1/16/2021

Web Games Development

Game & Website Design Document

Daniel N MacDonald B00394309
Linus Herterich B00400078

Contents

Website

Site Overview

The website needs to at least have four pages those being:

Home/Index page – the main page that people will first see, giving them an introduction to the game we will be developing by displaying a banner, screenshots and a brief description of the game.

Game page – A page where the game will be but will temporarily have placeholder but will also display the controls for playing the game.

Game Design page – this page will display the content that we will be using for our game and will be shown using HTML and CSS

Development Team page – This page will display about ourselves (game developers) with possible links to any other work that we have sent online.

Site map

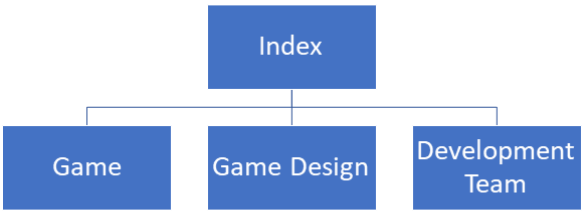


Figure 1- Site Map

Requirements

- The website has to be clean and easy to navigate
- Game design page can't just have a link to a PDF showing our concept
- Minimum of 4 pages
- A promotional video for the game must be somewhere on the website

We are thinking about having the website support mobile devices although the game itself will not support these

Design

In the web design, care was taken to use a uniform design language. This was characterised by uniform colours, elements and fonts that are repeated again and again in the web design. This creates trust in the viewer and makes him want to try the game much more when he is greeted by a neat and consistent website.

To make the website look less cluttered, a lot of whitespace was used between elements. The sans-serif font also looks clear and modern and also has generous kerning, which looks a bit playful but still neat.

Playful effects (e.g. when the elements appear) were used so that the website does not just look static, but invites you to play. You can interact with some of the web elements and play, because the bottom line is to play. Almost all elements appear gradually and you can interact with some of them by moving the courser over the elements.

Purple was chosen as the primary colour, as this is also the leading colour in our dungeons. Red was chosen as the call-to-action colour and white was chosen as the second call-to-action colour. Both colours stand out well and make interaction clear.

In the navigation bar, we have included a call-to-action button that redirects to the game page. As the menu is also visible when scrolling down, it is possible to start the game at any time (regardless of the page position).

Rounded corners are included throughout the design. This is UI-overlapping in the game as well as on the website. This also looks a bit more playful and not so serious or static.

The website is optimised for smaller screens, so for example the menu appearance changes with a small display size. Nevertheless, no great effort was made to achieve a perfect smartphone layout, as the game can only be played on devices with a native mouse and keyboard. The website can be called up on a smartphone, but a few more hours would have been necessary to optimise the layout even further for smartphones.

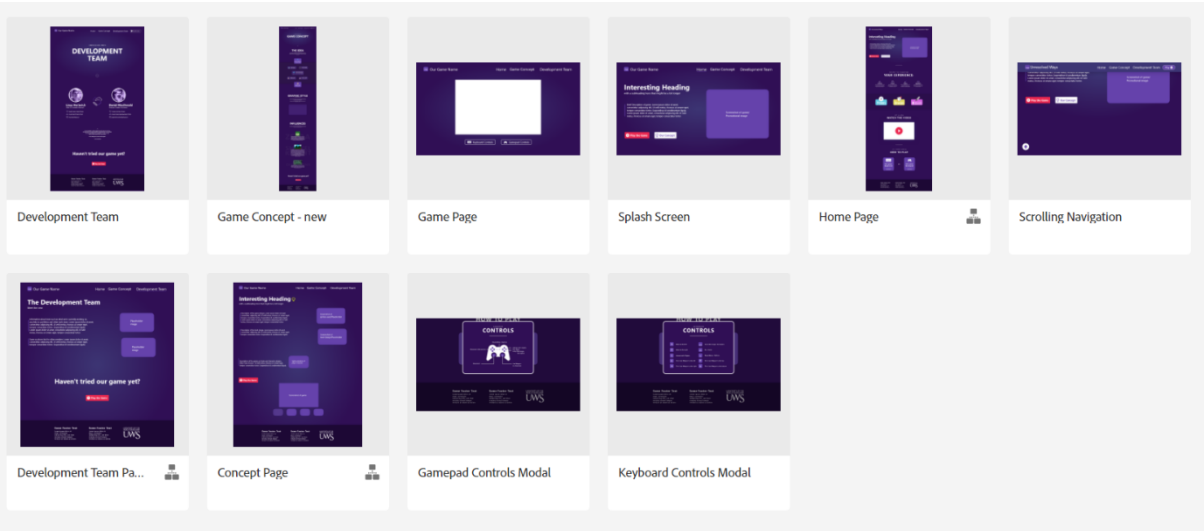


Figure 2 - Website concept using Adobe XD

Link to these concepts:

https://xd.adobe.com/view/12679156-e7f5-4c18-a2c9-9fd06fe173f9-4734/grid/

Game Design

Game name – Unresolved Ways

Overview

The type of game we're developing is a 2d sidescroller that will have some puzzle elements and enemies. It will have a pixel-based style with the player having a sword to fight the enemies and a limited amount of hit points. It should be possible to collect items / coins during the level.
To finish each level, a key must be collected in the level, which unlocks the exit. After each level there is the possibility to equip the collected coins / items to have better skills to create more difficult levels. This equipment is also possible in the main menu.
If the player loses all of them, he will be thrown out of the level and will lose all coins and items he has collected during the level.

Target Audience

The game is made for those who enjoy platformers and/or like puzzle elements. The game will have simple controls which can allow people who haven't much experience in this genre have an easy time getting used to the controls. With the style of the game, it can also be for those who enjoy the pixel art style.

Using Phaser and Phaser Editor

The game will be created using phaser which uses HTML5. Phaser is used to create games using HTML5 and JavaScript and is mainly used to put games onto websites and/or mobile devices. Phaser makes it easier to load in assets.

Phaser Editor is a level designer that is similar to the likes of unity, showing what the level and other screens will look like when it's running, can be edited easily and can also manage the assets you have. It supports applications like Microsoft Visual Studio Code which is what we'll be using when developing the game.

Requirements

Functional Requirements

- Playable Level
- Damage values for both enemies and player
- Health values

Non-Functional Requirements

- Main Menu
- Player and Enemy animations
- UI elements
- Quit game option
- Player and Enemy sounds
- Music for menu and levels

Necessary

- Work on our website
- At least 1 playable level
- Player and enemies have working animations
- Sound effects work as intended
- UI elements

Optional

- Quit game button
- Pause game button
- Controller support
- Purchase system to buy better equipment / cosmetics for the player with in-game currency

Game Interface Design

These representations were our first thought, but we made changes later. We noticed in the level selection that it would be better to start the levels directly after the tutorial in the welcome level.

Moreover, these sketches were created before the design of the website. After the website was finished, we tried to transfer the UI concepts (especially the call-to-action buttons) to the game. Thus, the buttons are finally rounded and the colours purple, red and white can be found more often.

Title screen

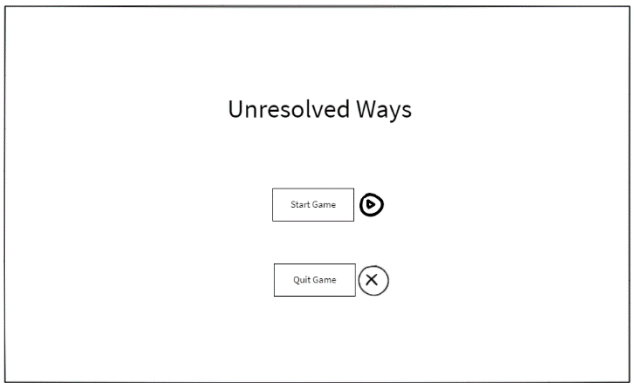


Figure 3 - Title Screen with start and quit game buttons

Level Select

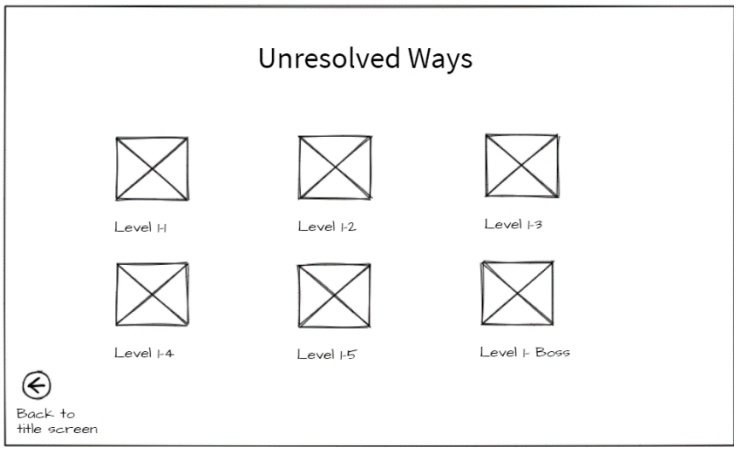


Figure 4 - Level Select

With the level select, the player will be able to choose which level to go to only once they have beaten it. When they first enter the level select, they will only be able to select the first level.

