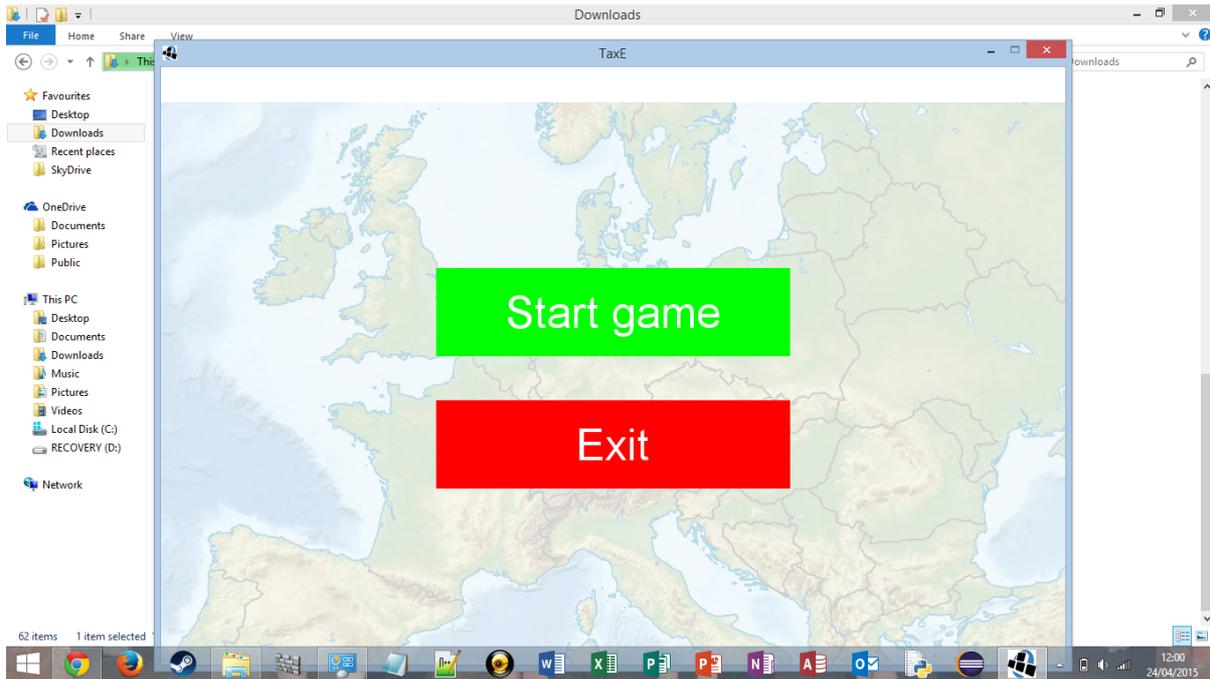


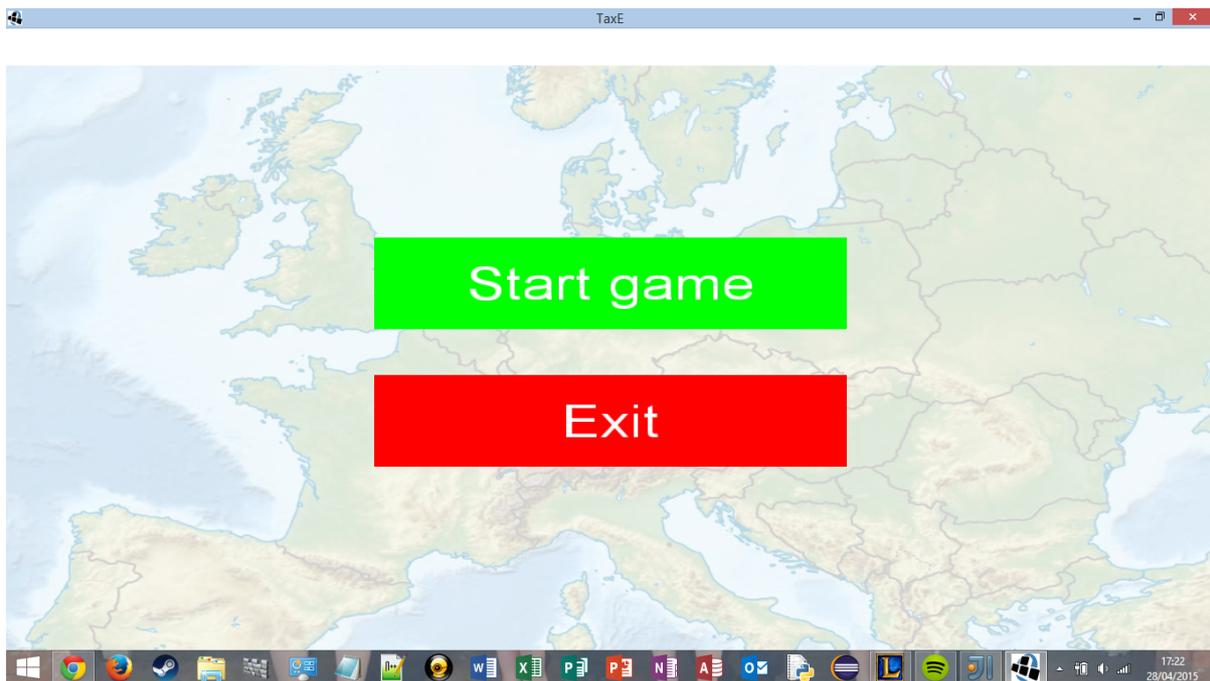
EEP's TaxE Software Engineering Project (Appendix M, Software Engineering Project (SEPR) Group Open Assessment)
Corrective Changes for Assessment 4

