

# Game GUI and Design Report



## Design Aims

### **Introduction**

In our project design, we've aimed to create a clean, flat graphical user interface. This idea behind this is to eliminate any distractions and focus on clear, simple and immersive gameplay. In theory, clean design reduces the effort you need to put in to find information. In practice, we think we have achieved this well, as the following report will explain.

### **On Splashscreens**

The first step for this was to rule out the addition of a splash screen upon opening. Game splash screens typically only contain two options anyway- start game and tutorial/instructions. As we are shipping the game with a User Manual, we didn't see the need for on-screen instructions, therefore it seemed pointless to add a splash screen that just contained a 'Play' button. Thus, upon opening, our game lets the user get straight into the gameplay.

### **Use of Colour**

When designing our user interface, we wanted to use colour to make it attractive to the eye, but in order to keep it clean, and not garish, we kept the range of colours to a minimum. As such, the final screen, excluding trains, only uses four colours- blue, green and black for the Europe map backdrop and white for any stations, junctions, checkpoints or track.

### **Elimination**

We have attempted to eliminate any unnecessary assets that designers are often tempted to include into interfaces. This includes things such as borders around buttons, bullet points in lists and lines between items. Adding such items to interfaces often creates a design that feels cluttered, while often *not* adding to the readability of text and also generally *not* improving the item's ability to stand out. Therefore, we have made sure that our buttons are plain-text, with no borders, any lists- for example, our list of goals- do not feature bullet points, and we have used space separation rather than dividing lines to distinguish between different parts of our UI.

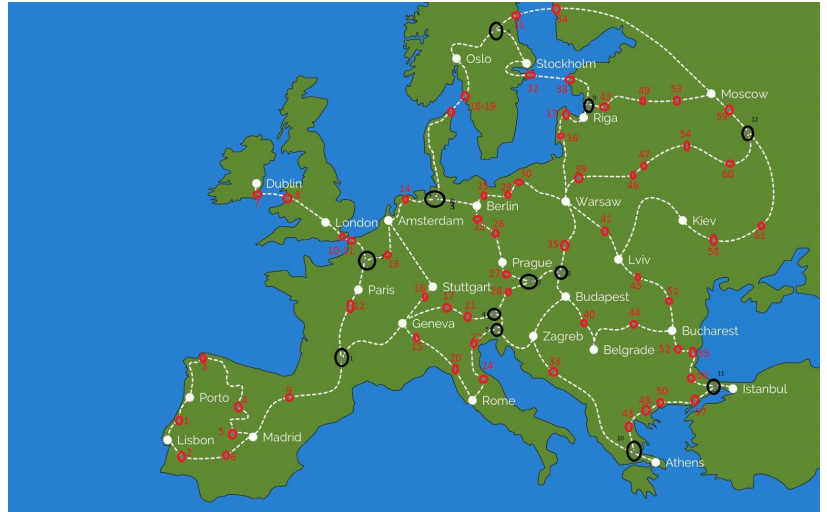
# Component Design

## Map

We decided that as there is a lot of information that the map needs to display to the player, we would create a very simplistic background image with a clear and instantly recognisable image of Europe. This 2-tone feature keeps the information clear and the user well orientated as to where they are working. In this Diagram (not for the User) the junctions are ringed in black, the cities are white dots and the checkpoints are ringed in red. When referring to these later in this document the GUI used in the game (lower image) is referenced.

The idea of a statically generated map is essential in our project as the complexity of the game is based on routing trains between stations, so a procedurally generated map would likely be unbalanced for such a problem and cause games to become un-enjoyable as a "bad" map would create trivial or even impossible problems.

The map remains visible at all times, we intentionally designed the map so that it can all be read and understood without needing to zoom in or out, therefore it never moves and all menus will lie translucently over the top. These menus will be toggleable to keep the screen clear if the player doesn't want the clutter.



The map is represented as a MapGraph object, which is displayed by the MapGUI object. The MapGraph object handles all the methods for retrieving, and changing all the information about the map state.

## Stations

We chose the stations as they are mostly capitals of their countries and are therefore recognisable, each station is represented by a *large* white circle to ensure that it is easily seen against the green background, it will turn dark blue if it is selected. The station's circle is the largest kind of circle on the map, this implies it's importance to the user- stations are clearly more important as they are necessary to complete goals. Some stations were added that are not capitals purely to ensure an even spread of cities over the map. Here is a list detailing the cities/stations used.

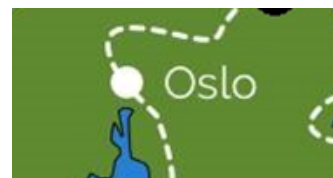
### Capitals:

- London
- Paris
- Berlin
- Madrid
- Moscow
- Oslo
- Warsaw
- Budapest
- Istanbul
- Stockholm
- Rome
- Prague
- Athens
- Lisbon
- Bucharest
- Kiev
- Amsterdam
- Geneva
- Belgrade
- Dublin
- Zagreb
- Riga

### Non-Capitals:

- Stuttgart
- Porto
- Lviv

Stations inherit all the properties of junctions, as they require all the same functionality, with the addition of a name, and the ability to be used in goals. The distinction is made because while they work in much the same way, the user sees stations as separate, more important entities.