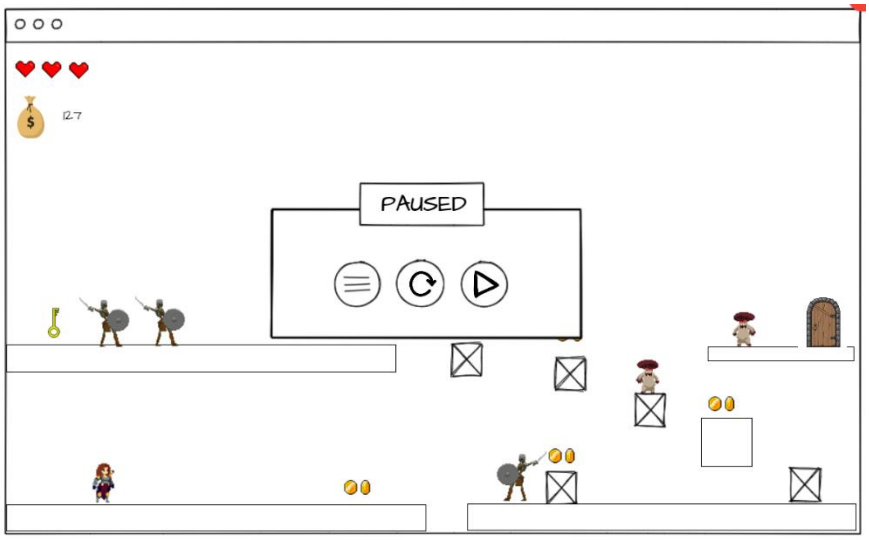


Figure 5 - Pause Menu

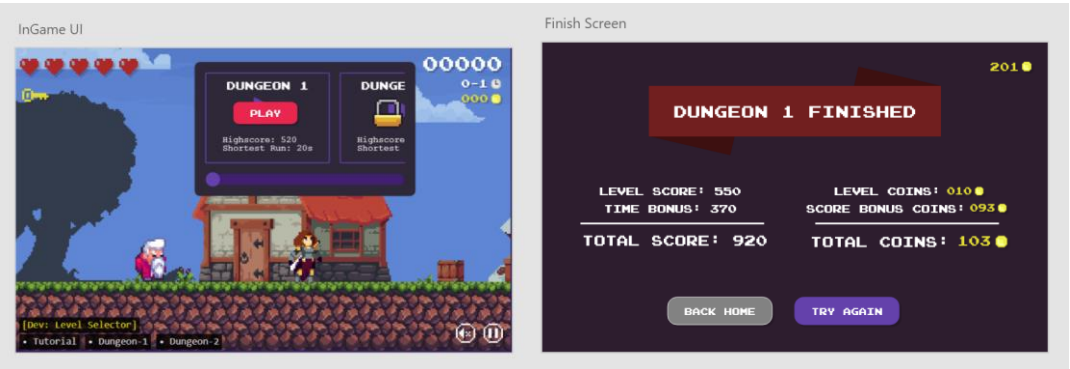
Concept for the pause menu which has return to main menu, restart level and resume buttons.

