Timmy’s Turbines

Game Technical Report

Authors:
Rick Baksteen
Raphael Klein
Kangqi Li
Panagiotis Papamanolis
Ruoqing Sun

Introduction

This report discusses the technical aspects of the wind turbine MOOC serious game Timmy’s Turbines. The decisions that were made are explained, the challenges that lie ahead discussed and some of the problems that were faced during development are looked at.

Technical choices

Engine

A requirement for the game is that it should run in a web browser, to maximize accessibility and multiplatform support. Taking information from the commissioners about the browsers that the target audience use:

- Chrome 50%
- Firefox 14%
- Safari 10%
- IE 5%
- Edge 1%
- Opera 1%

Approximately 85% of people in the target audience are using web browsers through desktops.

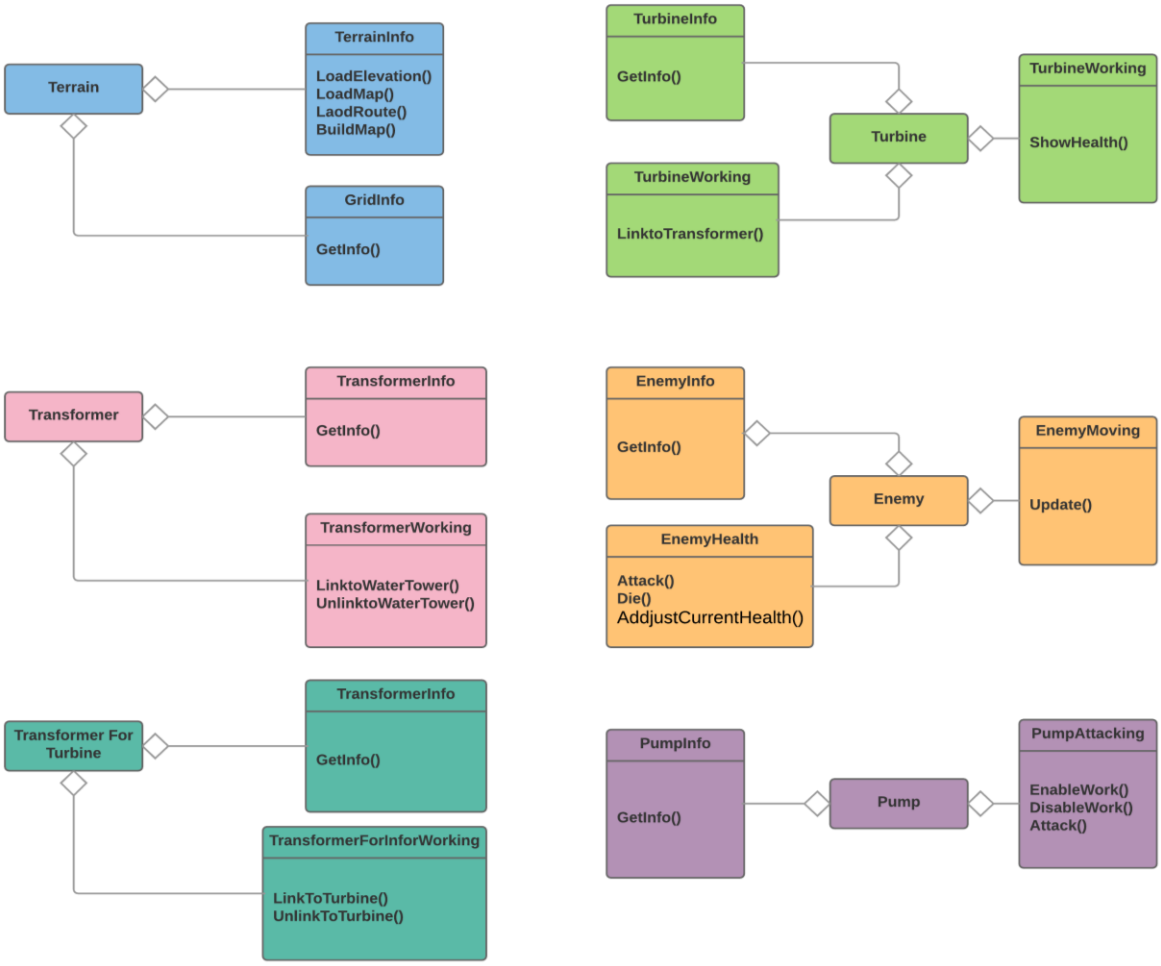
HTML5 has been selected as the base of the game instead of Flash, because HTML5 games can be played not only on PC, Mac and devices based on Android or iOS, but also on Firefox OS and all the others that support the HTML5 standard. This increases number of potential players, while also being more future-proof since Flash is uses less and less.

Considering team size, workload and the game scale, Unity was chosen as the game engine, which can handle the physics simulation, animation control, user interface creation and audio management in a single game system. For the prototype, Unity’s web player export was used, because it allows for a quick deployment of a new version of the game online. See the ‘*Unity Web Player*’ section in the Future challenges chapter for more information.

