UNIVERSIDAD TECNOLOGICA DE CHIHUAHUA

Software development and management

SENTINEL - Technical Documentation

Integrative Project of 10th quarter of Engineering in Software Development and Management.

Presents:

Prieto Lozoya Brayan

Tarango De La Vega Fernando

Torres Borunda Ilse Paulina

Index

Introduction 2
The use of technologies applied during the project 3
Technical requirements to run the videogame 7
Common issues 8
Conclusion 10

Introduction

This document is created to give a technical background about how Sentinel, the videogame was developed, including the technologies used during the project, the requirements needed to run the videogame, the process, and its common issues during the development of the project.

On the first section will have a brief description of the technologies used on the videogame, such as Unity, Blockbench and C#, then for the landing page we will have Ionic, Angular, TypeScript, MongoDB, NodeJS, and JavaScript, and also we have GitHub which is used in both. This section is important, even though someone who is not an expert on game development, this theorical framework helps to have a better idea of what creating a videogame needs.

Then the section for which are the technical requirements in a computer to run the videogame is next, here, are just the basic requirements such as RAM, Windows version, processor, graphics and storage, so the user can have a better experience with the game having the minimum requirements.

Then the common issues experienced during the development of the game will be described along with the solution of them, this section helps the reader to understand what involves creating s videogame from cero and the possible issues it can affront.

And finally the conclusion of the whole team regarding their experience during this project, most difficult part and what was learned.

The use of technologies applied during the project

The following technologies are used on the development of the videogame and its design:

Unity

Unity is a cross-platform game engine and development platform that is used to create, deploy, and operate interactive 3D, 2D, VR, and AR experiences. The Unity engine provides a wide range of tools and features that enable game developers and designers to build games, simulations, and other interactive applications for desktop, mobile, and console platforms.

Blockbench

Blockbench is a free and open-source 3D modeling software specifically designed for creating models and textures for use in Minecraft. The software features a user-friendly interface and a range of tools and features that make it easy to create, modify, and export 3D models and textures for use in the popular sandbox game. Blockbench provides a powerful and intuitive framework for creating custom 3D models and textures for use in Minecraft, making it an essential tool for modders and game designers looking to add unique content to the game.

C#

C# is a high-level, object-oriented programming language developed by Microsoft as part of the .NET framework. It is designed to be a modern, efficient, and easy-to-learn language for developing a wide range of applications, including desktop and mobile applications, games, web services, and more.

C# is based on the C programming language and shares many of its syntax and features, but it also includes a number of advanced features and concepts that make it a powerful language for modern software development. These features include garbage collection, support for generics, and the ability to create and manipulate objects using classes and interfaces.

The following technologies are used on the development of the landing page and its design:

Ionic

Ionic is a popular open-source framework for building cross-platform mobile applications using web technologies like HTML, CSS, and JavaScript. It is designed to simplify the development process by providing a range of pre-built UI components, plugins, and tools that make it easy to create high-quality, feature-rich mobile apps.

Angular

Angular is an open-source framework for building web applications using TypeScript, a superset of JavaScript. It is designed to simplify the development process by providing a range of pre-built components, tools, and services that make it easy to create high-quality, feature-rich web applications.

TypeScript

TypeScript is a popular open-source programming language that is a superset of JavaScript. It is designed to improve the development process by providing a range of advanced features and capabilities that are not available in standard JavaScript. One of the key features of TypeScript is its strong typing system, which allows developers to define the types of variables and data structures used in their code. This helps prevent common errors and bugs that can occur in JavaScript due to its loose typing system.

MongoDB

MongoDB is an open-source NoSQL database that is designed to be highly scalable and flexible. It is designed to store data in a document-oriented format, which allows developers to work with data in a way that is more natural and intuitive than traditional relational databases. MongoDB also includes a range of advanced features and capabilities, such as support for indexing, aggregation, and advanced query operations, that make it easy to work with large and complex data sets. Additionally, MongoDB includes a range of tools and services for working with data, such as a powerful query language, an interactive shell, and a range of drivers for popular programming languages like JavaScript and Python.

NodeJS

Node.js is a popular open-source server-side JavaScript runtime environment that allows developers to build scalable and high-performance applications. It is built on the V8 JavaScript engine, which is also used by the Google Chrome browser, and allows developers to use JavaScript for both front-end and back-end development.