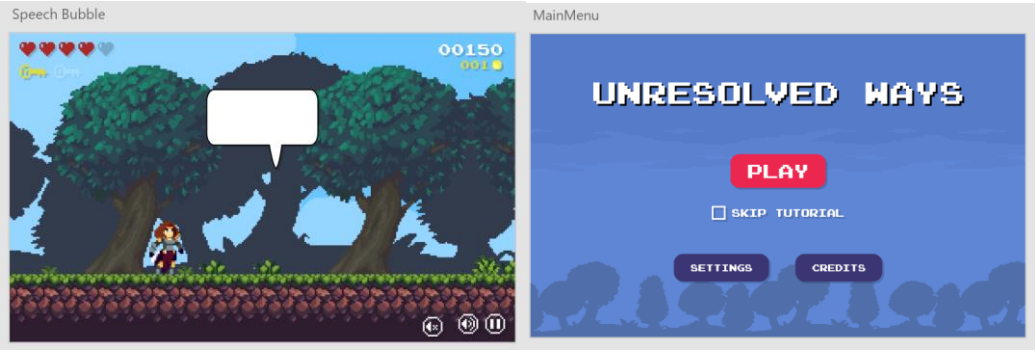
In game UI

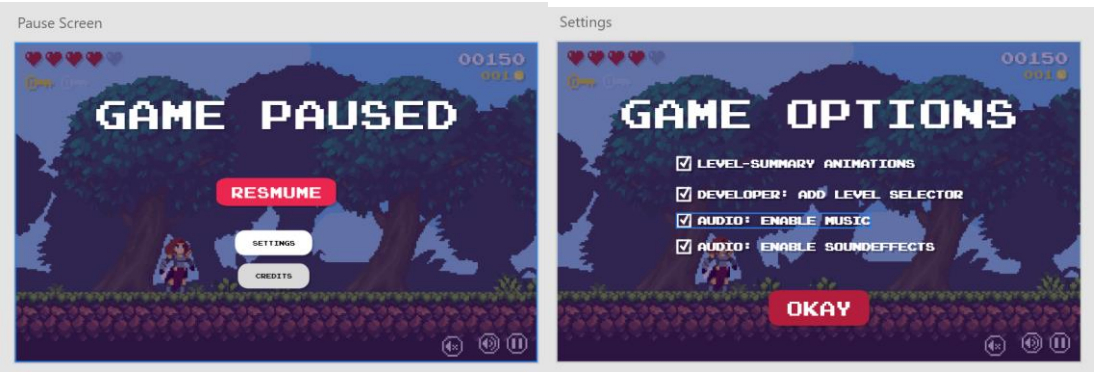
The UI will display the player currency and hit points, keeping track of how much money hit points the player has. They will be on the top left side of the screen as it will not obscure anything for the player.

Updated UI Sketches (Adobe XD)

→ see: https://xd.adobe.com/view/12679156-e7f5-4c18-a2c9-9fd06fe173f9-4734/grid







Game Environment/ Level Design

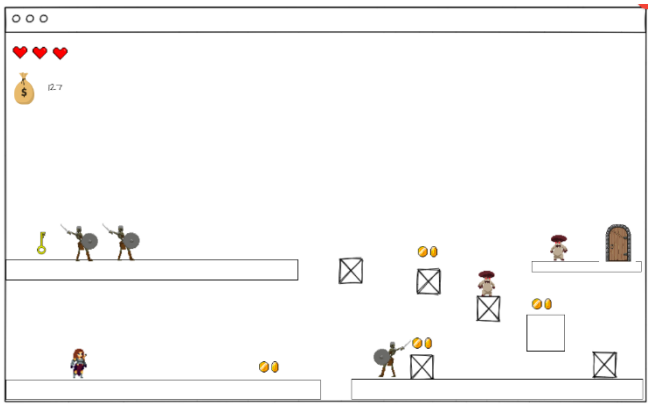
The theme of the level is of a dungeon with platform elements and puzzle solving. The player will need to overcome many obstacles using precise platform jumping while needing to figure out how to get to the next area. The first level will be more of a tutorial taking place near a village, where the player will get to know the controls and the objective of the game. The next level will be the player entering the dungeon. While the village was bright, colourful and peaceful looking, the dungeon will be darker and menacing.

As the player progresses through the levels, they will become more challenging with more obstacles that the player will need to avoid with precise jumping.

In the Village and the first dungeon level, care was taken to make the background move differently relative to the camera with a parallax effect. This created a more immersive world, as this is also the case in the real world. This was no longer possible in the pure dungeons (from level 2), as there was no background further back.

Care was taken in the level design that the keys were placed in such a way that they could be reached again if you accidentally walked past them.

The jumps that the levels require are often a combination of activating the sprint key and the correct length of pressing the jump key. This often makes the levels not so easy and it takes a bit of time to figure it out. However, this also gives the player a progression in the game, as it is a mechanic that can be learned well.



