

FVS Change Report

Choice of project

We wanted to choose a project that was based on our original project to make the transition between projects as easy and smooth for us as possible. Fortunately, our project from Assessment 2 was quite popular and we were presented with plenty of good choices. DUS stood out with its high quality improvements and features that had enhanced the playability of the game, which in turn gave us ideas for further improvements that would suit the game perfectly. Additionally, DUS' code seemed the most reasonable out of all realistic candidates and we felt that it would be the best project for supporting future extensions. Also, DUS promised to be and has been very proactive with helping us making the change of management as painless as possible with always being willing to answer questions about their code.

Development approach

As for the software engineering approach we used Scrum to some degree. Before Easter break we held a sprint planning meeting to establish the work to be done. We prepared the sprint backlog based on the requirement changes and general ideas to improve the game, the backlog detailed the work to be done and assigned people to specific tasks. We chose four weeks for the length of the sprint with the deadline set as the first day of the Summer term. The rather long sprint duration was chosen to allow the team time for holiday celebrations and general time off between the terms.

We did not hold daily scrum meetings due people working from home, but when questions arose or discussions were needed, team members could do it in Facebook team chat. At the beginning of this term we held the Sprint review session and also prepared another smaller Sprint for the remaining work which consisted entirely of bug fixing, general enhancements, testing and report writing. Obviously, we did not strictly follow all the rules of Scrum, but this kind of very relaxed approach has worked for us for previous assessment and was sufficient for this assessment as well.

Extension summary

The project was changed to include Replay mode (Table 1), i.e add a replay mode to the game, such that a selected set of turns can be recorded and then 'replayed' quickly and Track modification mechanisms (Table 2) such that players can dynamically add new track, and remove existing track from the game. Additionally, we modified our game's UI and undertook some playability improvements to make the game more playable and marketable by improving the overall player experience (Table 3-8). For marketing purposes we also ported the game to Android to be able to present the game on a tablet to potential customers (Table 9). The Replay mode and Track modification features had the highest priority and the rest of the improvements were done as time permitted with lower priority.

The UI changes in particular were motivated by the need to provide a good end-user experience that met the users typical expectations of the system and would allow for future expansion of the game to include additional form factors and functionality. The UI introduced in Assessment 2 was purely for the purposes of having a means for the user to interact with the game, and the same approach was taken in Assessment 3. Therefore, the redesign of the UI was long overdue. It also fell short in the gaming area with regards to the

experience level it provided, so additional options for configuring the UI and audio elements needed introducing.

We purposefully did not try to spend much time refactoring to enable us to get all the necessary features out as soon as possible, but we think that the previous team's code quality could've been improved before starting with our features. Subsequently, the code of our own features suffered a bit and is messy and coupled at places, but the rise in code complexity was unavoidable either way.

Appendix A displays the overview of all the classes in our project, the ones marked green are classes that were created for this assessment, the yellow ones are the modified classes. Rest of the classes were unchanged.

Modifications

Table 1.

Detail of change made
Mechanism for allowing the players to add or remove tracks on the map while the game is played, the UML in Figure 1 displays some of the the relationships and inheritances of this change. (Green boxes are new classes)
Reason for change made
Changes in the requirements (requested by the Client) (User Requirement 9 and System Requirements 9.1, 9.2, 9.3)
Affected elements
Affected: Button, DialogButtonClicked, ResourceController, ResourceManager, Map New classes: ConnectionClicked, DialogResourceConnectionModifier, ConnectionModifier
Specific challenges encountered
No existing way to check whether a route uses a certain track/connection (routes are coded as lists of stations), had to create a method specifically for that. Numerous bugs encountered with goal allocation and dijkstra scoring mechanism, such as goals allocated with unreachable stations and having maxInt as the score reward. A bug when cancelling the usage of engineer/obstacle/connection modifier resource after having selected one station (the resource stored the selection for the first station instead of resetting it).
Git commits
<ul style="list-style-type: none">• ef35070• 9e65e5b• 77f33e1• 1eac8c3• 17d6b32
Relevant testing
<ul style="list-style-type: none">• System tests 8 - 10• Unit test dijkstraUpdateTest

