



Axel Backelin Marelus

Worktest for FutureGames

This work-test is primarily a look into my thoughts behind the design of my game: “Gizmo-Gears”. My expressive ability shines through in my level, gameplay and visual design, so a lot of the text focuses on those. Later in the text, I go through my collaborative experiences at the FutureGames preparatory course, and talk about my career goals as a developer.



Gizmo-Gears

Genre: A tile-based puzzle-game.

Target audience: Children (around age 10), casual gamers, non-gamers.

Target platforms: PC, console, tablet (Mobile if screen-size allows it).

Engine: Unity, using C#.

Description: A bright & friendly puzzle-game where you help two gears spin their way through dozens of puzzling stages. The game is single-player, and the players goal is to get both gears to a really big gear placed in the level. When the big gear is spun the player moves on to the next level.

Background: I made this game because I wanted a controlled environment, to apply level/puzzle design in a concrete way. This game was made specifically with the work-test in mind.

The game is meant to be accessible for casual audiences, and I've designed the pacing, visuals, and control-scheme to suit a person who is not used to games.

- The [visual presentation](#) is very non-threatening, with mild colours and low intensity. I've tried to stay away from a stereotypically "gamey" aesthetic (explosions, laser-beams, etc) so as to not deter anyone who might not see themselves as a "gamer". The fact that the player characters are gears provide some real life context for what is interactable, and what the players goals are. Putting faces on them provides some additional sense of character, making them more relatable.
- The [control scheme](#) is very simple and consists of moving the two player characters left and right using A/D and the Left/Right Arrows. Movement is also tile-based. Complicated control-schemes have a tendency of causing frustration in casual players, so keeping it really simple is important for accessibility. The game takes plenty of time introducing new mechanics, and regularly tests the player to make sure they're comfortable using mechanics, and understand them sufficiently.
- The [pacing](#) is gentle, but engaging, with short levels and a lot of opportunity to give rewarding feedback to the player. Things like victory-confetti and cute idle-animations would fit well into the game overall. However, nothing is ever stopping the player from taking a break to think or catch their breath.

Game Dynamics

One of the hardest things about making this game was guiding the player, and making sure they have the tool-set needed to progress, while avoiding holding the players hand too tightly through the experience. One of the ways I tried to counteract this was designing levels with different goals in mind.

1. Some levels are meant to introduce the player to new mechanics.
2. Other levels are meant to test the players understanding of a mechanic, making sure they're ready for future challenges.
3. Lastly, there are the levels that aim to have the player do puzzle-solving. The only actual goal of the other two level-categories is to make the puzzling levels possible. These levels are the bread & butter of the game.

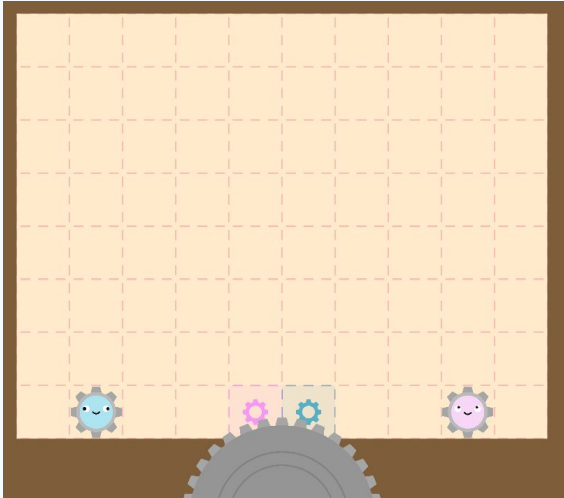
The puzzling levels were by far the hardest to design, and required a lot of research into what makes a puzzle engaging & satisfying. Some of the main sources of knowledge for my research were the plethora of [GDC talks](#) available on youtube, as well as [Mark Brown's \(gaming journalist\) youtube channel](#).