Early game concept

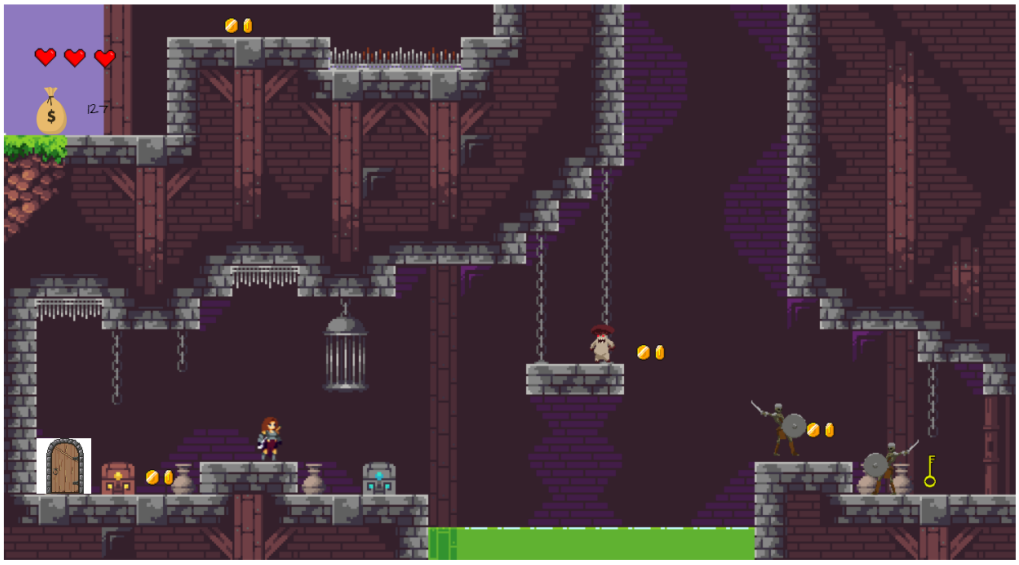


Figure 6 - Early concept using sprites, may change during development

Plot

Story

The story of the game is a young adventurer who is wanting to find the world's greatest treasure to make a name for him/herself. Once they enter this dungeon, they will need to face off against some strong creatures and solve puzzles.

Protagonist

The protagonist is a Thrill seeker who loves adventure and is excited to enter the dungeon. Equipped with a sword and shield, they are ready for combat but are also agile to make it through large gaps and higher places.

Enemies

The enemies in this game are some wild and forgotten creatures in the world with some trying to prevent the treasure from being taken. There'll be enemies that'll attack you with weapons and others throwing projectiles at you.

Player Interaction

Player progression – at the end of the level the player will continue to the next area, if they have left any items behind, they cannot go back and get it.

Items – the player can collect items such as gold, items to recover health and possibly new weapons to pick up and use.

Keys- the player will need to find the key to unlock the door

Enemies – when the player encounters an enemy, if they touch it without attacking they will lose a hit point but if they hit with their weapon then the enemy will damage the enemy.

Mechanics

Life system – the player will be able to take a few hits from enemies which will always be displayed for the player, when the player loses all their hit points its game over.

Combat – the player will be able to attack enemies with their sword, for most enemies 1 hit will kill them, but others may need more (not implemented in the end)

Jump & Run – the player needs to combine running speed and jump height to cross certain obstacles and find the way threw the dungeons

Exploration - the maps will have key(s) that the player will need to find to open the door at the end

Implemented Features

We have managed to program an executable 2D dungeon jump & run, which contains 4 levels of varying difficulty, as well as a tutorial level.

The tutorial guides the player through the start of the game with a few speech bubbles and explains the controls and the key mechanics. The tutorial is designed in such a way that after a few speech bubbles the player is asked to practise what he has just learned. Thus, after being prompted, the player must first move, then jump and finally find the key and unlock the first dungeon.

As soon as you have completed the tutorial tasks, the level overview in the main level is unlocked. In this overview, you can see the high scores and the shortest run. However, only the levels that you have unlocked are visible.

In the levels themselves, you are required to find a way through corridors and obstacles (spikes) to the door. You should also find a key along the way. You can also collect coins, which are then available globally. You also receive additional coins after completing the level, depending on the score you achieve. The score in turn depends on the time you have taken. The shorter the time, the higher the score. In addition, the score increases in the level per coin and key.

All levels have background music and sounds for collecting coins, the key, opening the door and taking damage. Damage is taken when you run into spikes. There is a damage cooldown of 2 seconds per damage. When all 5 hearts are used up, a death screen is displayed. There you can restart the level or go back to the main level.

