Attendant, the OS

The case study represents the challenge of building a Linux-based operating system for gambling machines.

Due to NDA restrictions, I can't showcase screens, flows, prototypes, etc.
Nevertheless, I can share my experience and highlight the challenges I faced.

mailto: anton.vasyliev@gmail.com contact: https://www.linkedin.com/in/vasyliev/ web: https://vasyliev.design/

About Project

Novomatic is a large international gaming company headquartered in Gumpoldskirchen, Austria. It is well-known for manufacturing various gambling machines. In Poland, where I have been working for 7+ years, there is an assembling and manufacturing point for gambling machines from different global markets, such as the UK, Italy, and Austria.

The precursor for this project is the ongoing global chip shortage that started in 2020. The company's gaming machines were originally built with a specific architecture adapted for Windows-based operating systems. To overcome limitations posed by the chip shortage, Novomatic decided to make a Linux-based system, which is more adaptable to different available architectures.

The second thing is the overall user experience of the previous solutions, which resembled a computer BIOS designed by programmers.

These two factors sparked the idea of creating a new, consistent, and usable operating system.

Novomatic gambling machine

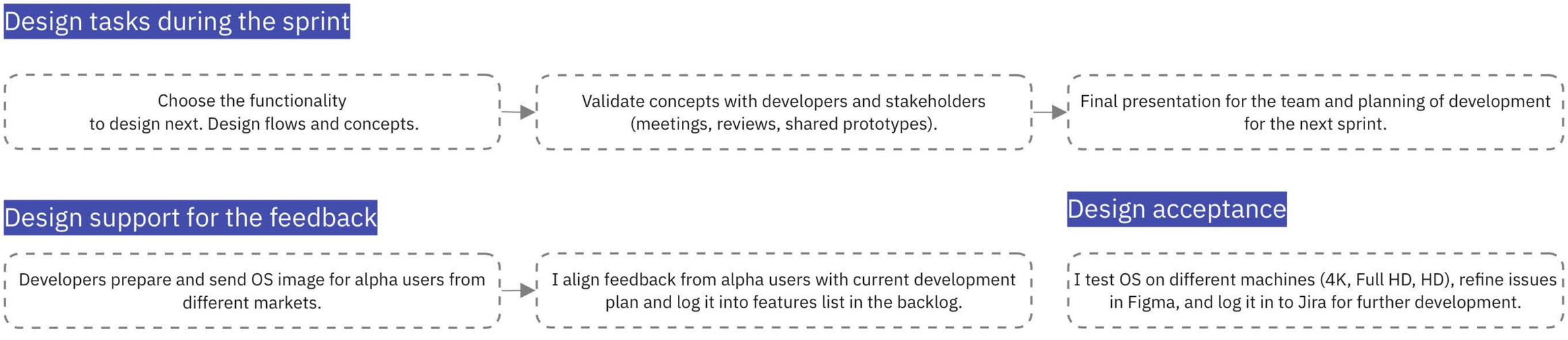
My Role

I have served as the lead designer for this project since its beginning (2019), and I would like to delineate my responsibilities from the project's outset to the present:

- Organising and conducting the discovery phase, gathering requirements from casino workers to align them with stakeholders' vision.
- Delivering interaction design and UI design assets. Initially, I created flows using Sketch and compiled them into documentation in Miro. Now, we utilise Figma for documentation.
- Performing design acceptance tasks. I prepare reports while conducting OS testing on various gambling machines. These reports highlight visual and interaction issues for further consideration and resolution.
- Designing interactive prototypes for specific applications, which are shared with stakeholders in Austria and the UK. These prototypes serve to test concepts and gather feedback before the development phase.
- Collaborating closely with the development team to overcome the challenges inhered by the legacy environment. This environment comes with numerous dependencies and requirements from games.
- And the main task is to be a user advocate, highlighting user requirements during functionality planning and aligning them with solution restrictions.