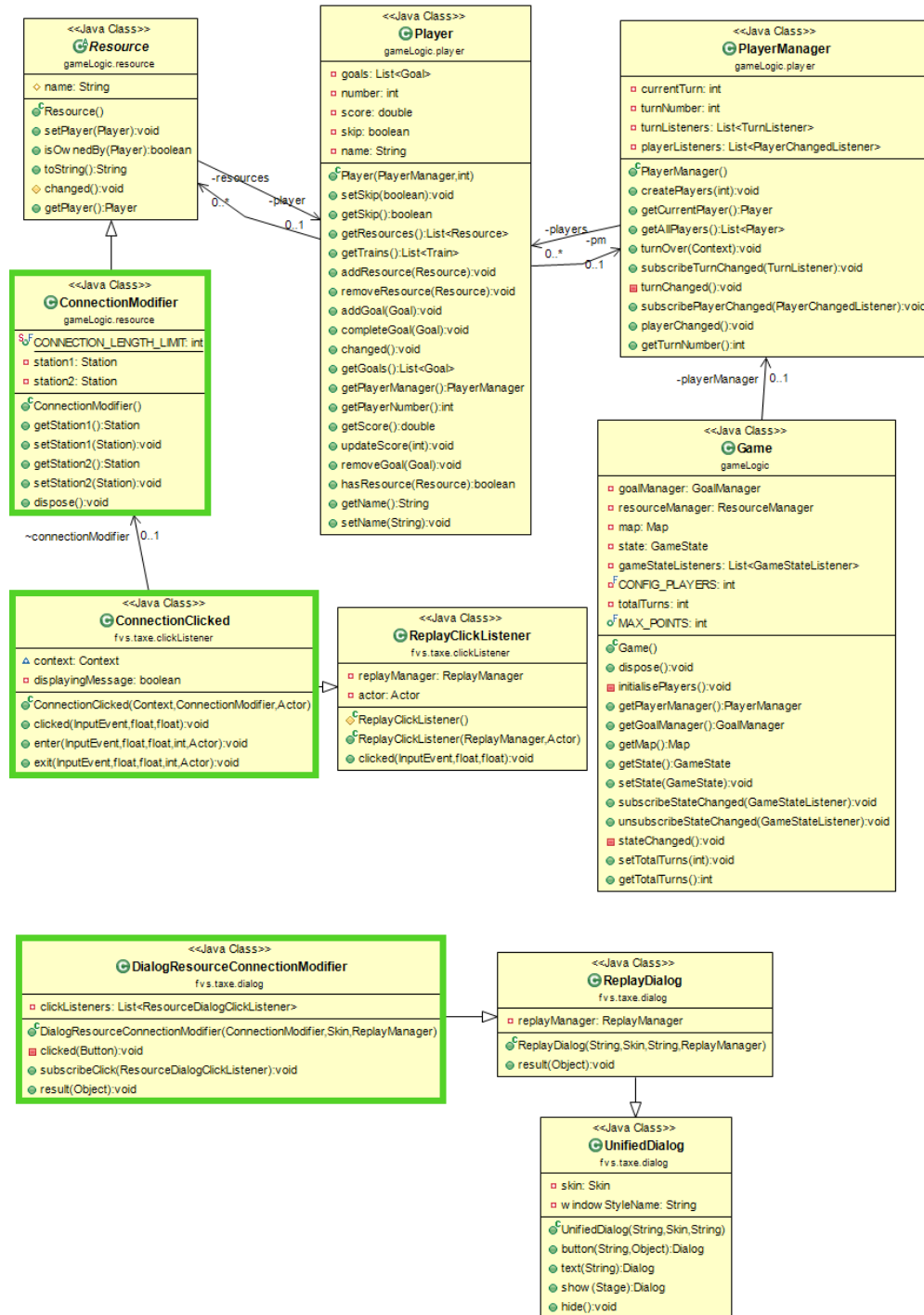


Figure 1. UML of add or remove tracks feature

Table 2.

Detail of change made
To be able to replay the game, we first give each actor (clickable screen element) a distinct name. Each click is logged and stored in a ReplayManager. When the player wants to replay the game, a whole new instance of the game is created, and each actor is added with the same distinct name, we then “replay” the

clicks on each actor. We also store the random seed used to generate the Goals and Resources and use the same seed in the replay to make sure random events happen in the same way.

This necessitated a way to control movement between replay and to control replay. ReplayControlsController introduces a table in the lower-right hand of the map. On GameScreen this shows a “Start Replay” button and control buttons on ReplayScreen. To maintain MVC separation, ReplayManager can only interact with the options via ReplayToggleListener to reflect changes in the replay state at the model end.

Reason for change made

Implementation of requirements requested by the client. These requirements are documented in: System requirements 8.1 and 8.2 and user requirement 8.