In the game there is a start menu and a pause menu. The latter can be opened at the bottom right next to the option to mute the sound. In both menus you can make settings for the sound (music and effects), animations (whether the summary should be animated after a level or appear immediately) and the dev level selector. The dev level selector makes it possible to go straight to the next level without completing one.

All game credits can also be displayed in the menu.

Controls

Keyboard

Movement – Arrow keys or A and D

Jump – W or Space

Run – Shift

Pause – Escape

Interact with Speech Bubbles – E

Controller (Xbox controller) – not implemented

Movement and menu navigation – left analogue stick

Jump – A

Run – B

Pause – Start button

Select (Menu) – A

Interact with Speech Bubbles – A

Influences

The Legend of Zelda A Link to the Past

Figure 7 - In game screenshot, owned by Nintendo

The Legend of Zelda A Link to the Past originally released on the Super Nintendo Entertainment System but has been ported to many Nintendo consoles since then. It is highly praised by many for its gameplay being responsive, easy to learn and fun to play. It is also praised for its art style having everything in detail to what was limited at the time and its music. The game has a large world for the player to explore with dungeons in certain locations. This game has an influence due to it having excellent gameplay and level design.

Shovel Knight: Treasure Trove

Figure 8 - In game screenshot, owned by Yacht Club Games

Shovel Knight: Treasure Trove is an 8-bit style 2d sidescroller, while it does have a few expansions the one that has inspired the game that we are developing is Shovel Knight: Shovel of Hope. It has levels where you get though using items and upgrades you collect throughout the game and encounter a boss at the end of nearly every level. It has amazing sprites, music and level design which has helped it win various awards, being released on many devices and is highly regarded as one of the best indie games of all time.

One of the main objectives in the game is to collect money, you can do this by many ways digging, defeating enemies and bosses, looting chests and completing objectives. With the money you can use it to buy magic items or upgrade your health, magic, armour and shovel making you more powerful.

Super Dangerous Dungeons

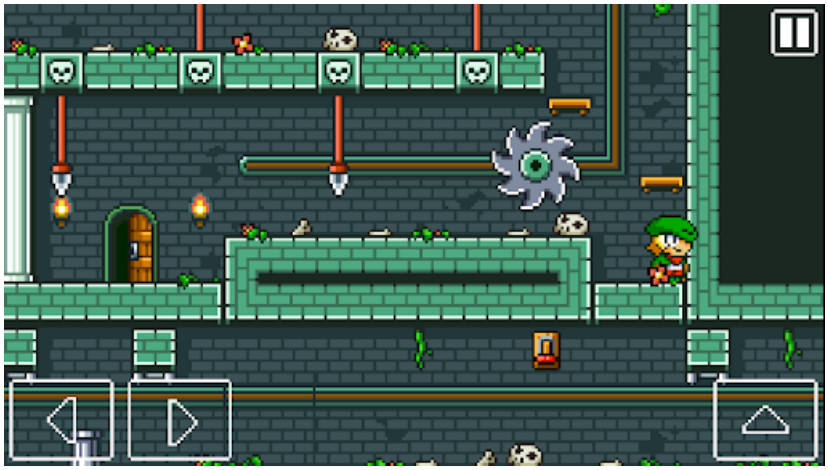


Figure 9 - In game screenshot using mobile device

Super Dangerous Dungeons is a platformer for PC and mobile devices that has simple controls as you can only move and jump. The objective is to go through the dungeon by avoiding enemies, spikes and other hazards. While the game is short it has good level design that has some puzzle elements to it.

Software/Resources Used

- Phaser 3
- Phaser Editor
- Tiled
- Microsoft Visual Studio Code
- Git Bash
- GitHub
- Piskelapp.com
- Leshylabs Tilesheet Editor
- itch.io
- Slack.com
- OneDrive
- Google Drive
- Todoist
- Audacity
- YouTube
- Adobe Photoshop
- Adobe XD
- Google Chrome
- Firefox
- OBS Studio and Lightwork for edited in-game footage

Testing

We will both be testing the game making sure that everything we add works as intended once we implement it to make sure there aren't any errors later. We will be testing more the menu navigation, player inputs and player interaction with enemies and items. Once the game is ready, it will be tested on different web browsers to make sure it is supported on those platforms e.g., Chrome and Firefox.