Process

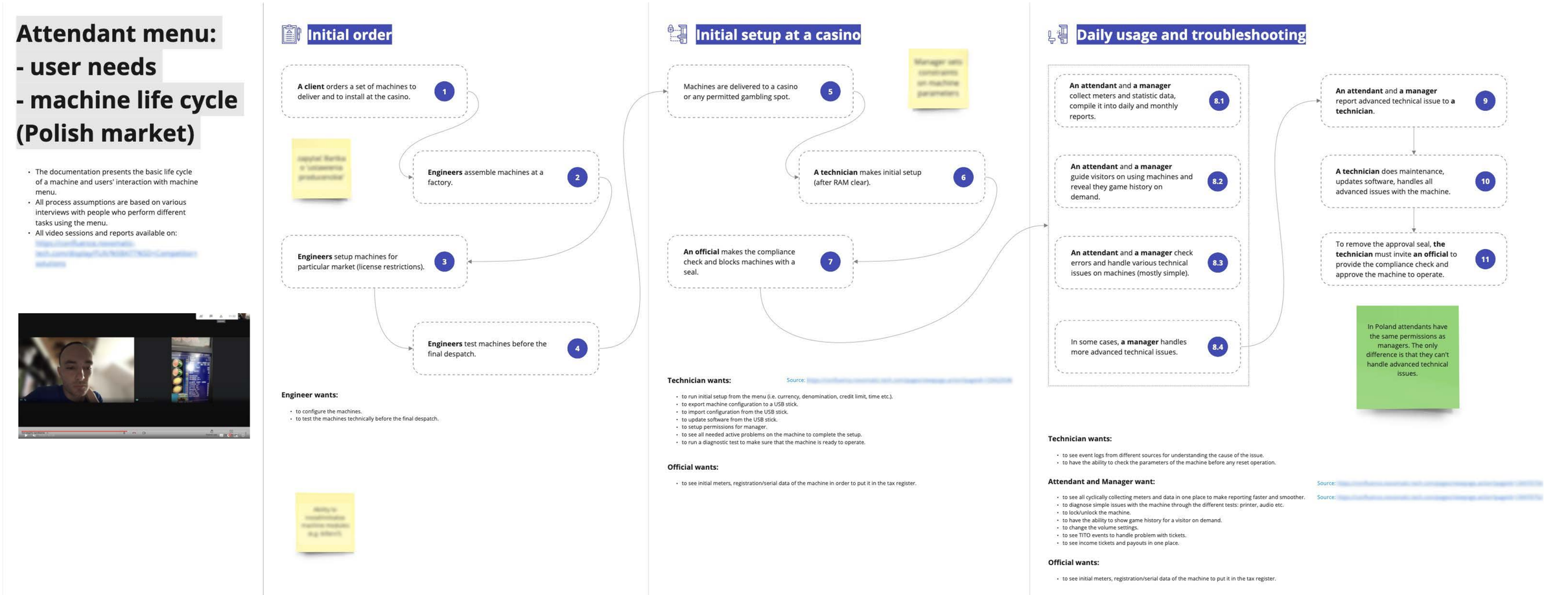
- The discovery phase took approximately two months and involved extensive research and data collection from casino workers and technicians. This information was gathered through online interviews with casino staff.
- The next phase involved establishing a connection with developers. Since the OS is based on a legacy version with several limitations, we organized a series of meetings to align our new vision with the existing constraints. Overcoming technical challenges in the UI is a daily part of this project.
- The development process follows an Agile methodology. As I prepare designs for specific functionalities, developers are prepared to implement them for the current or upcoming sprint, allowing us to conduct user tests quickly.
- One of the most distinctive aspects of our process is the local or remote testing on machines with stakeholders from various markets. We refer to this as internal and external testing. Both sources of feedback are invaluable.



Events during two weeks sprint

The discovery phase

The primary objective of the discovery phase was to compile user flows from different markets and various user roles. I conducted interviews, both onsite and offsite, with casino attendants, technicians, and managers. These three roles are core users of the Attendant OS. As a result, we gained a clear understanding of the machine lifecycle and user needs. Despite the Polish market being the smallest, it became the only market available for onsite collaboration during the COVID pandemic.



User needs and machine life cycle visualisation (Polish market)

Attendant, the OS

The Attendant OS, also known as NLX (Novoline Linux OS), is built on the Debian Linux distribution. The user interface is rendered by Chromium and can extend across up to four screens, depending on the model. The primary input is via a touchscreen, with mouse and keyboard as optional features. The OS comprises three key software modules:

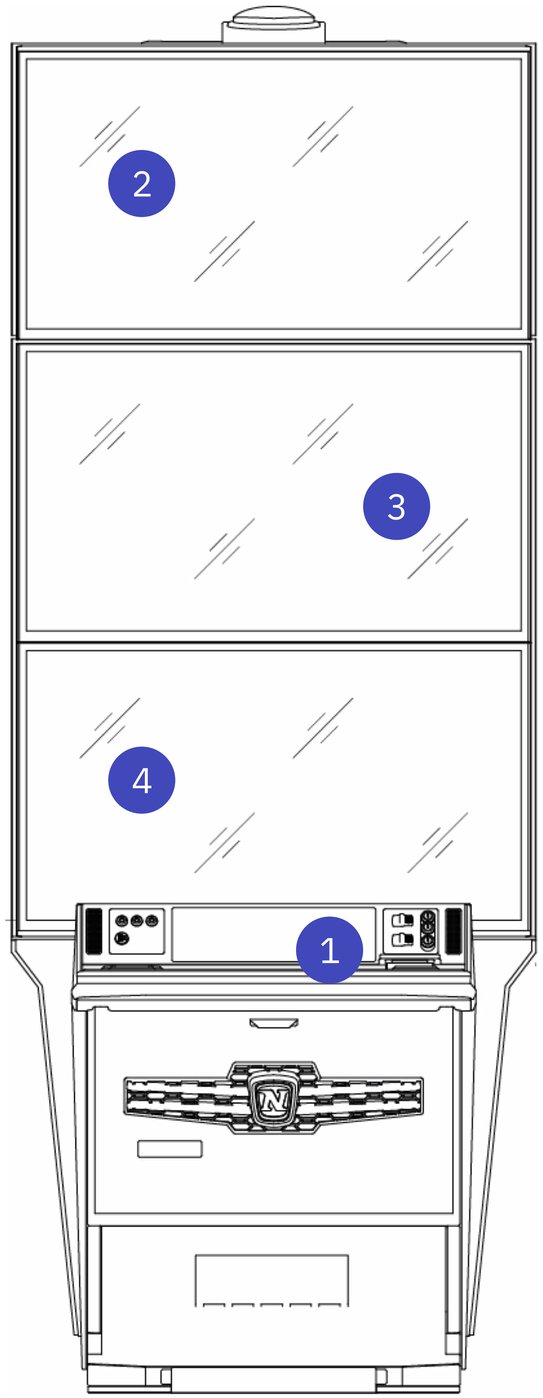
Attendant Module: Assisting casino attendants, managers, and technicians in daily tasks such as device diagnostics, accounting, game history checking, play statistics, machine and game settings, etc.

Game Manager: Facilitating the management of packages and certificates, with 80% dedicated to games.

Game Server: Encompassing roulette and automated table server settings and control.

Working areas for four screens include:

1. Default screen with the initial menu and apps (typically touch-enabled in 99% of cases).
2. System status screen displaying machine, network, and troubleshooting information (occasionally touch-enabled in 1% of cases).
3. Screen for apps and loading status (touch-enabled in 40% of cases).
4. Additional screen for apps (touch-enabled in 80% of cases).



Four screen model

Challenge #1: screens

There are two challenges with the slot machine screens.

Firstly, these screens come in various resolutions and aspect ratios. Some models feature 90° rotated curved 4K screens, resulting in non-standard breakpoints and vertical content distribution. Some models have notably different screen widths, resembling the scenario of connecting a 14'' laptop to a 55'' TV screen. It's similar to designing for an oversized smartphone, which I find both fun and challenging.

Secondly, some models may have resistive touchscreens. These screens alter user interaction compared to capacitive screens, which offer better support for multitouch actions and gestures. When designing scrolling for a table or drag action, I always consider how it will work well on resistive touchscreens. In some cases, I add additional controls or states for resistive screens to overcome limits. After deploying on the machine, the process includes setting clickable hotspots and tuning scrollability in cooperation with developers. It's crucial because the same device can have both types installed, and the experience should be as seamless as possible.

Challenge #2: machine, game settings and dependencies

It is my favourite part when it comes to challenges. The settings (both games and machine) application has a simple look – just a plain table with interaction elements.

The true complexity lies in real-time updating and communicating parameter dependencies to users, especially when changes can have cascading effects. To handle this, the application employs a 'Parameter Changes' mechanism. This feature provides a summary of user modifications along with their potential consequences before applying the changes. Additionally, there's a message centre that alerts users about changes that are not acceptable for machine operation.

Moving to the next layer, Novomatic machines are a multigame solution, sometimes running up to 70 games on a single machine. The new solution encounters grouped parameters from the game settings section. Each game comes with a set of parameters, contiguous with other game parameters but subject to different limitations defined per game module. This intricacy means that bulk editing of game parameters can result in complex validation outcomes displayed in the 'Parameter Changes' summary.

That was the best interaction design challenge for me.

Challenge #3: team and legacy

These two challenges were initially distinct, but in this scenario, they intertwined. The backend team, responsible for maintaining the legacy version, had a bias towards user experience due to their constant handling of user issues from diverse markets, leading to a divergence in vision. Back in 2019, me and the frontend developers were new to the team, and we found it hard to connect with the ‘legacy crew’.

Numerous challenging meetings and unsuccessful sprints ensued until we discovered a pathway to establish mutual trust. The turning point arose during demo sessions with users. I designed prototypes using Figma or replicated them on actual machines to reflect our vision for showcases and gather feedback. Presenting reports to the entire team helped validate or refute ideas, resolving many uncertainties.

Moreover, the legacy backend, essential for the new version, offered limited communication mechanisms. One method, REST, proved more flexible and aligned with our preferences, allowing application styling. Conversely, YAML, the second method, imposed significant constraints on design.

That was the communication challenge for me.

Summary

My key takeaway from Attendant is about the value of relationships with a team.

Empowering teammates to contribute to the design process by sharing and discussing concepts and user feedback is incredibly rewarding. It helped me overcome a lot of design refinements and cleared the way to good feedback from our users.

Thank you, contact me on anton.vasyliev@gmail.com