My Work-Flow

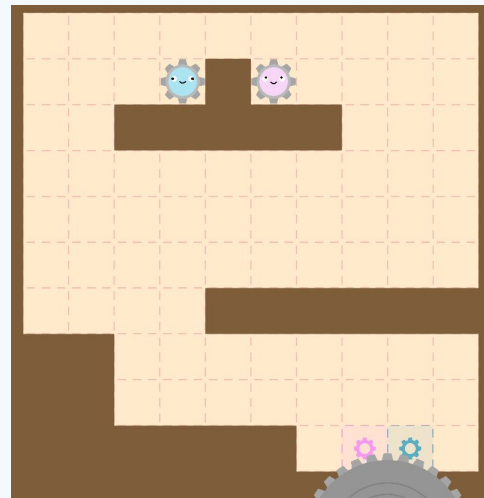
First, I would typically try to find out what the purpose of the level was. What was it that I wanted the players to learn, conclude or experiment with. Once this was clear, I sketched out my levels in photoshop using color-coded squares to represent the tiles. Once I had a design I liked, I quickly implemented it into Unity. From there on I focused on testing (both myself and others), tweaking and repeating.

The Levels

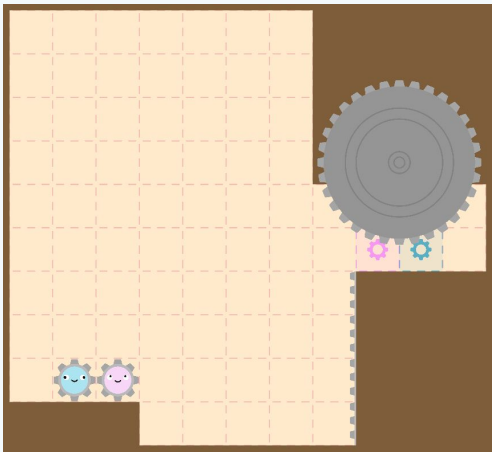
The first four levels of the game aim to introduce the player to the movement, goals, and basic mechanics of the game. The pictures below show the first 4 levels of the game, where the player learns about the win-condition, falling and climbing. The player is then tested on these mechanics, to make sure they understand them well enough to continue.



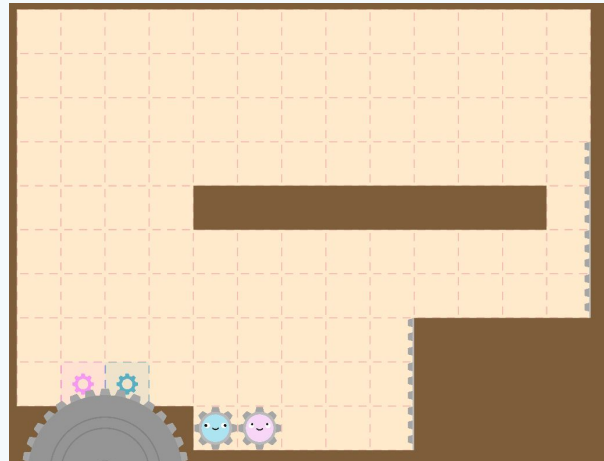
Lv. 1. The player is introduced to the game's win-condition.



Lv. 2. The player is introduced to falling.

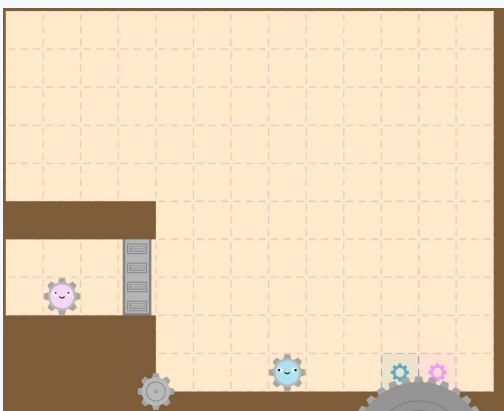


Lv. 3. The player is introduced to climbing. All the walls with grey notches on them are climb-able.

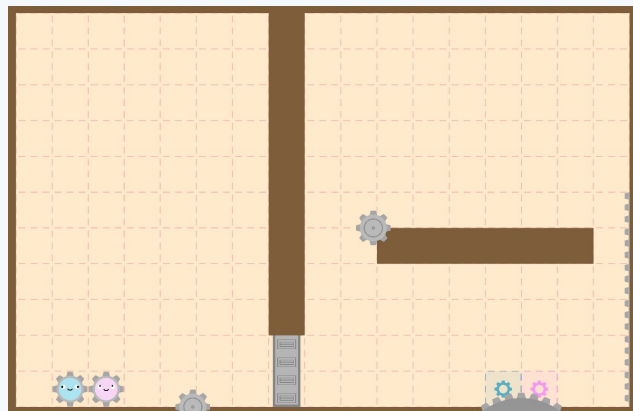


Lv. 4. Testing the player to make sure they understand falling/climbing.