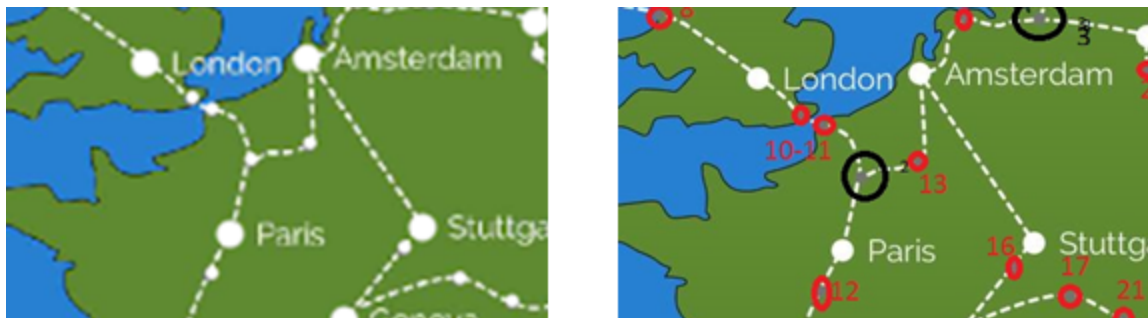


## Junctions

A junction is anywhere where multiple tracks join that isn't a station or a checkpoint. We made these larger than checkpoints to show that they are selectable and trains can be directed to them. They maintain the white circle graphic to keep continuity, but are slightly smaller. This subtle difference implies to the user that they are less important than stations.

The Junction class is used as the base for any point of interest on the map, it can be extended to station, checkpoint, or exist in its own right. Junctions have methods to check which trains are presently at them and which junctions are connected to them.

A choice is required at any junction or station



## Pathing (Checkpoints)

The small white dots represent checkpoints on the track, each checkpoint uses one of a train's movement for that turn and so it is imperative that the players can easily see where each track goes but also how long it is going to take to travel down any given one.

Checkpoint is an extension of junction, with additional functionality to allow for faults in the track. Checkpoints are assigned buttons in MapGUI, but they aren't valid selections for a move command as explained in the User manual, which means that they cannot be interacted with as the game ruleset does not allow for checkpoints to be set as destinations. (This would be an easy extension if required)

## Tracks

Tracks are represented as dashed lines connecting the junctions, checkpoints and stations, these lines are faithful to the white colour scheme in order to maintain consistency and make them easy to follow. The representation of junctions and stations mean that there is never any confusion about when one track crosses another as is clearly displayed. Trains will never appear on pieces of track that aren't some kind of junction, so the routes won't get obscured. They are displayed through the GUI purely to maintain the player's immersion, and add clarity to the connections between each junction.









## Trains

Each train is represented by a different coloured shield image. If a train has been selected by the player then the current selected train will be highlighted by a black line around the outside of the shield. The train level is represented by a number in the middle of the shield, for instance: a level 2 diesel train that the player has selected will be represented as:



At the end of a turn the location of each train will be updated if the player has made a successful move through adjacent junctions. The colour scheme for each train type is documented below.

-  - Electric (selected)
-  - Electric
-  - Diesel (selected)
-  - Diesel
-  - Flying (selected)
-  - Flying

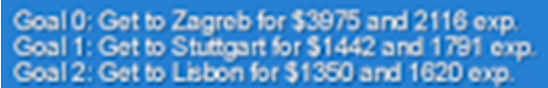


## Goals

Displayed in the top left of the screen, goals are contained within an otherwise unused section of the UI, as a result they can be constantly displayed without obfuscating the map. Each goal is displayed as a string for the player to see and achieve. Goals are replenished as they are completed in order to always have 3 goals active at any point in the game (one of our requirements), this means that the section can maintain a static shape and position. The shop window would be between the goal section and the player's info.

Goals are represented by Goal objects, which can be extended to specific goal types, enabling different kinds of goals to be developed quickly and easily.

The goals are managed by a GoalEngine object, which creates goals when they are needed, and destroys them when they have been completed.



Goal 0: Get to Zagreb for \$3975 and 2116 exp.  
Goal 1: Get to Stuttgart for \$1442 and 1791 exp.  
Goal 2: Get to Lisbon for \$1350 and 1620 exp.

## Player Info

This section sits at the bottom left corner of the window like the goals section, this is a static field filled with the score and wealth of the current player. Translucent, but not toggleable this window will be small enough not to be intrusive on the rest of the UI. In the player info section we have also placed the move-train, end turn, and quit buttons to keep actions and information in the same place and reduce screen clutter.



Player 1  
Wealth: 0  
Score: 0  
Move   End turn   Quit

## Summary

Simplicity is the main push of this UI design, with bright colours and only one screen we hope to have managed to keep the game simple to use and pretty to look at.