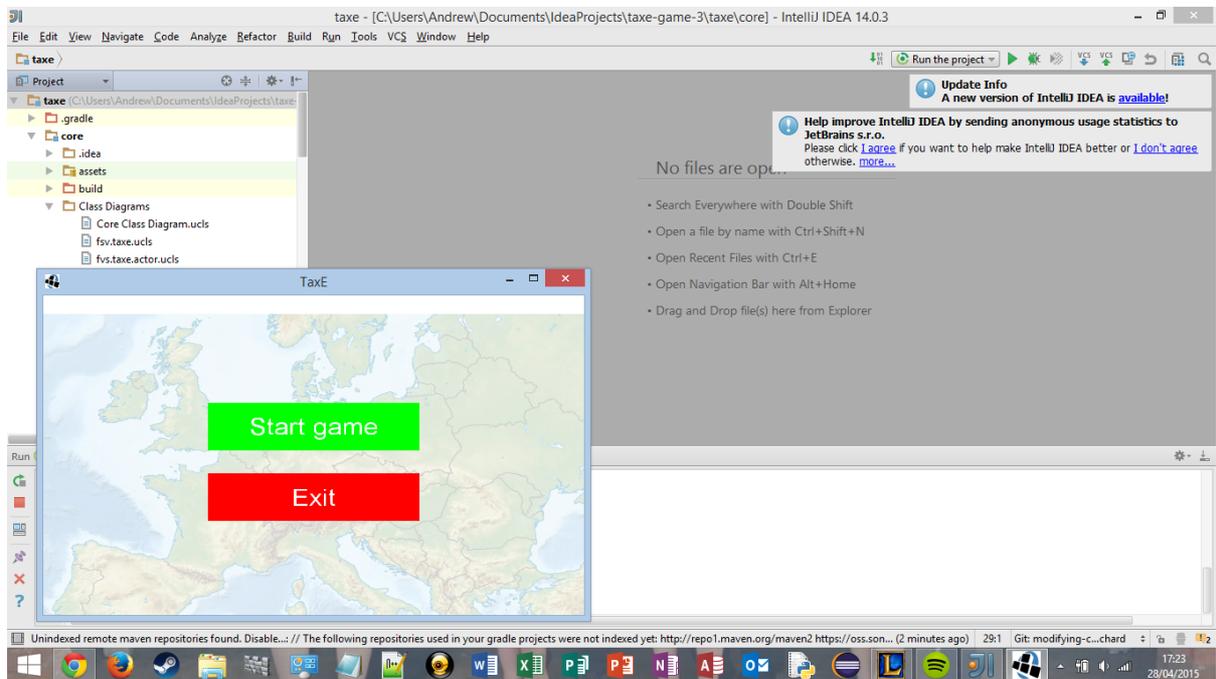
1. Resolution

On a standard resolution laptop screen (1366 x 768), the following screenshot would be displayed:



As shown, the screen is a fixed resolution of 1022 x 678 with the maximise window button disabled, restricting the users freedom to play the game in whichever resolution they wish, like in most modern games. The screenshot below shows the changes made using viewports about how it can be adjusted:

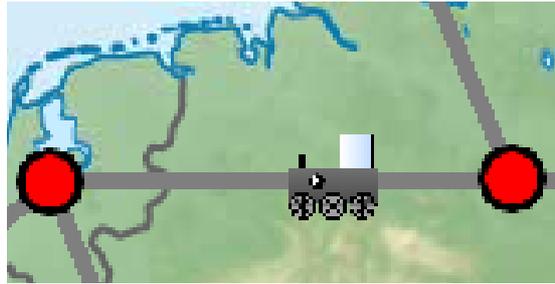




The two screenshots show the game in a maximum window and a customised one, indicating that the viewports correctly function which allow the user to alter the game to any resolution they wish.

2. Train Graphics

The original graphics for the trains were regarded by the clientele and our team as of a poor quality, hence the change was required. An original train image is shown below on the track (Hydrogen train):



Two examples are shown below for an electric and green train:

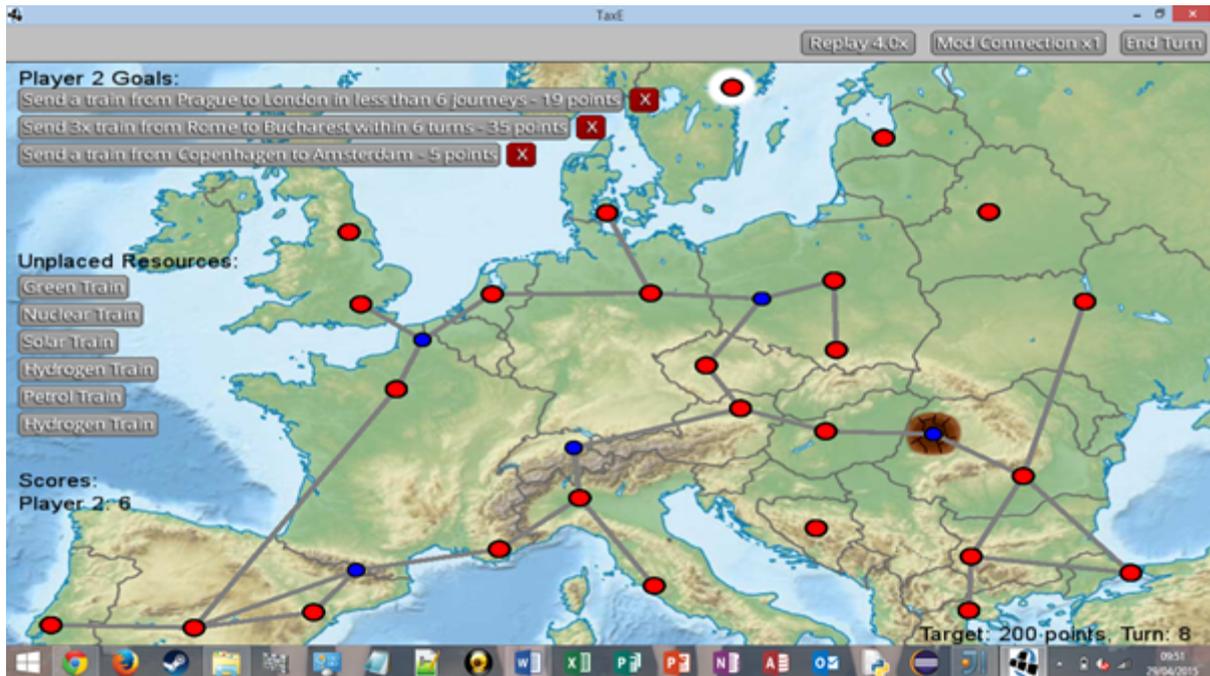


Train images were obtained from an open source website [3] and adapted to our needs for representing each train.

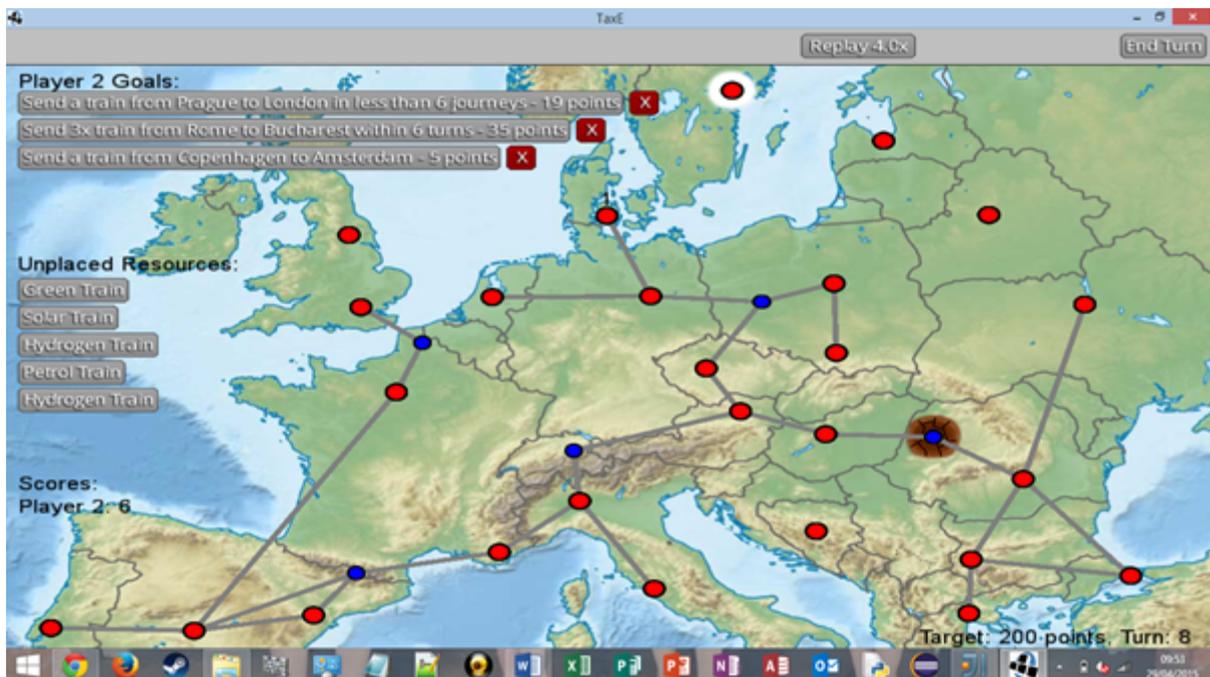
EEP's TaxE Software Engineering Project (Appendix N, Software Engineering Project (SEPR) Group Open Assessment)
Additive Changes for Assessment 4