The next two levels (5 and 6), introduce switches and doors. Level 6 also primes the player in anticipation of level 7, which is a slightly more challenging puzzle-level. The small grey gears hold the door(s) open while the players stands next to them.



Lv. 5. The player is introduced to switches and doors.

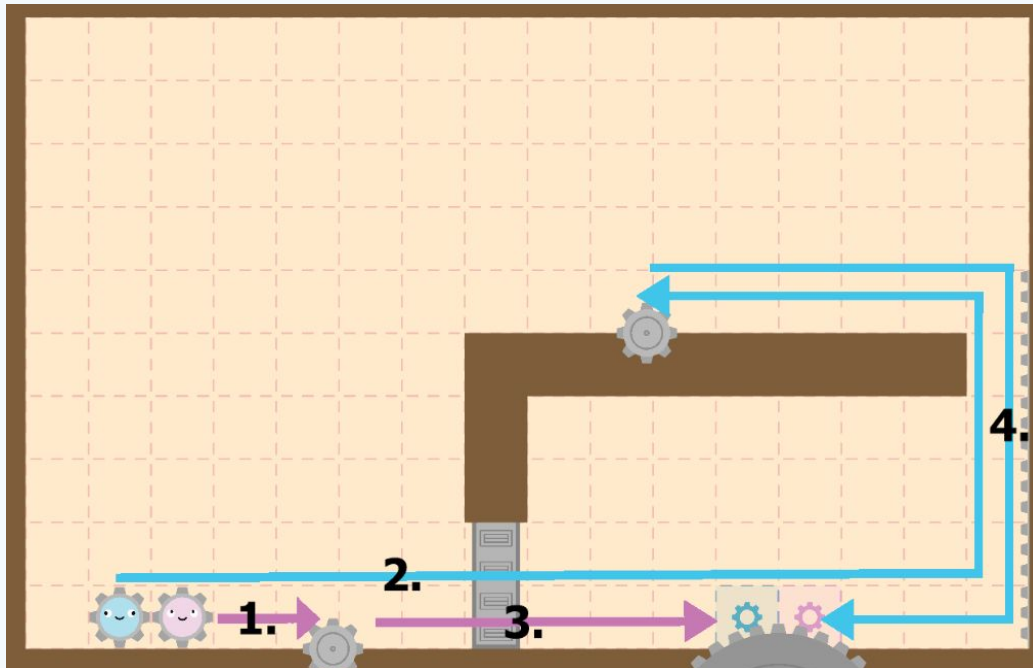


Lv. 6. The player completes a simple puzzle by using climbing/falling in combination with the recently introduced doors.

The pacing of a puzzle is important. They can't be too hard nor too easy, and the player should always feel a desire to participate in the puzzle. If the puzzle appears impossible at first glance, the player might be discouraged to even try. On the other hand, if the solution is obvious the completion of the puzzle may not be rewarding. Below is a **template for puzzle pacing**, which I applied throughout my level design. The goal of the template is to make an engaging and rewarding puzzle-solving experience for the player. The arrows in the images represent the player's train of thought/approach.