Libraries used

During the game development, standard unity environment asset is the only library applied in this game in order to achieve water effect in the flood. No other libraries are used.

Class Structure



GUI

The GUI has been split up in 2 main parts. The left part contains the game settings like pause, fast forward, audio controls, access to the tutorial and chapter selection, as well as some statistics of the current level. This has been placed here so it doesn't interfere with the gameplay. While playing the game, the player will mostly be active in the game screen and the right part of the GUI, which contains all the gameplay elements. The right part can be split down into 2 different elements. The top half always shows the amount of Turbine coins the player has, and gives access to the build menu which allows the player to select and build turbines at all times. The lower half is a dynamic GUI element that shows information of anything the player is currently interacting with.

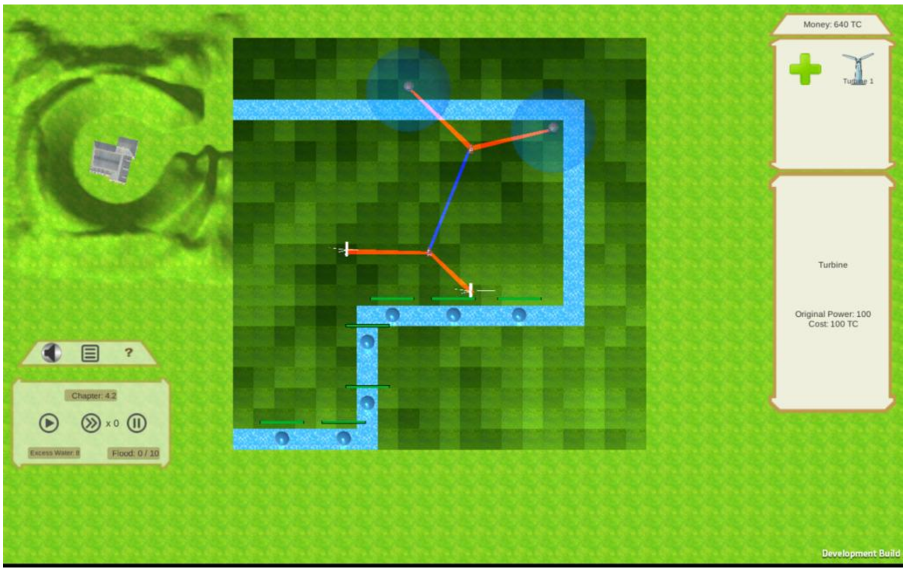


Figure 1 – Graphical User Interface.

The game screen has been kept void of unnecessary clutter, so where possible drawing text or numbers in the game screen is avoided To achieve this, graphical visualizations for many things like elevation, turbine status, health bars etc, pump targets, and power loss are used. Because there can be a lot going on during a game, some numbers have to be communicated to the player sometimes, which are shown and hidden automatically whenever they are relevant. All in an attempt to keep the game screen free of clutter, while still conveying all the required information.

Graphics and audio

Camera perspective

To complement the 2.5D graphics, an orthographic projection is used. This projection is parallel instead of having a viewing angle, which means there is no sense of distance. Because of that, every tile will be equally large, independent of the camera position. It gives the game screen a more uniform look, and allows players to see and interact with every part of it equally well.

3d Models

The 3d models for the turbine and the city with their respective textures and materials were free models, found on turbosquid.com. The 3d models pump and transformers were free models from Unity asset store.

Textures

Textures used for all the UI elements were designed specifically for this game. The textures used for the tiles were free textures found from http://www.myfreetextures.com/

Audio

The background music used in the game is royalty free from bensound.com and has been adapted slightly for the purpose of this game, so it loops cleanly. Most of the sound effects have been recorded or synthesized specifically for this game, with a couple of exceptions where samples from free online sound libraries have been edited to suit the needs of the game.

Player Interaction

Tiles

It has been decided to build the interaction structure on a tile-based system. That means that the entire game world is built up of square tiles, where each tile has specific properties. The obvious drawback of this is that the game world will look a bit artificial. The river for instance, is limited to 90° turns. A continuous terrain model would graphically look much more appealing and realistic.

However, it was quickly noticed that creating these models was relatively labor intensive. (see the '*Special Purpose Algorithms*' section for the current solution.) More importantly, a continuous terrain proved to be more difficult to interact with from a player perspective. Placing a turbine in 'the perfect spot' would become a pixel hunt which didn't fit the purpose of the game. (See the '*3D terrain*' section in the Recommendations chapter for more information and a proposed solution for future development.)

Camera control

Because the game is built in 3D, the player could theoretically be allowed to control the camera in 6 degrees of freedom. This has been restricted to 3 degrees of freedom (pan and zoom) to keep it simple. These options allow the player to customize the view to their preferences and screen size. The other options would not add any functional value and just be a gimmick.

Because the entire game is controlled just by mouse, camera control has been added to the mouse as well, using the scroll wheel and right mouse button as modifier.