1. Track Modification

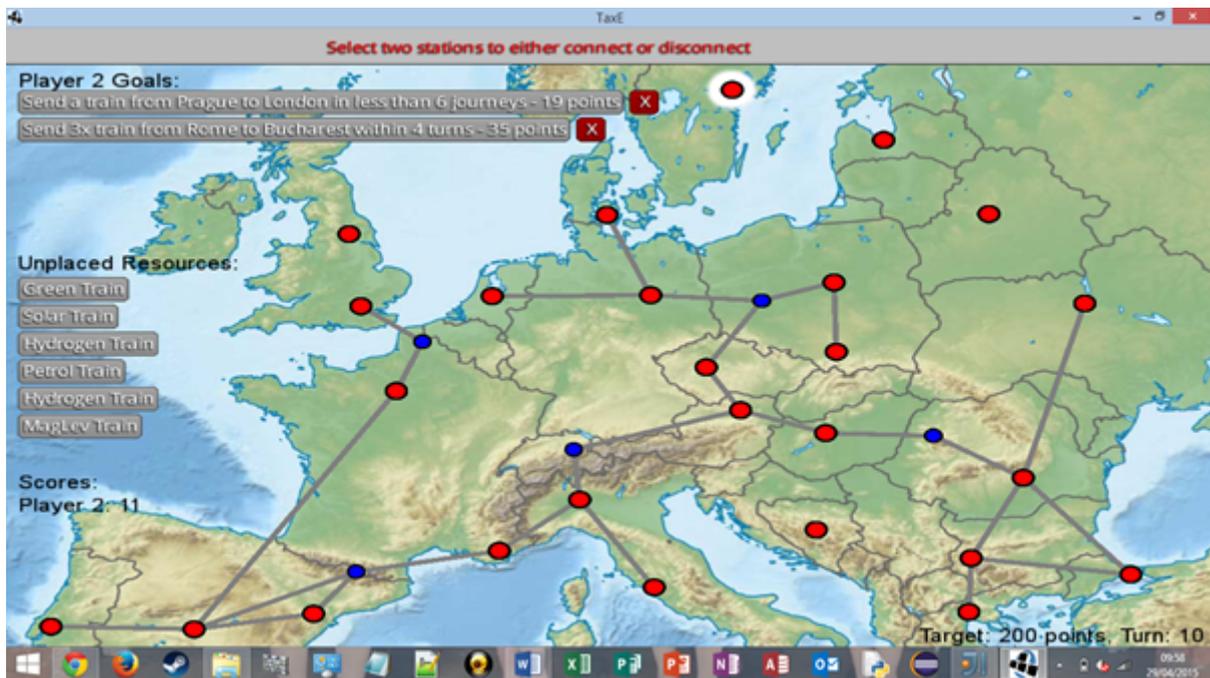
Screenshot showing a goal had been accomplished and the “modify connection” button is now visible in the top right hand corner::



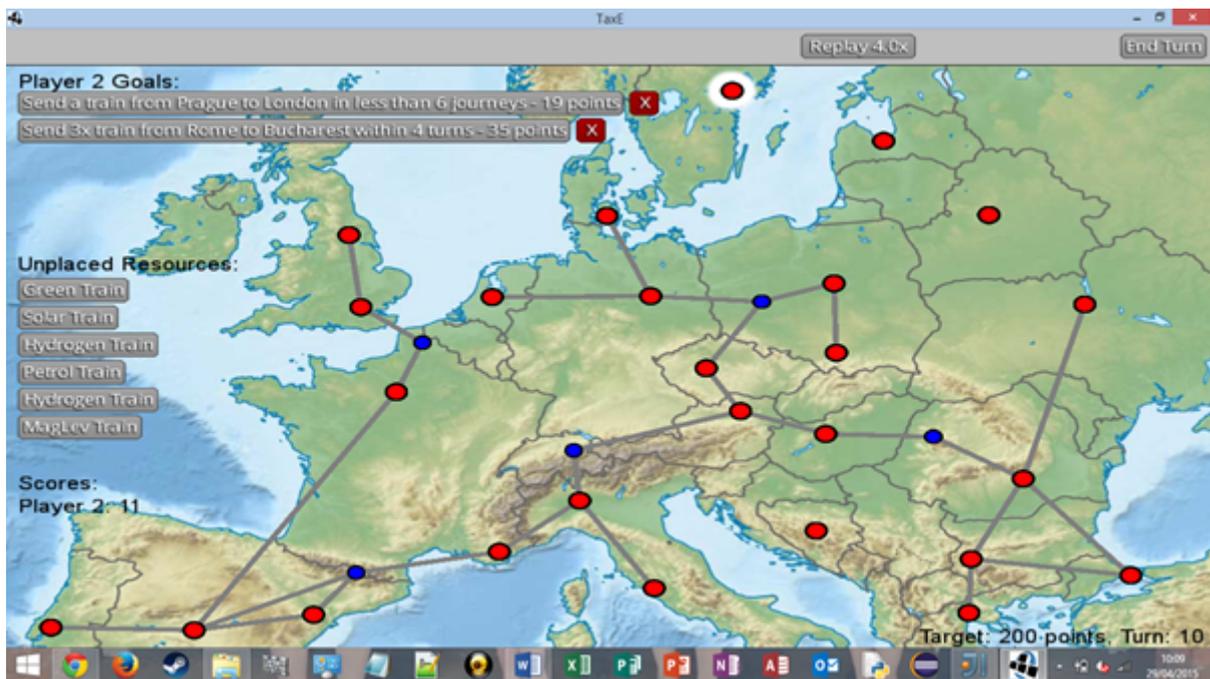
Visual GUI representation of removed connection between Lille and Amsterdam after being prompted at top of screen to select two stations:



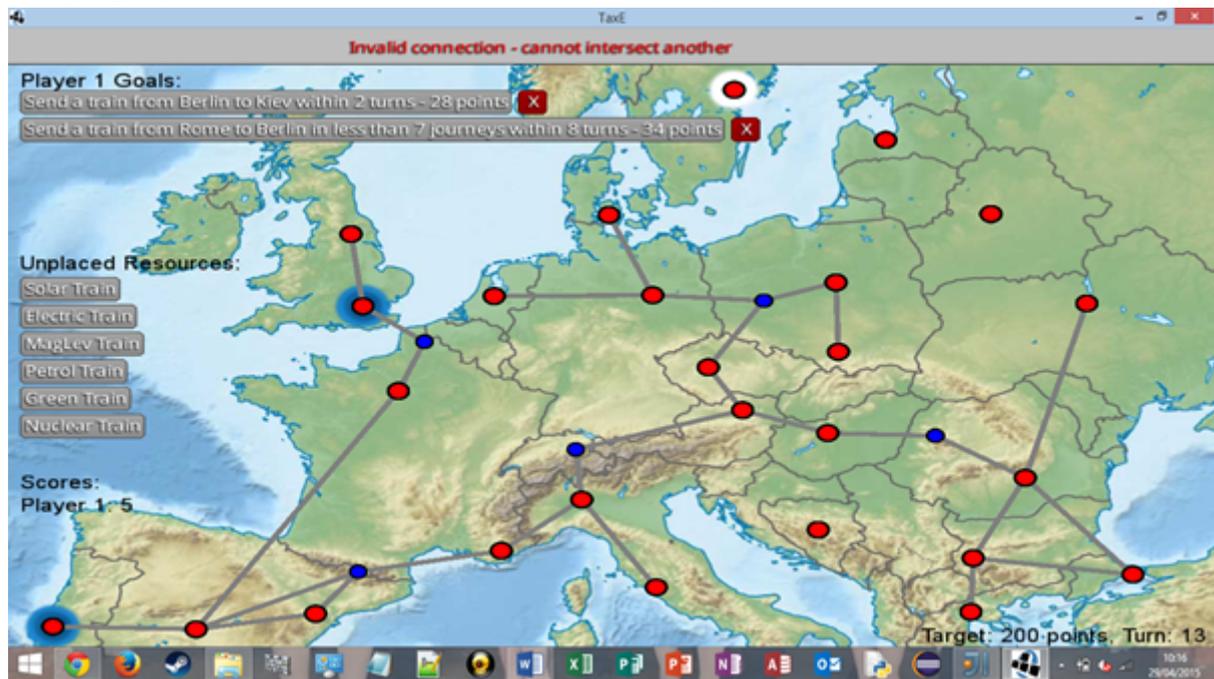
Track Modification screen with message prompting user to select two stations at top centre of screen:



Visual GUI representation of adding a track back into the game between London and York:

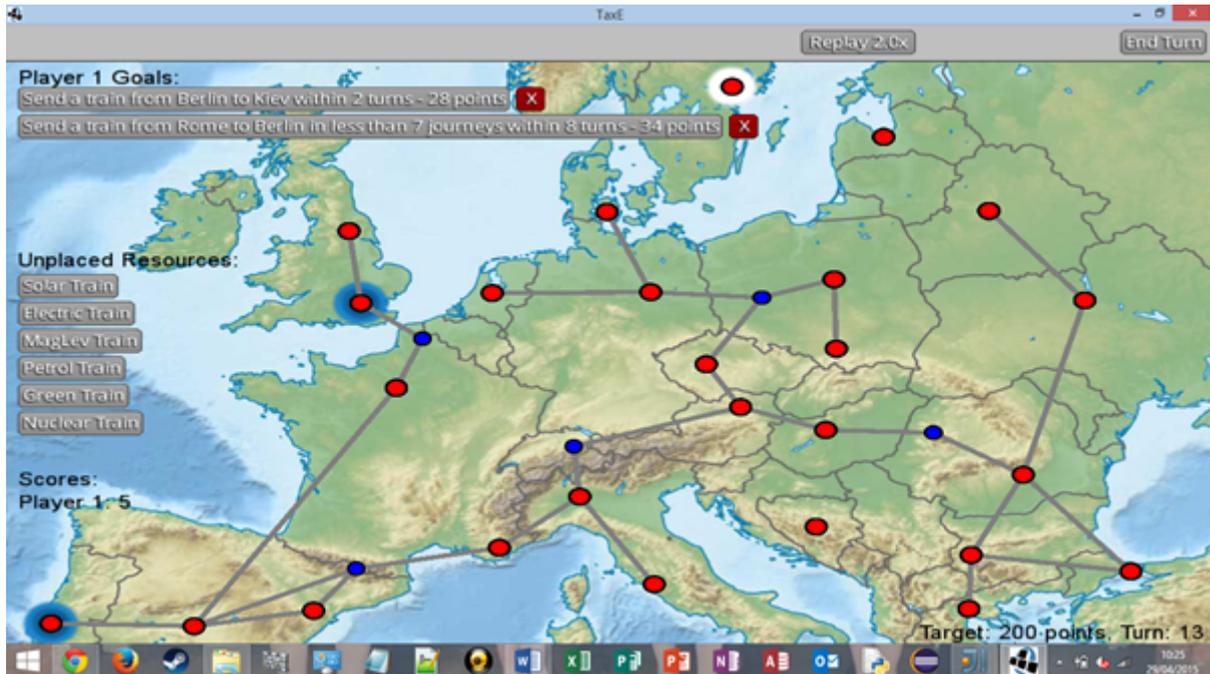


Error message being displayed at the top of the screen when an invalid connection occurs (e.g. attempting to place a new track between Barcelona and Paris):