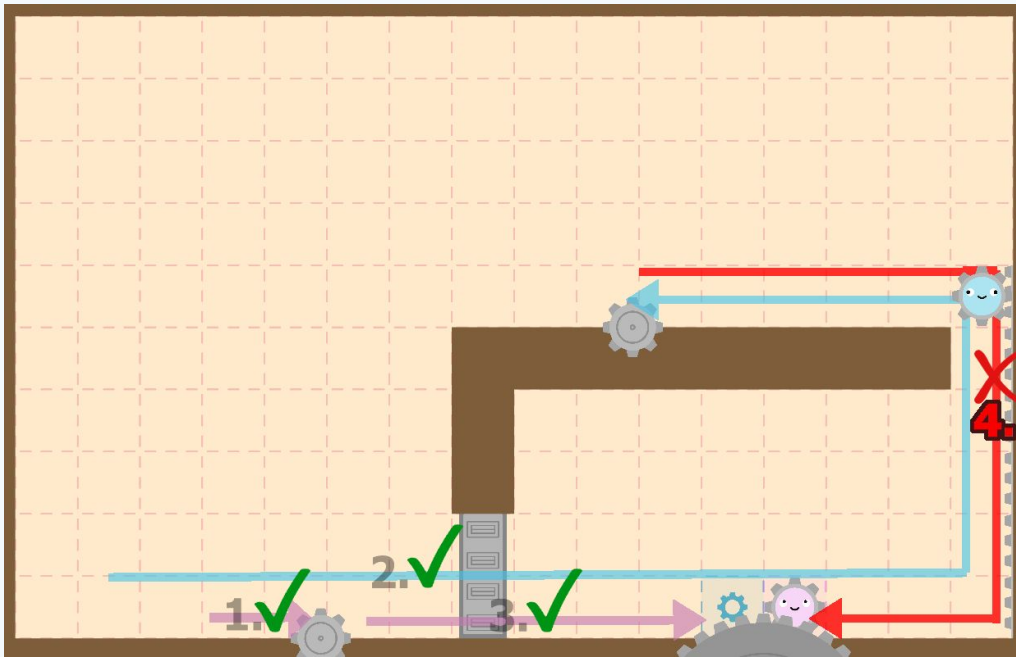
Level 7 - Puzzle Pacing

1. **The assumption** - At the start of a puzzle, a good way of engaging the player is to encourage them to take immediate action, often based on previous solutions. I believe a good puzzle shouldn't appear impossible from the start. The player needs to be encouraged to participate in the puzzle, so, for example, placing something like a switch to a door close to where the player starts might prompt them to immediately try to go through the door using said switch. The switch doesn't have to be the solution to the puzzle, and in many cases it shouldn't be. The switch is just there to get the player actively participating in the puzzle, to make them understand the mechanics of said puzzle, and eventually bump into the actual problem or "catch".



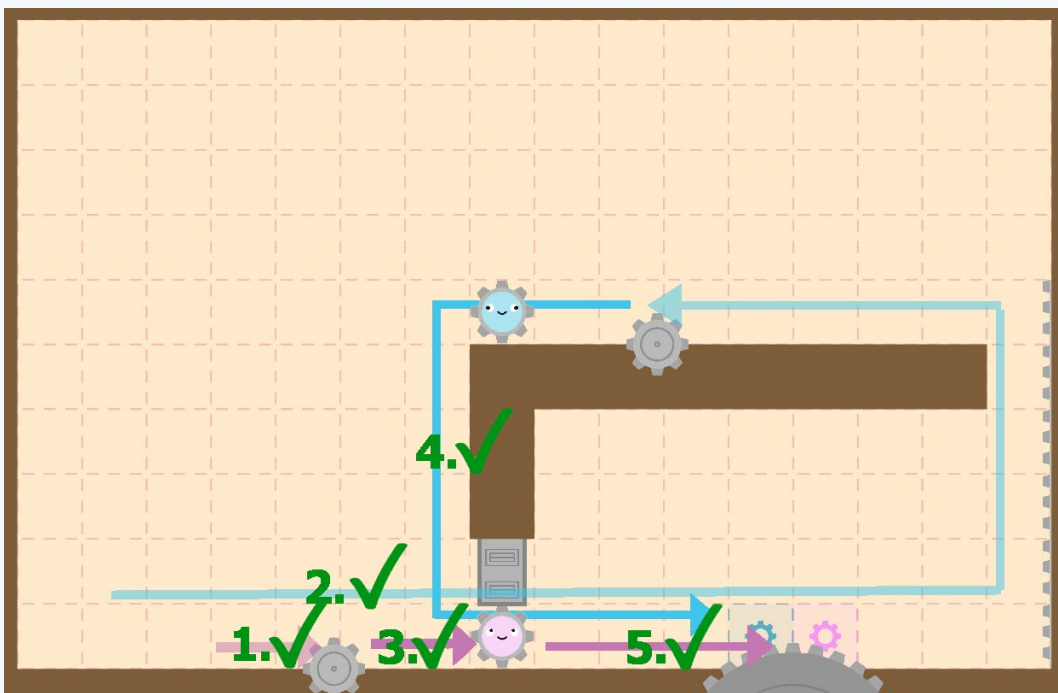
Lev 7a. The player assumes the solution based on the similar layout to the previous puzzle. The small cogs hold the door open while operated, the right hand wall is climb-able, and both the player characters must make it to the big cog to finish the level.