Another important feature of Node.js is its flexibility and ease of use. Because it is based on JavaScript, developers can use the same language and tools for both front-end and back-end development, which can simplify the development process and reduce the learning curve. Additionally, Node.js has a large and active community of developers, who contribute to a range of open-source projects and provide support and resources for others.

JavaScript

JavaScript is a popular programming language used for creating dynamic and interactive websites and web applications. It is a client-side language, meaning that it runs directly in the web browser of the user, and is responsible for adding interactivity to web pages.

One of the key features of JavaScript is its ability to manipulate the Document Object Model (DOM), which is the programming interface used to interact with HTML and XML documents. With JavaScript, developers can add or remove elements from a web page, change the style of elements, and respond to user interactions like clicks and scrolls.

JavaScript is also used for creating more complex web applications, often with the help of frameworks like React, Angular, or Vue. These frameworks provide additional tools and libraries that simplify the development process and enable developers to create more sophisticated and scalable applications.

Visual Studio Code

Visual Studio Code is a free and open-source code editor developed by Microsoft. It is available for Windows, macOS, and Linux operating systems.

Visual Studio Code provides a lightweight and extensible platform for developing various programming languages such as Python, Java, JavaScript, C#, C++, and many others. It includes a rich set of features like syntax highlighting, code completion, debugging, version control, and more.

Visual Studio Code also has a large and active community that creates and maintains extensions to add functionality to the editor, making it a highly customizable and flexible development environment.

And finally, GitHub, which was required in both, videogame and landing page:

GitHub

GitHub is a popular web-based platform that provides a range of tools and services for developers to collaborate on software projects. It is built on top of Git, a popular distributed version control system, and provides features like code hosting, project management, and collaboration tools.

GitHub also includes a range of social features, such as the ability to follow other developers and projects, star and fork repositories, and contribute to open-source projects. This has led to a large and active community of developers on the platform, who share code, contribute to open-source projects, and provide support and resources for others.

Technical requirements to run the videogame

MINIMUM:

OS: Windows 10, 11

Processor: 2.0 Ghz

Memory: 2 GB RAM

Graphics: 2Gb Video Memory

Storage: 300 MB available space

These are the requirements to run the videogame. Overall, there are a low technical requirements but the aspect that can affect more is the video memory.

Another aspect to have in consideration is that the game does not have graphic settings, so the user needs to accomplish the requirements if they want to play.

Common issues

1. Making a mechanic that can only be used a limited number of times:
   One of the most common mistakes in game development is creating mechanics that can only be used one or two times. This can make the game feel repetitive, and players may lose interest quickly. Instead, we create a mechanic that can be used in multiple ways and in different situations, that is the wizard combination. This will keep the game fresh and engaging for the player.
2. Achieving level design that is both enjoyable and simple for the player, but not boring:
   Level design is crucial in video game development, and finding the right balance between fun and simplicity can be a challenge. One common mistake is creating levels that are too easy or too difficult, which can make the game frustrating or dull. To avoid this, we create different abilities and enemies with varying difficulty in the same level.
3. Following an artistic design pattern:
   Artistic design is another important aspect of video game development, and it is essential to have a cohesive and consistent look throughout the game. One common mistake is not following a design pattern, resulting in a disjointed look and feel. To avoid this, we establish a design pattern early on in the development process and ensure that all assets, characters, and environments follow the same theme and style. The style is voxels and for the UI we used pixel art.
4. Ensuring that all aspects of the game are consistent:
   To create a cohesive and enjoyable game, all aspects of the game need to be consistent with each other. This includes art, music, mechanics, sounds, animations, and effects. One common mistake is creating elements that don't match the overall feel of the game. In this aspect, almost all the art was made by us except VFX, music and sounds, and with them, we pick the styles that match with what we want in the videogame.
5. Scope of the video game:
   The scope of a video game refers to the extent of its features, gameplay mechanics, and content. A common mistake in game development is having a scope that is too large or too small. A game with a scope that is too large may take too long to develop or exceed the budget, resulting in a rushed or incomplete product. Conversely, a game with a scope that is too small may not provide enough content or replay value, leading to players

quickly losing interest. It's important to define the scope of the game early on in development and make sure it aligns with the resources and time available. In this aspect, Sentinel could have dozens of levels with different enemies and abilities to make the player engage in it.