Affected elements

Classes Added:

- gameLogic.replay.ReplayManager
- gameLogic.RandomSingleton
- fvs.taxi.ReplayScreen
- fvs.taxi.StageNamedActor

All Dialogs in fvs.taxi package now extend ReplayDialog. All click listeners extend fvs.taxi.clickListener.ReplayClickListener

Specific challenges encountered

Two approaches to implement this feature failed. The first approach was to store screenshots but this would use a lot of memory to store if the game length was long. Even if this wasn't an issue, there would be too much reading from disk to “play” these screenshots back. The second approach was to store the x y coordinates of each click, and use them to replay the clicks on each actor, but after attempting this, we found that it was too unreliable.

When implementing controls for replay, there was the possibility of the user interacting with the game being replayed and influencing the outcome to deviate from the recorded actions. This issue was nullified by placing the controls for replay on a separate stage in ReplayScreen and only accepting input to that stage, preventing interaction with other elements on screen.

Git commits

- [332db87](#), [66011a2](#), [1b5434b](#), [b150c17](#), [f6860b7](#), [7394198](#), [e2b300c](#), [9e959a6](#), [a4c6909](#), [787d131](#), [b0f811f](#), [b6e9f18](#), [ae30daa](#), [b4c17f0](#), [9a3520c](#), [d8db5cf](#), [6ea5a01](#), [369e7e7](#)