2. **The catch** - A catch is when the player realizes that her assumed approach won't work. Two or more conditions that are required for puzzle completion seem contradictory. The player cannot meet both conditions with her current understanding of game mechanics. This forces the player to re-evaluate the puzzle's mechanics in order to continue. To think outside of the box. A good catch should make a player feel stumped. This contributes greatly to the dramatic arc of a puzzle, making the solution of the puzzle that much more rewarding.



Lev 7b. The player has assumed that they can go back down the climb-able right hand wall. They can't. This new insight about the games mechanics contradicts the players initial plan. The puzzle might seem impossible to solve. This frustration spurs the player to find the correct solution.

3. **The “A-ha!” moment** - The moment of realization, when the player has successfully reached some new insight of the games rules or mechanics. In a well-paced puzzle, this feeling can be hugely rewarding.



Lev 7c. The player approaches the puzzle from a new angle, discovering a new way to interact with the game. (In this case, standing under the door in order to prop it up without using the switches.)

To try to make the “A-ha!”-moment really effective I believe strongly in these two guidelines:

“The player has to think of the solution by themselves. Don’t leave hints, or put training wheels on the player. Rather make sure the player has the tools to think of the correct solution.”

“An elegant solution isn’t hard to execute. The challenge should be thinking of it, not executing it. Simple (but clever) solutions make you feel extra smart!”

Play-testing

As I started attending the preparatory course, a lot of opportunities opened up for me. For this project in particular, it meant I could do playtesting. It was a much more engaging and enjoyable way of evaluating the design of my levels, and there were a lot of refreshing perspectives on the gameplay, and levels of the game.

A. One really common observation were how players assumed the characters could interact with each other. Players thought it would be intuitive for the gears to spin each other around, like real life gears do. In the tested version of the game, they couldn't.

B. There was some frustration with the pacing of the game. The game's early levels are very easy, and at this time moving quickly through levels was not very convenient/smooth. For a slightly more experienced player, the early levels weren't as much of a challenge as just an introduction to mechanics.

C. A common question was whether the game was meant to be multiplayer. Since the game featured two player characters, it was easy for players to assume the game was meant to be played as two.

Other than these three points, there were a lot of small tweaks to mechanics and levels as a result of players stumbling upon puzzle solutions, breaking something, etc.

The changes I made based on feedback

A. An interaction between the player characters was something I wanted to include when I started making this game. This functionality didn't end up in my prototype, because of the time needed to code it. However, the play-testing and feedback made me realize how important this really is, both for the games sense of character, and it's mechanical depth. It is something I would definitely take the time to get working, if I were to continue development.

B. The pacing got reworked hugely during playtesting, as well. The movement speed was increased, and moving the cogs fast along the levels was made more convenient. The goal of these changes was to maintain the gentle pacing of the game, but removing a lot of the frustration caused by the games early levels being very easy. One level was removed from the game entirely. The level was not crucial to the game in terms of progression, and it was (for some players) detrimental to the early-game pacing.

C. The game is, and has been made as a single-player experience. However, playtesting made me realize how easy it is for players to assume that it is a multi-player game. I don't feel I need to change anything about the core game in regards to this, however it would be crucial to be clear about the intended play in my in-game presentation and my marketing.

The Scope

If I were to continue development on the game, these are some things I would like to rework & implement.

- Re-work the code of the game.
- One additional mechanic: a gear that rotates the level by 90° when the player spins it.
- Meaningful interaction & collision between the player characters.
- More organic animations, softer shapes.
- Character sounds. Background music.
- I would like for the game to be around 30 levels long, and take about an hour to play through.
- I would like to spend at least half the time I spend on development on playtesting & polishing.

[>>Level Concept/Sketch for the Rotation Mechanic<<](#)

Thank you for reading!