Project Management

Software Development Lifecycle

With the way that we're approaching the development such as testing what we implement right away, it is similar to the waterfall lifecycle as we try to finish the one task before going to the next.

Development Planning

We listed up all our features, bugs and project-todos in a software called todoist. There we created a Kanban board which you can see below:

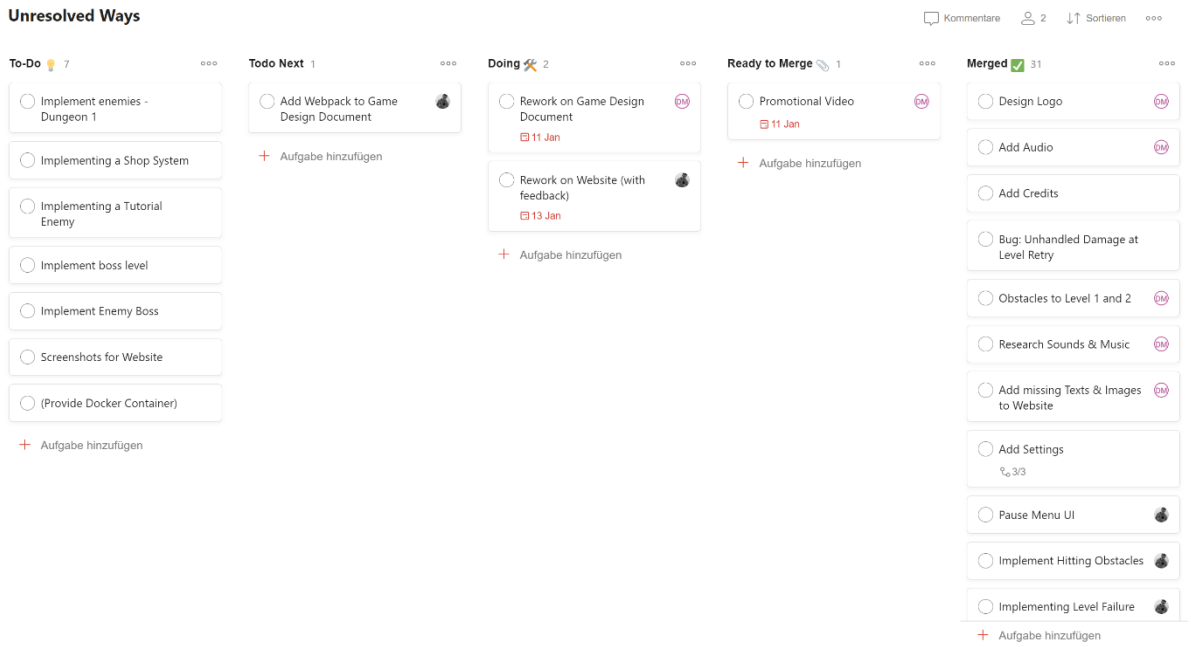


Figure 10 Screenshot of our todois project

GitHub

We used Git as our version control tool and GitHub as the host for our git repository. Feature-based branches were created before one of us started to implement a new independent feature. These were regularly merged to our master-branch and deleted afterwards.

Communications/project discussions

We primarily use Slack for communicating and have channels for each topic for the game to keep all messages and images tidy which you can see below:

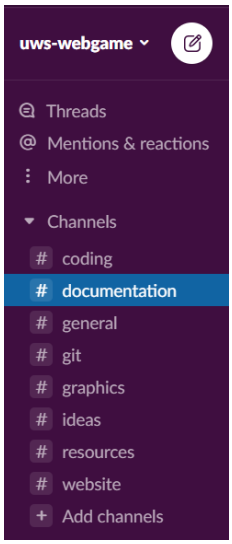


Figure 11 - Our Slack channels

We also use Microsoft teams on a weekly basis for discussions but also to screen share any information.

Changes to project through development

We first thought on creating a dungeon crawler but decided to instead develop a 2d sidescroller as it would be better for setting up obstacles for the player to overcome.

Due to time constraints, we are not able to add in enemy AI to the game, but we have changed some of the levels to be more challenging with obstacles.

We have just been using phaser and not the phaser editor as the both of us do not have much experience using phaser 3 and didn't want to spend time trying to learn phaser editor if phaser could work just as well.

We changed the original UI idea and built the level selection directly into the tutorial level, as we found it more intuitive and immersive.