6. What the video game wants to transmit to the player:

   A video game can convey a wide range of messages and emotions to the player. Some games aim to entertain, while others seek to educate, challenge, or inspire. A common mistake in game development is not having a clear understanding of what the game wants to communicate to the player. This can result in a game that lacks direction or a consistent message. To avoid this, we define the core message or theme of the game and ensure that all aspects of the game, from the mechanics to the visuals and audio, support and reinforce this message.

7. The feeling of the gameplay:

   The feeling of the gameplay refers to the emotions and sensations that the player experiences while playing the game. A game with a great feeling of gameplay can keep players engaged and invested in the experience. However, a common mistake in game development is creating gameplay that feels flat, repetitive, or uninteresting. To create a strong feeling of gameplay, we focus on the mechanics and ensure that they are intuitive, responsive, and engaging.

Conclusion

The development of a video game is a complex and challenging process that requires a high level of skill, dedication, and collaboration from the team of developers. During the development process, the team had a series of difficulties and obstacles, including technical challenges, creative decisions, and time constraints.

One of the key challenges in game development is ensuring that the game is both engaging and fun to play. This requires a combination of technical skills, such as game mechanics and user interface design, as well as creative skills, such as storytelling and character development. Balancing these different elements can be difficult, and often requires extensive playtesting and iteration to get right. In Sentinel, we spend a lot of time to playtest de UI and the feeling of the game with different questions: Is this good enough? Is it fun to play? This art match with this mechanism? What other variations can we put in this aspect of the videogame?

Another major challenge in game development is managing the scope of the project. Video games can be incredibly complex, and it's easy for the team to get bogged down in details and lose sight of the bigger picture. It's important to set clear goals and milestones, and to be willing to make tough decisions about what features and elements are essential to the game and what can be cut. This was an important aspect for us, as we narrowed the scope over the first month until we achieved what we did with what is Sentinel right now.

Finally, game development is a highly collaborative process, and requires effective communication and teamwork from the entire development team. This includes designers, artists, programmers, and producers, as well as external partners such as publishers and distributors (that in this case they do not apply right now). Ensuring that everyone is on the same page and working towards the same goals can be a significant challenge, but is essential for creating a successful game.

Despite these difficulties, the process of developing a video game can be incredibly rewarding, both creatively and financially. By staying focused, working together, and being willing to adapt and iterate throughout the process, game developers can create truly engaging and memorable gaming experiences that resonate with players around the world.

Individual conclusions:

Paulina Torres: This project was challenging for the team, it requires a lot of creativity, during the design and while the stage of learning was going on is important to keep an open mind, the design of possible bodies, characters, or just random objects. Also, the communication and being open of new ideas from other teammates is essential, accepting corrections and learn from the team is always an option. Being perseverant also was an important thing during the development of the videogame, because not everything comes right at the first time the team wanted to achieve something, or sometimes the ideas were not exactly the same after actually developing the idea.

Brayan Prieto: Within software development, the branch of video game development is one of the most complex, including aspects such as modeling, animation, sound and narrative, creating a video game involves a more complex process, because unlike other developments, the video game being an experience requires special detail in aspects that other types of applications is not so important, and one of these can make the difference between a good and bad game.

As a personal experience, video game development can become overwhelming if it is not planned correctly, and even so, it is a development so changeable that without a good communication a too big project is unsustainable. Besides, being the processes as a chain effect, a bad planning can waste weeks in nothing.

Fernando Tarango: The development of a video game is something extensive and one of the most difficult areas of software, as it involves many areas of knowledge for a single goal. Leadership, UX and UI design, level design, game design, art, music, sounds, effects, animations, modeling, finance, QA, programmer... It's not something that can be achieved without effort and learning from diverse areas in small teams. Sentinel is a project with a medium scope despite the time we had, and we managed to get a good Minimum Valuable Product. Sentinel could grow more and be more fun, with more variables of magic skills, enemies, bosses and many other things. It was a very difficult project to do, which took many hours but from which I learned a lot and I will be able to export that knowledge to other projects and other areas.