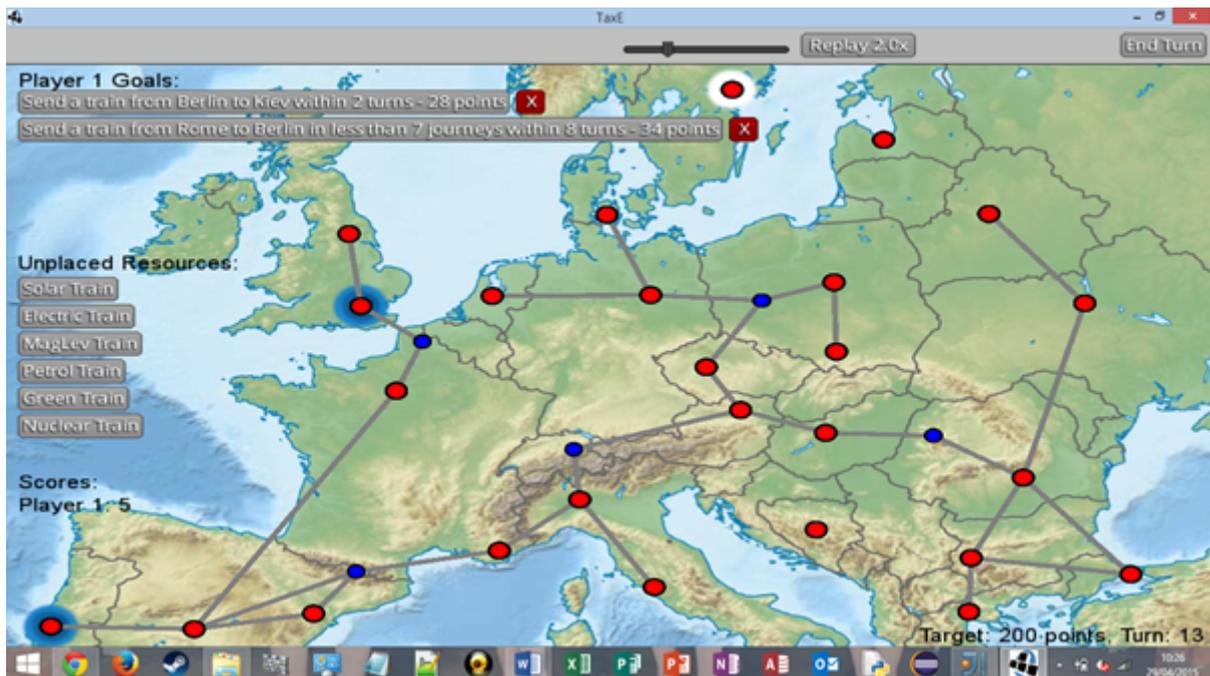


2. Replay System

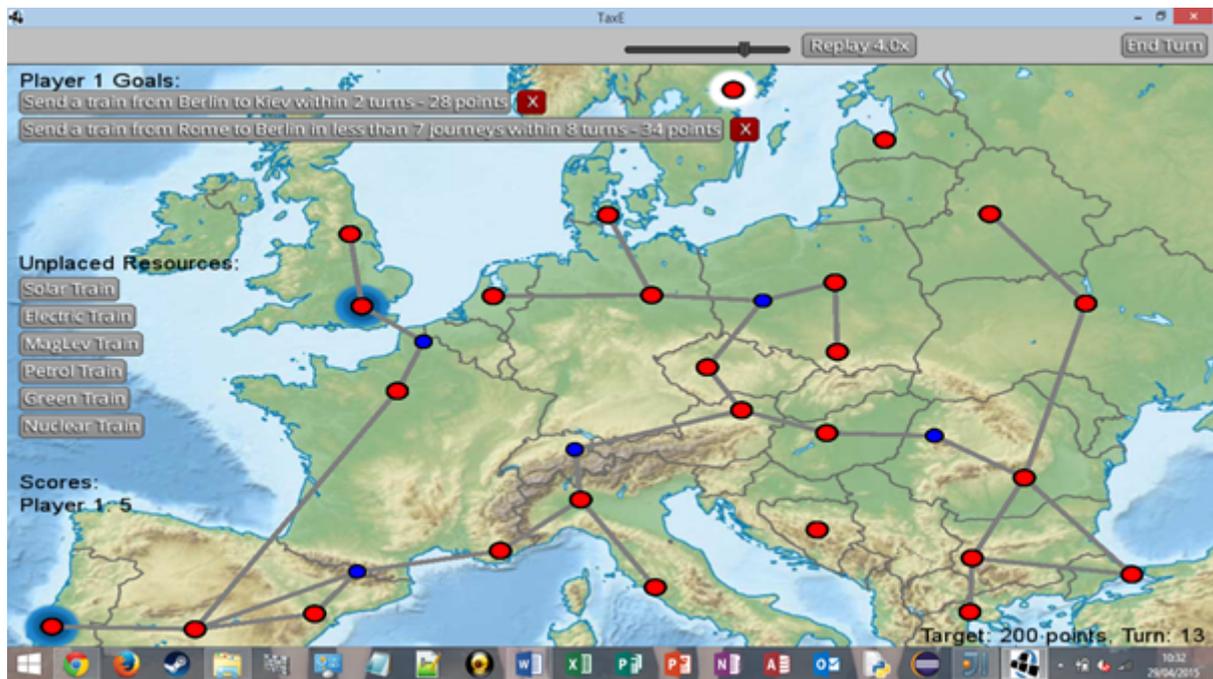
GUI showing replay button and slider in top right of screen. Slider adjusts speed value specified in the button text and hides when not in use:



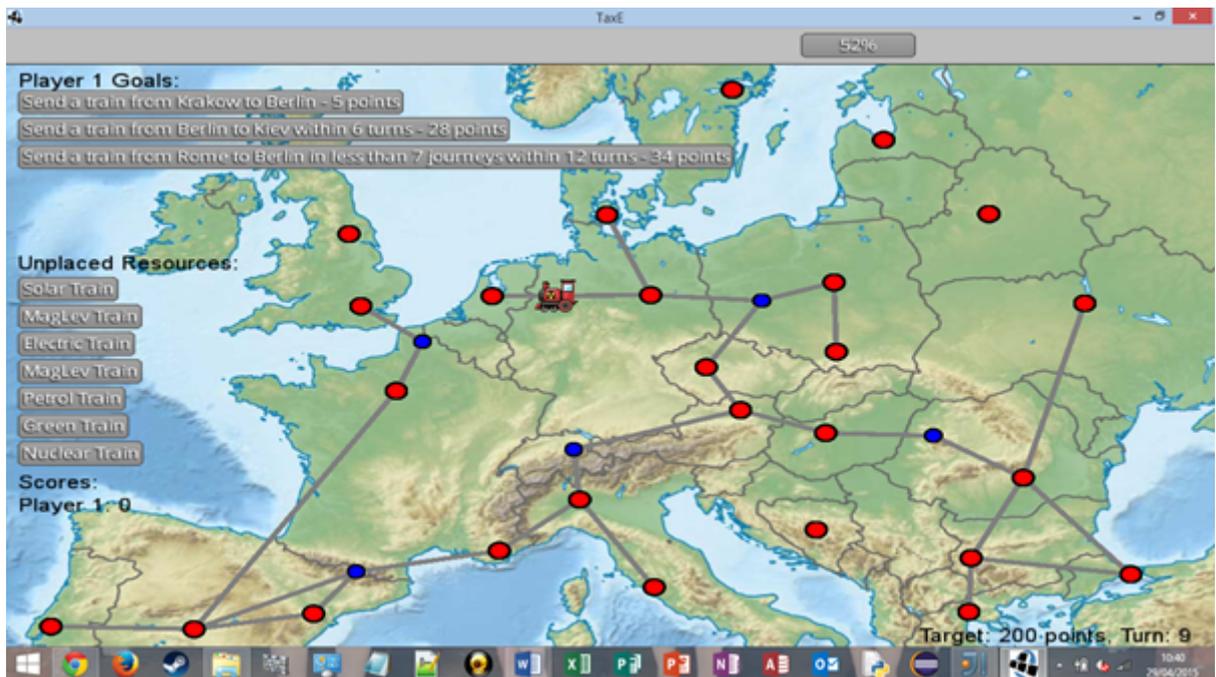
Slider becomes visible when replay button is hovered over:



Speed value of replay is adjusted by moving slider right (increase) or left (decrease):



User enters replay mode once replay button is clicked:



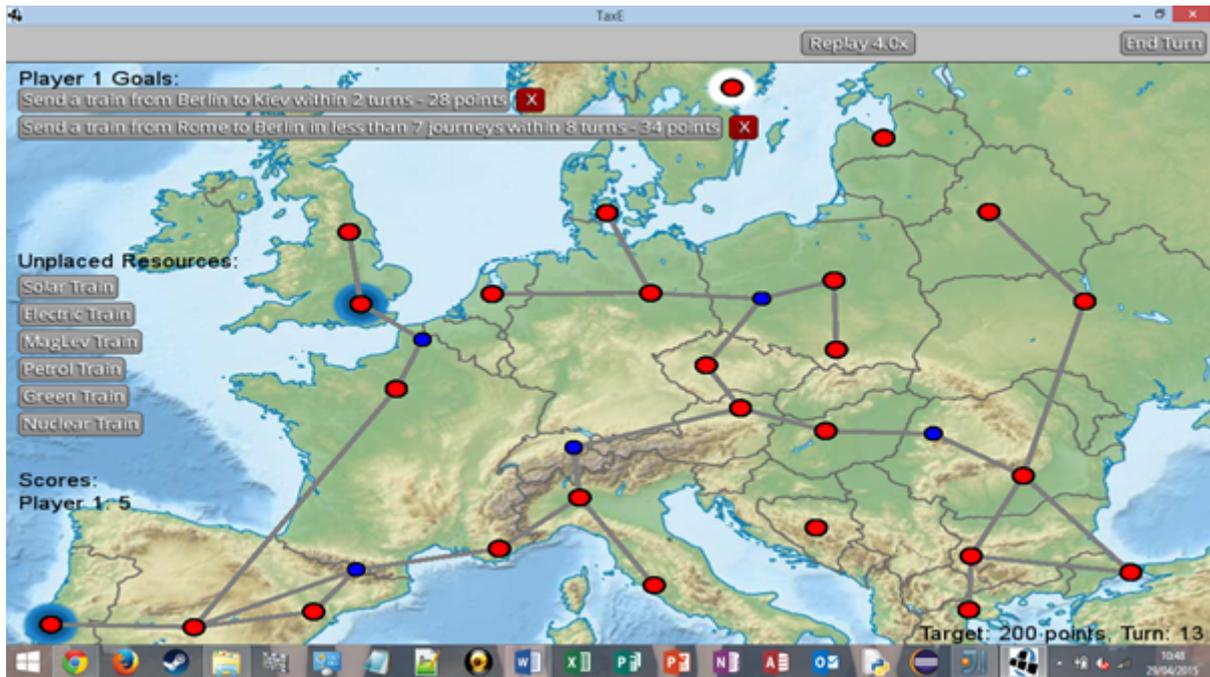
3. Cross-platform game for mobile devices

The system allows the game to be available on iOS and Android operating systems to become portable and usable on mobile devices or tablets:

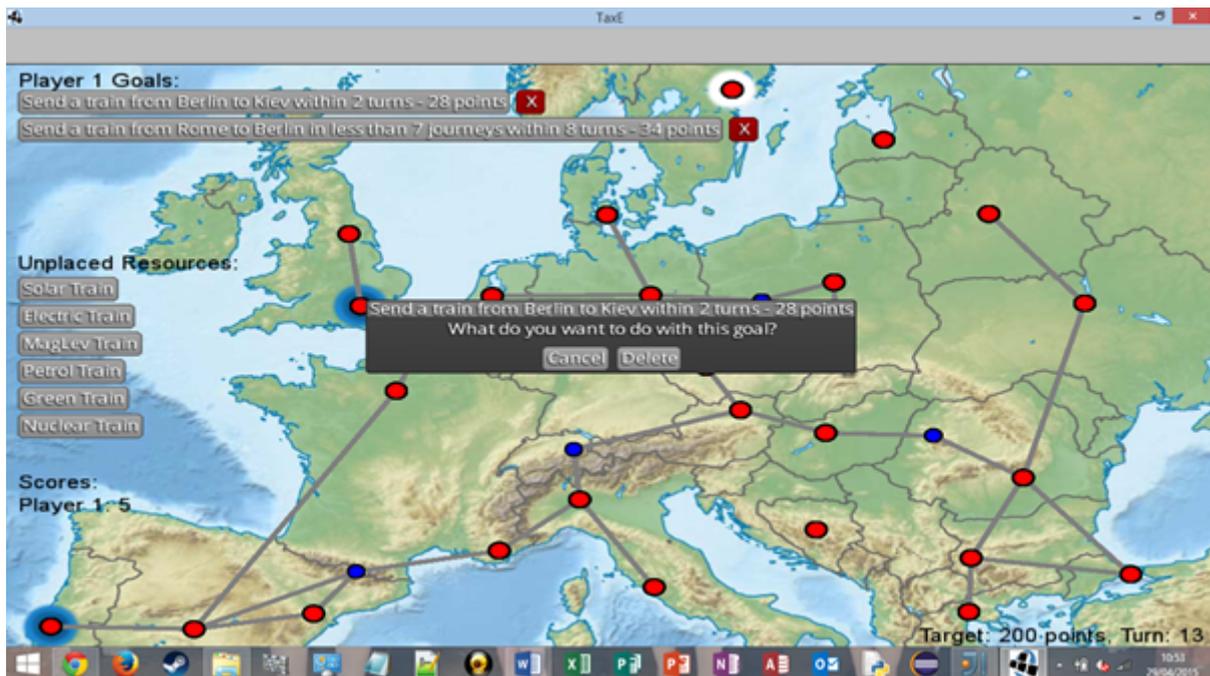


4. Cancellation of Goals

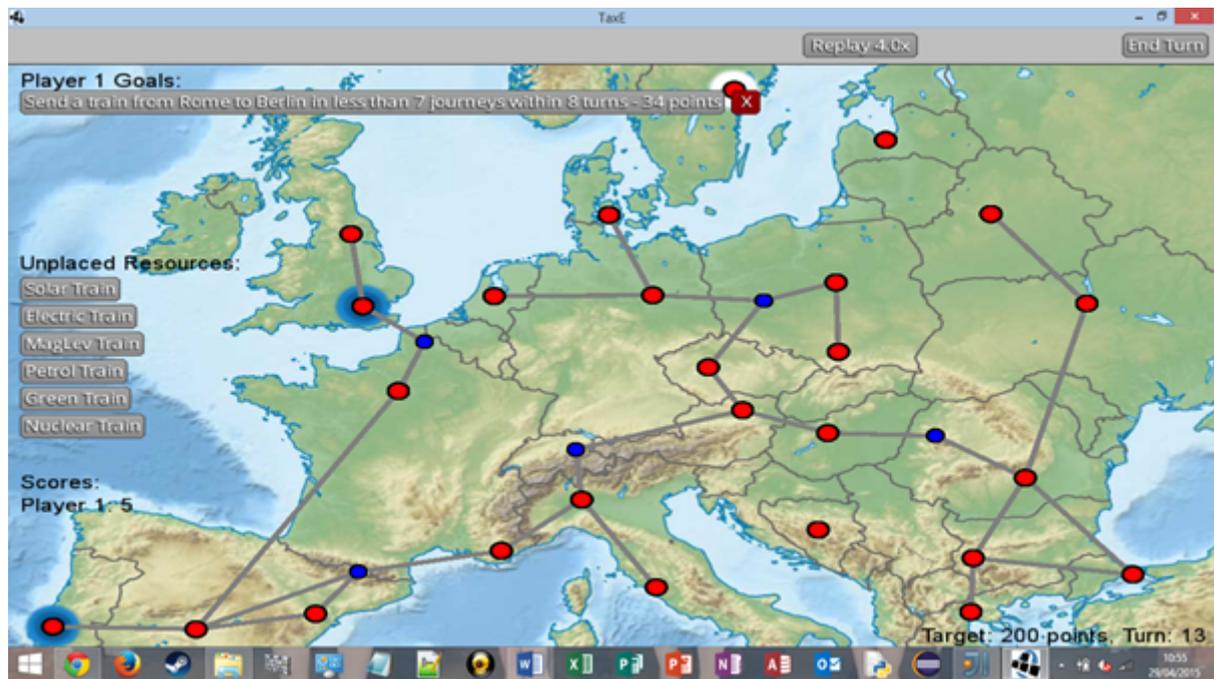
GUI shows an X button next to each goal to remove it. Removing goal 1 by clicking the X button next to goal one in the top left hand corner:



User is prompted with notification to verify that they wish to remove the stated goal:



Goal 1 is now removed from the list:



**EEP's TaxE Software Engineering Project (Appendix O, Software Engineering
Project (SEPR) Group Open Assessment)
Perfective Changes for Assessment 4**

1. Lack of MVC Structure

As stated in the extension report, team ADB followed on from team FVS with the architecture in which they followed an MVC approach similar to that shown below in Figure 1:

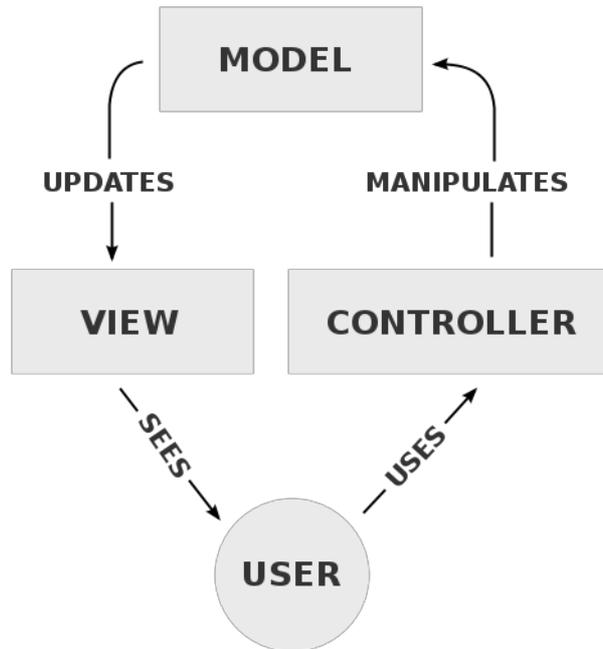


Figure 1 - MVC diagram showing the exchanging of information [1]

As stated in FVS's architecture report [1]:

"We were not as successful with separating the Controllers from the View code. In our project, our View objects are Actors. These contain very little logic of how they are display because this has been done in the controllers. For extra separation, our controllers could deal with user input and game events exclusively, and not modify Actors, we could then have some other class which deals with ViewLogic, but this would add some code duplication and more complexity (in terms of another layer) for little benefit. An answer on Stack Overflow [2] generalises our case by saying "a pragmatic balance between an idealized MVC and strict enforcement of DRY", where we have chosen to lean more towards the DRY side."

([2] in the direct quote relates to [2] in the Appendix reference list)

References

1. FVS, "FVS Architecture Report", Internet: <http://www-users.york.ac.uk/~oeh503/fvs/> , [Feb. 23rd 2015]
2. tvanfosson, "How much logic is allowed in ASP.NET MVC views?," StackExchange, 14 January 2009. [Online]. Available: <http://stackoverflow.com/a/443213/138251>. [Accessed 18 January 2015].
3. Cliparts, "Free to Use & Public Domain Train Clip Art - Page 2", Internet: <http://cliparts.co/clipart/2363980> , [Mar. 6th 2015]

**EEP's TaxE Software Engineering Project (Appendix P, Software Engineering Project
(SEPR) Group Open Assessment)
Change Log**

This document contains records of changes made to the game for assessment 4, including the status of tests or any new tests written. It should be used for traceability; justification of changes can be found within the Extension report.

Corrective Changes

"Fixing faults and errors"

No.	Class & Method(s) Changed	Description of fault/error	Description of change/fix	Any new tests written?	Did all previous tests pass? (Regression Testing) If not, state severity of error	Author
1.	GameScreen MainMenuScreen CustomTexture(Added) TaxeGame All subclasses of Actor class	Game only had a set window size which was too small to for players to experience complete immersion. A better solution would be supporting full screen mode for every screen size.	1. Added a StretchViewport to the Stage of GameScreen in order to support resizability. 2. Added class CustomTexture which extends Texture and in the constructor sets a linear filter to the object. This way images which use CustomTexture will not lose their quality when window is resized. 3. Added linear filter to the font used in the game. This way all text will not lose it's quality when window is resized.	No	Yes	SK
2.	Assets folder	Trains looked like a five year old child had drawn them.	Added new train images in assets folder.	No	Yes	SK
3.	MainMenuScreen, ConnectionActor,	Game only had a set window size which was too small to for	Added a Stage and Viewport with Camera to MainMenuScreen. The projection matrix of	No	Yes	SK

	TopBarActor	players to experience complete immersion. A better solution would be supporting full screen mode for every screen size. With this fix the game has become fully resizable and runnable on smaller screen devices(e.g Android)	the shape renderer, used to render the menu buttons, is set to that of the viewports camera. Since ConnectionActor and TopBarActor are rendered using a shape renderer, the same technique is used, where the shape renderers projection matrices are set to that of the game screen viewport's camera.			
4.	StationActor, CollisionStationActor, StationController	Lack of Inheritance in StationActor and CollisionStationActor(junction actor) caused game to crash when modifying connections to/from junctions	Added new constructor to StationActor. CollisionStationActor now extends StationActor like it is supposed to. Removed unneeded methods for drawing CollisionStations seperatly from normal stations in StationController.	No	Yes	SK
5.	TopBarController	Many events were made to be triggered by click listeners on buttons, such as confirming the route of a train or triggered directly by casual events, such as natural disasters. This was not compatible with replays.	We changed these mechanism to set a specific new state for the game. After that, some new <i>StateListeners</i> would take care of executing the relative action the action. For example, when clicking the confirm button, the state would change to "CONFIRMING", then a StateListener would be activated and create the route for the train.we decided to change	No	Yes	AEF