Relevant testing

- System tests 1 - 7
- Unit test addClickToReplayTest

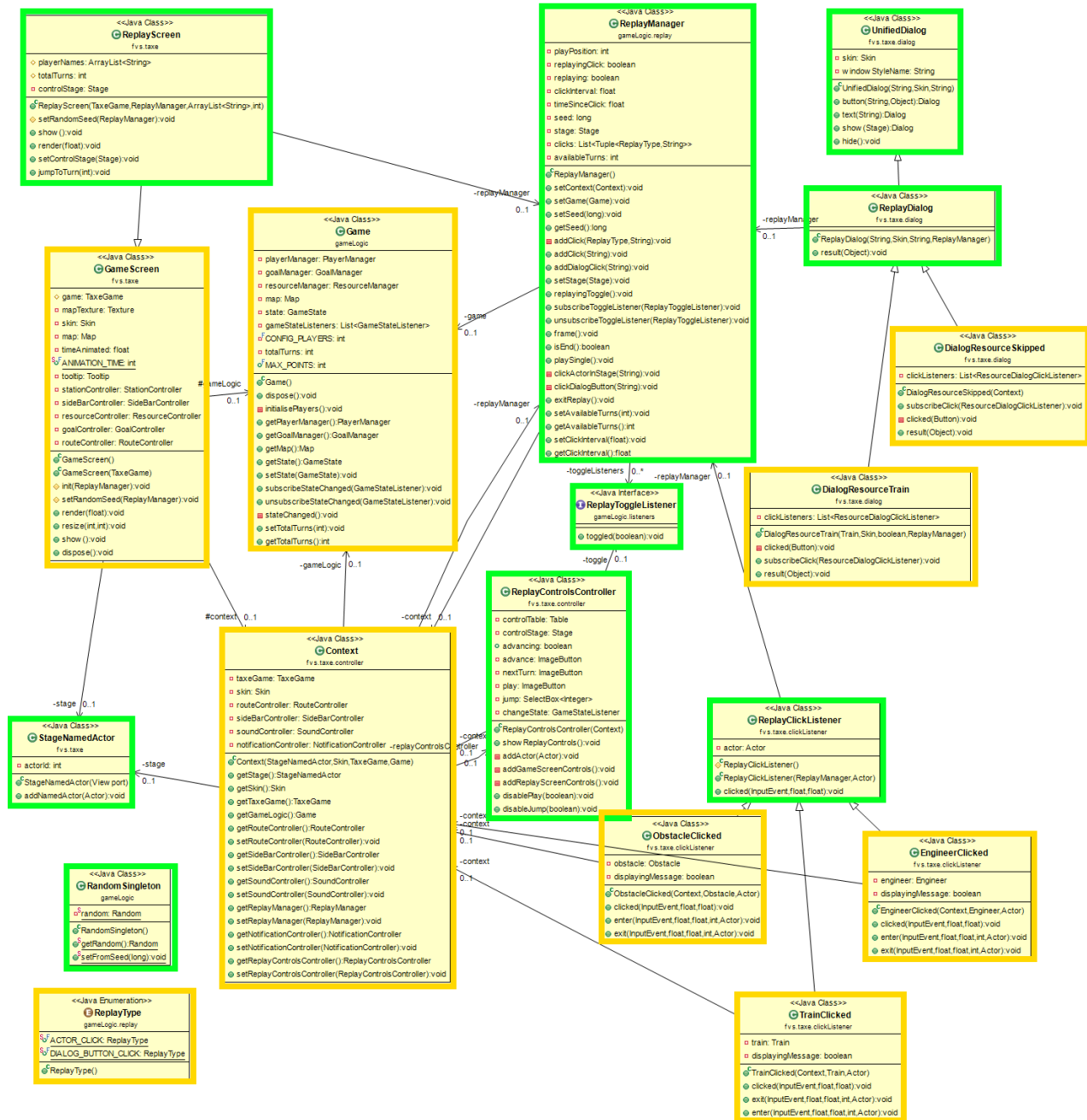


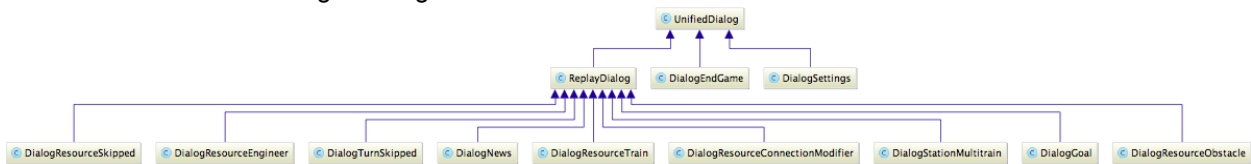
Figure 2. UML of Replay feature (some pre-existing but modified classes omitted for readability)

Table 3.

Detail of change made
Enabled scaled resizing of window
Reason for change made
Allows the game to be played on screens of different sizes and makes it more portable to other platforms in the future using LibGDX. Better meets User Requirement 4.
Affected elements

MainMenuScreen, GameScreen
Specific challenges encountered
The first resizing method used did not scale the window content but stretched to fill the window area, resulting in some difficult to use graphic elements and so a new method was used to ensure the content of the window always maintained proportions
Git commits
50bc90d , 3d93160
Relevant testing
<ul style="list-style-type: none"> System test 11

Table 4.

Detail of change made
<p>Numerous UI and playability improvements: Moved options from being at the top and left to the right and restyled UI elements to have a consistent appearance, including a new font. New Dialog class, UnifiedDialog, also implemented to make it easier to change dialog message appearances. Also, display station acronyms on screen to make it easier to find stations. Additionally, highlighting stations upon clicking when using Engineer/Obstacle/Connection modifier resources.</p> <p>The UnifiedDialog, combined with the introduction of the ReplayDialog, have caused significant changes to the class structure for Dialog messages as detailed below.</p>  <pre> classDiagram class UnifiedDialog class ReplayDialog class DialogEndGame class DialogSettings class DialogResourceSkipped class DialogResourceEngineer class DialogTurnSkipped class DialogNews class DialogResourceTrain class DialogResourceConnectionModifier class DialogStationMultitrain class DialogGoal class DialogResourceObstacle UnifiedDialog < -- ReplayDialog UnifiedDialog < -- DialogEndGame UnifiedDialog < -- DialogSettings ReplayDialog < -- DialogResourceSkipped ReplayDialog < -- DialogResourceEngineer ReplayDialog < -- DialogTurnSkipped ReplayDialog < -- DialogNews ReplayDialog < -- DialogResourceTrain ReplayDialog < -- DialogResourceConnectionModifier ReplayDialog < -- DialogStationMultitrain ReplayDialog < -- DialogGoal ReplayDialog < -- DialogResourceObstacle </pre> <p>ReplayDialog and UnifiedDialog have been added, with all other Dialog classes having been modified to inherit from these. DialogSettings and DialogResourceConnectionModifier are also new additions, but these classes perform functions which is not relevant to this change.</p>
Reason for change made
Make the game more marketable and playable with a consistent, user-friendly interface. Also, highlight information that is important to the player by making certain pieces of text larger than others (Goal headlines larger than turn number, scores larger than player name, etc). Highlighting stations upon clicking to make user clicks more responsive. Improvements around User Requirement 4.
Affected elements
GameScreen, MainMenuScreen, TopBarController, GoalController, ResourceController, CollisionStationActor, Selectable, StationActor, DialogButtonClicked, Station
Specific challenges encountered
The use a third party skin editor tool (https://github.com/cobolfoo/gdx-skineditor) and font packer (https://github.com/mattdesl/gdx-fontpack) was required to allow the skin to be customised and reliably generate the skin code for the game. The skin tool did not have functioning support for font alterations so these had to be done using a separate tool which created bitmap fonts which were then manually added to the skin file.
Git commits

7bd6cc8 , d1277e8 , fbd5c37 , e8d7a7f , 354b5bb , 279cc41 , 338f4b4 , 3ee2ed5 , f4f8891 , a4c6503
Relevant testing
<ul style="list-style-type: none"> System tests 12 & 13

Table 5.

