


www.gamasutra.com
Appendix

The Prince of Moominland

Loyal:

The Book of Lost Songs

The Lumps and Their Songs

Appendices and Index