6.	All actor classes, GenericActor	All of the actor classes for the models extended the Image superclass but were not related by any ancestor accessible to us.	A new GenericActor class was created. All model actors were set to be children of this class.	No	Yes	AEF
7.	Most models, HasActor<Type>	Many models contained as a property the relative Actor. This was a problem because as the Actor is unserialisable, it was impossible to serialize the Game data. Moreover, it is bad practice to have views mixed with the models. Finally, all models used the same signature to access the relative actors, but there was no interface or common ancestor class in place.	A new HasActor<GenericActor> was created. This would act as an ancestor for all model classes. The type parameter contains the type of the relative actor. A static hash map is created to contain pairs of (HasActor, GenericActor) and the common method "getActor" was created, allowing to link each model to the relative actor, which is stored in the HashMap.	No	Yes	AEF
8.	HasActor and all child classes.	Using the Model (HasActor) itself as a key in the global HashMap was not conveniente. After being	A new string unique ID was associated to each child. On the first time an Actor is set for the model (HasActor), a new unique ID string is generated. This can easily be serialised and deserialised and used,	No	Yes	AEF

		<p>serialised and deserialised from a snapshot, any object may -even if identical- have a different reference. This would case Actors to get lost in the HashMap.</p>	<p>after deserialisation, to reach the corresponding Actor even if the model object has not the same reference, i.e. is not identical to its predecessor.</p>			
--	--	---	---	--	--	--

Additive Changes

“Adding functionality or features”

Assessment 4 required us to implement two new features:

- Replay mode
- Track modification

The cross-platform portability of the game was an additional feature that was not necessary to implement, but was recommended by the clientele, hence it was added.

The following new user requirements were devised for these new features and given a priority level (Essential/Preferable/Optional).

Gameplay

- User Req Replay.1.1 - Essential
 - Players must be able to replay all of their previously executed, hence see their train’s movements replayed in-game and corresponding scores updated.
- User Req Replay.1.2 - Essential
 - Players must be able to specify how fast they wish the replay to be, e.g. 2X the normal speed.
- User Req Modify.2.1 - Essential
 - Players must be rewarded with the ability to modify one connection, upon completing a goal, by being given a “Connection modifier resource” (a token that can be spent).
- User Req Modify.2.2 - Essential
 - Players must be able to spend their connection modification resource on adding a new connection between stations/junctions, which does not already exist.
- User Req Modify.2.3 - Essential

- Players must be able to spend their connection modification resource on removing an already existing connection between stations/junctions.
- User Req CP.3.1 - Preferable
 - Players should be able to play the game on a mobile device (i.e. on the Android and iOS operating systems)
- User Req Goal.4.1 - Preferable
 - Players should be able to remove a goal they do not wish to continue with

UI

- User UI Req Replay.4.1 - Essential
 - Players must be able to use a slider to specify the speed of the replay mode, with leftmost position indicating normal speed, and rightmost indicating the fastest possible replay speed.
- User UI Req Replay.4.2 - Essential
 - Players must be able to press a 'Replay' button located on the top bar, which will start a replay at the specified speed.
- User UI Req Modify.5.1 - Essential
 - If a player has a connection modification resource to spend, a button should appear at the top right which when pressed allows a modification to be made. It should indicate how many modifications the player can currently make.
- User UI Req Modify.5.2 - Essential
 - The UI must indicate a connection has been added or removed from the game by drawing/removing the track's connection image.
- User UI Req Modify.5.3 - Preferable
 - The UI should show an error message when a connection is invalid or could not be made.
- User UI Req CP.6.1 - Preferable
 - Players should be able to have a smaller version of the UI that can be presented onto an Android or iOS device, particularly a smart phone.
- User UI Req Goal.7.1 - Preferable
 - Players should have a button next to each goal that will remove the goal when clicked.
- User UI Req Goal.7.2 - Preferable
 - Players should have a notification to check whether they wish to go through with the removal of the goal

The following functional requirements were derived from these user requirements and each were placed with a respective priority level.

- Func. Req Replay.1.1 - Essential

- The system must record and keep snapshots of every change of state of the game, keeping track of; the map, goals, resources, obstacles, trains and player's score, at every stage of the game.
- Func. Req Replay.1.2 - Essential
 - The system must be able to iterate over every recorded snapshot of the game at a specified speed.
- Func. Req Replay.1.3 - Essential
 - The system must reanimate the movements of trains during the replay, such that they move along exactly the same original route.
- Func. Req Replay.1.4 - Essential
 - The system must redraw the map at each iteration of the replay, to reflect any modifications that are made to connections during the game.

- Func. Req Modify.2.1 - Essential
 - The system must reward a player with a connection modification resource upon completing a goal. This can be spent once, allowing the player to enter the modify connection mode and make one modification. Connection modification resource will not be removed if an invalid connection occurs.
- Func. Req Modify.2.2 - Essential
 - The system must allow a player to add a connection, if it does not already exist and does not intersect an already existing connection.
- Func. Req Modify.2.3 - Essential
 - The system must allow a player to remove a connection, providing it does not currently have a train travelling along it.
- Func. Req Modify.2.4 - Essential
 - The system must update a train's route whenever a connection is removed which the train was due to travel along. The train must stop at the station/junction before the removed connection, so the player can reroute it.
- Func. Req Modify.2.5 - Essential
 - The system must not allow goals to be generated which are not completable, due to the origin and destination station not being connected by any sequence of connections.

- Func. Req CP.3.1 - Preferable
 - The system should be playable on mobile based operating systems, particularly Android and iOS. The mobile version of the game should have the same functionality and theme as the desktop version.

- Func. Req Goal.4.1 - Preferable
 - The system should allow a goal to be removed from a Player's goal list to allow them to receive another goal in the event that one may become not completable or wishes to be dropped.
- Func. Req Goal 4.2 - Preferable
 - The system should allow the user to cancel the removal of a goal in the event that they change their mind

No.	Class & Method(s) Changed	Description of new functionality	Relevant requirement no.	Any new tests written?	Did all previous tests pass? (Regression Test)	Author
1.	MapController(Added), RouteController	MapController class created. The MapController is responsible for adding and removing connections during a game. methods addConnection and removeConnection created. Added reroute method to RouteController in order to be able to update a route of a train if a connection is removed from it.	User Req Modify.2.1, 2.2, 2.3 User UI Req Modify.5.1, 5.2, 5.3 Func. Req Modify.2.1, 2.2, 2.3, 2.4, 2.5	1.1, 1.2, 1.3	Yes	SK
2.	Game.Snapshot	Game.Snapshot class created. A Snapshot represents the state of a Game at a given point in time.	User Req Replay.1.1, 1.2 User UI Req Replay.4.1, 4.2 Func. Req Replay.1.1, 1.2, 1.3, 1.4	3.1	Yes	AEF
3.	Game.loadSnapshot Game.createSnapshot	Two methods created. Game.createSnapshot saves the data of the Game into a new Snapshot. Game.loadSnapshots, given a Snapshot, loads the data from it into the current game.	User Req Replay.1.1, 1.2 User UI Req Replay.4.1, 4.2 Func. Req Replay.1.1, 1.2, 1.3, 1.4	3.1, 3.2	Yes	AEF
4.	Connection Modifier	A new class which extends Resource. It acts a "token" which indicates a player can make one connection modification. Once they make the modification, the resource is spent.	User Req Modify.2.1, 2.2, 2.3 User UI Req Modify.5.1, 5.2, 5.3	2.1	Yes	RC

			Func. Req Modify.2.1, 2.2, 2.3, 2.4, 2.5			
--	--	--	--	--	--	--