Detail of change made
Add background music and sound effects to the game, and settings to control them.
Reason for change made
Make the game more playable and marketable by improving the overall player experience and introducing another method of providing feedback to the user. Settings allow these to be fine-tuned and removed if the player does not want to use them without having to modify system settings.
Affected elements
SoundController, GameScreen, MainMenuScreen, DialogSettings, Context
Specific challenges encountered
Settings needed to persist between uses of the application to provide the behaviour that a user would expect. To provide cross-platform usability of the game native features, such as registry values or environment variables, could not be used and the decision was taken to store settings in a file in the same directory as the game runnable using the Gdx.files.local location provided by the application to prevent any cross-platform issues.
Git commits
<ul style="list-style-type: none"> df7a436 b1eb953 d351776 70fdef5
Relevant testing
<ul style="list-style-type: none"> Unit tests defaultVolumeTest and saveSettingsAndLoadTest

Table 6.

Detail of change made
Allow the user to customise the player names and number of turns at the start of a game
Reason for change made
Make the game more playable and marketable by allowing limited customisation of the game environment and increasing the possibilities for gameplay by allowing both shorter and longer games in addition to endless games. Improves system requirement 1.3 by making a more natural distinction between players on the same machine.
Affected elements
GameScreen, MainMenuScreen, GameLogic, SetupScreen, ReplayScreen
Specific challenges encountered
Once the ReplayScreen had been introduced, which resets elements of the gameLogic when it is opened, player customisations would not be honoured and so must be reset during the transition. This was achieved

by adding some code which ran in the ReplayScreen overriding that in the main GameScreen to ensure correct display.
Git commits
<ul style="list-style-type: none"> • 47caf3f • b49b9c6 • b88f53a • 84d542f • 550ffdb
Relevant testing
<ul style="list-style-type: none"> • System test 16

Table 7.

Detail of change made
Add News screens and an activity feed to the game delivering content based on in-game events.
Reason for change made
Provide increased playability of the game for the user as well as introduce reminders of recent events to assist players with planning their moves. Also makes a step towards some sort of achievements system in the future to reward player actions further.
Affected elements
NotificationController, NewsDialog, GameScreen, Context, GoalController, TrainMoveController, ResourceController
Specific challenges encountered
A large amount of online research was required to have a correctly functioning Notification area which allowed scrolling to take place whilst sizing elements within it correctly. Multiple attempts were tried with using additional container UI elements, such as tables, and property modifications until a working model was established that would function correctly on both desktop and mobile platforms.
Git commits
f030b0a , cf69300 , 161f26d , 189f658 , d3a7823 , 941ce8c , eb00e32 , c53d331
Relevant testing
<ul style="list-style-type: none"> • System tests 1, 3-6 (News screen visible at times) • System test 14

Table 8.

Detail of change made
New train images and trains which face in direction of travel
Reason for change made
Train images looked out of place given the the new UI changes and as such were updated to reflect the rest of the UI styling. As the new train images were longer and thinner the mechanic of trains facing either left or

right no longer seemed appropriate so the train is now rotated to face the next station it will visit. Improves system requirement 1.2 of notifying player of current game state by offering better visual information.
Affected elements
TrainActor, RoutingController, Train, All train images and cursors
Specific challenges encountered
Initially performing the calculations the way round suggested by the Javadocs for LibGDX (giving in the form (Δy , Δx)) the train would not face the correct location. Reversing these provided the correct angle of rotation. The train initially also rotated about the lower left hand corner of the actor, so the image origin had to be specified to be the centre of the actor to ensure the train was still moving along the tracks.
Git commits
<ul style="list-style-type: none"> • df68e95 • 46d8c3b • 1c792e5
Relevant testing
<ul style="list-style-type: none"> • Unit test rotationTest • System test 7 (new images are visible)

Table 9.

Detail of change made
Make the game available on Android
Reason for change made
Increase possible game audience by introducing additional platform for the game to run on.
Affected elements
AndroidLauncher, GameScreen, MainMenuScreen,
Specific challenges encountered
There were initially issues with the game rendering incorrectly on certain screens, in particular train lines and stations not aligning correctly. This was resolved by modifying stretching in the game to ensure that all elements of the game were being resized correctly when the screen size was altered.
Git commits
<ul style="list-style-type: none"> • 07e1b70 • c6da5df
Relevant testing
<ul style="list-style-type: none"> • System test 15

Appendix A