Tutorial

To introduce the player to new concepts in the game, an interactive tutorial was designed. At the start of each level, the game is paused and the tutorial shows the player the location and functions of new UI elements, as well as explain new gameplay mechanics. Because the tutorial is an integral part of the game, players are always confronted with explanations of new mechanics and are more likely to actually read them. To keep the flow of the game intact for experienced players, the tutorial can be skipped at any time.

Special-purpose Algorithms

Map Generation

To keep the game fresh, a different map has been designed for each level in the game. To be able to quickly design new maps, a method to generate these maps with relatively little effort has been developed. Manually assigning elevation values to each of the 400 tiles in the map for instance, would be a lot of work.

The designed algorithm starts by setting 1 tile to elevation 100, the maximum value. From here on, it randomly assigns elevation values to surrounding tiles, with a threshold for the maximum allowed deviation from the neighboring tiles' elevation. This creates a naturally flowing landscape. The elevation value is then used to adjust the shader of each tile, so that the brightness is automatically adjusted. The only input required now, is to draw the river turning points on the map, and adding pre-placed elements like the pumps and transformers.

Future challenges

Since the game in its current form is only a prototype, there is still lots of work to be done to complete it. This chapter discusses some of the technical challenges that lie ahead, should the game be developed further.

Database for player accounts

If the game is developed further, it will become too big for players to complete in 1 go. They will have to be able to save their progress. Since the game runs in their browser, the only way of doing that is through an online database where their account information is stored. Not only game progress needs to be saved, but as turbine designs become more complex the player will want to save those too. Finally, the replay value of the game could be increased by adding online leader boards where players can compete for the high score in each level.
All of these will require a carefully set up database that can efficiently store and recall all of this information.

Unity Web Player

The Unity Web Player that is currently used has been discontinued by Unity because several major browsers have stopped supporting it. Currently, the web player only works in FireFox, Safari and Internet Explorer 10. For this reason, it is no longer a viable platform for this game. However, the latest Unity 5 can successfully export the game onto the web platform supported by HTML5 and WebGL. This is still in development, but Unity has announced that it is working with browser developers to offer web integration at a level that is as close to the webplayer as possible. The current version of the WebGL exporter already creates a fully functional game that runs in most modern browsers. There are some issues with performance on some systems still, but these should hopefully be resolved in the near future.

Level design for future features

Once the game is expanded with more features, balancing will become harder. More turbine designs, more events that influence the efficiency of the player and different enemy mechanics will require some thought on how to make the game still challenging without being too difficult.

GUI design and scaling

The build menu and design menu will have to be split up in multiple tabs in the future, to accommodate for all the extra options that the game will provide. Keeping everything clean, but also clear and accessible will be a challenge that requires some creativity.
Currently the UI is scripted to scale, depending on the screen resolution. Ideally, the entire game would scale depending on the resolution. That would make the game look good even at very low and very high resolutions. In the prototype, this requires the user to pan and zoom the camera to manually achieve this.

Animation

There are currently very few animations in the game, and they are very simple. To make the game more engaging, things like rotating turbines and turbine blades, a more convincing pumping animation and flooding animation would be great. That would require some more knowledge of animation and particle systems. This is an area of Unity that we have not touched yet.

Recommendations

This chapter will discuss some features or ideas that were planned, but didn't make it into the game as they were originally intended, because of varying technical difficulties.

3D terrain

As mentioned before, 3D terrains look amazing but compared to the 2D solution they are very labor intensive to create, especially when continuously rebalancing the game. Although some nice looking terrains were made for this game, the time investment for sculpting and texturing them for every level was not worth the effort, especially because they added nothing for gameplay and complicated the player interaction.

The main problem with player interaction is the fact that every location on the terrain is different, which means the player has to be very precise in his placement. This adds uncertainty about construction costs for instance.

Additionally, it is more complex to restrict the construction options when not using tiles. Limiting 1 turbine per tile is much easier than making sure a player can't build 2 turbines nearly on top of each other on a continuous surface.

A possible future solution would be a hybrid system, where a continuous terrain is used for visual appeal, with an underlying structure of tiles that (albeit invisible) takes care of the user interaction. This requires some careful thought on how to align these tiles with the terrain, but it would offer the best of both worlds.

Audio mixer

Unity has extensive built in audio functionality. However, most of its options are only available when audio clips are part of game objects. This is a problem in our game, where we are continuously creating and destroying game objects. Destroying a game object will also destroy it's audio source which means you can't play a sound when destroying something. The workaround we implemented involves playing audio clips directly to the audio listener from script, but this allows no audio routing. To make a fully customizable audio mixer for a player, we would have to design our own audio handling scripts which was too much effort for the scope of this project.

Game window scaling

As mentioned, ideally the game would scale entirely depending on the player's screen resolution. In the prototype, the UI can scale with resolution, but the game itself proved to be more difficult. To overcome this, a fixed resolution of 800x600 has been enforced to make sure the game looks good on every screen. This is the universal standard for web content. Although this works, it's not a great solution because the game will be small on high resolution screens which can take away from the immersion.