We also slightly adapted the in-game UI by adding a score and a timer that counts up. This way you can try to beat yourself.

We also didn't think about the success and death screen at first, but that came to us when we were further along in the development process.

Technology: Webpack

We worked with the technology Webpack. Webpack is a bundle manager that reads various files, compresses them, optimises them and makes them available packed in a folder. It makes it easy to split up the code into logic snippets in many files.

The technology was integrated into the project after a few weeks, when we realised that it would be too confusing to do the entire game development in a single JavaScript file. WebPack is an industry standard and very good at compressing JavaScript as efficiently as possible.

At the time of the WebPack integration our entire website was already finished, so we decided to use WebPack only for our game JavaScript and not for our website HTML, CSS and JavaScript. WebPack has its own peculiarities when it comes to importing resources and it would be very time-consuming to restructure the whole website. So we decided that WebPack should not touch the website files.

In concrete terms, WebPack takes the "src/website" folder and leaves all the files there as they are. The "src/game" folder, on the other hand, is optimised and packed by WebPack and either made available in RAM (with the command 'npm start') or delivered in the "build" folder (with the command 'npm run build'). The entry point for game optimisation is the game.js file in the "src/game" directory. There, all required modules (which are either designated files or previously installed npm modules) are loaded (with ‘require’ or ‘import’ commands at the top of each file) and called step by step.

Deployment

Since Linus already has a web server that he knows well, we decided to do the hosting there. All we had to do was upload the contents of the "build" folder created by WebPack to the web server via FTP. Now the game is available at https://unresolvedways.nfl-server.de/.

References

In-Game Assets

- Village Tileset: https://clembod.itch.io/country-village-free-pack
- Warrior Character: https://clembod.itch.io/warrior-free-animation-set
- Sensei Character: https://bakudas.itch.io/generic-rpg-pack
- Dungeon Tileset: https://bakudas.itch.io/generic-dungeon-pack
- Pixel Icons: https://thenounproject.com/icon/16863, https://thenounproject.com/icon/1537411, https://thenounproject.com/bluetip/collection/pixel-icon-circle
- Mountain: https://muratcanbaykus.itch.io/pixelmountain
- Music (1)
    - Author: ArcOfDream
    - Tracks: Invitation, Tonal Dissonance, Gone Fishing, Dream of Battle
    - https://soundcloud.com/garoslaw
    - https://arcofdream.itch.io/monolith-ost
- Music (2)
    - Author: Crezno
    - Track: Return to the Monolith
    - https://soundcloud.com/creznohip/return-to-monolith
- Soundeffect Coin: https://opengameart.org/content/10-8bit-coin-sounds

- Soundeffect Jump: https://opengameart.org/content/platformer-jumping-sounds
- Other Soundeffects:
    - Author: Lena Sturz, Daria Gadaleann

Useful Articles and Videos

- Creating a platformer using phaser: https://gamedevacademy.org/how-to-make-a-mario-style-platformer-with-phaser-3/
- RexPlugin Phaser 3: https://rexrainbow.github.io/phaser3-rex-notes/docs/site/
- Collecting objects: https://medium.com/@alizah.lalani/collecting-objects-in-phaser-3-platformer-games-using-tiled-4e9298cbfc85
- Learn Webpack: https://www.youtube.com/watch?v=MpGLUVbqoYQ&feature=youtu.be
- Phaser and Webpack: https://phasergames.com/set-up-phaser-3-npm/
- Audio Getting started with Phaser 3: https://www.youtube.com/watch?v=COncYQLGJS8
- Phaser Scenes: https://www.youtube.com/watch?v=gFXx7lgxK9A

JavaScript Libraries

- Phaser 3.24.1: https://phaser.io/phaser3
- Bootstrap 4.5.3: https://getbootstrap.com/
- JQuery 3.5.1: https://jquery.com/
- TiltJS: https://gijsroge.github.io/tilt.js/
- AOS: https://michalsnik.github.io/aos/

Influences

- The Legend of Zelda a Link to the Past image - https://www.theverge.com/2014/1/30/5361028/the-legend-of-zelda-a-link-to-the-past-wii-u-virtual-console
- Shovel Knight: Shovel of Hope image - https://yachtclubgames.com/shovel-knight-of-hope-press-kit/
- Super Dangerous Dungeons - https://store.steampowered.com/app/856540/Super_Dangerous_Dungeons/