

# Implementation report

## Essential requirements

In terms of requirements, formally, the team was only required to implement support for up to four players, instead of the previous two, and a “capture the chancellor” mode [1]. It was evidently essential to have at least these new requirements fully implemented in the final version of our product.

The engine of Team Fractal’s software from Assessment 3 already supported up to nine simultaneous players, at least one human and up to eight AI players [2]. Therefore, there were no major changes to our software when it came to the implementation of support for up to four players. In fact, the team decided, after consulting the clients, that it would be better to leave support for up to nine players and not reduce it to four. Since Team Fractal’s software supported up to nine players already, additional code refactoring to reduce that to four would have yielded extra workload, which was thought to be unnecessary. As such, the final product has support for a total of nine players, with at least one of them having to be a human player, and any desired amount of AI players.

When it comes to the “capture the chancellor” mode, a new phase named “CHANCELLOR” was added to the game [1], where the players are challenged to capture the chancellor, which randomly pops in and out of the tiles for very brief periods of time. The chancellor can be captured by clicking on the tile the chancellor is on while he is visible. The outcome of successfully capturing the chancellor is the award of fifty monetary units [1]. This design was thought to be appropriate because, as the new requirements stated, the chancellor randomly appears on the map for fifteen seconds and the player is rewarded if they manage to capture the chancellor (with monetary units rather than points). In terms of the GUI, the name of the aforementioned phase is appropriately displayed on the game screen for context, and a suitable sprite to represent the chancellor has been selected to continuously flash in random tiles. To carry out all of the aforementioned implementation of the “capture the chancellor” mode, a new *Chancellor* class was added to our software, where all relevant code can be found [3]. Said class is included in the new architectural documentation as well, where one can see in more detail how the *GameEngine* class interacts with it [4].

## Extra features

In terms of extra features, as we had worked on the same inherited software from another team in Assessment 3, some features from our previous assessment, which were judged relevant, were brought back.

Firstly, the inclusion of a more comprehensive map of the university, as described in our previous change report documentation [5], was carried out. The team decided that reusing the previously developed, more detailed map, would be a relevant addition to the product, by improving the verisimilitude of the map if compared to the actual university.

A “How to Play” menu which can be reached by clicking a button on the main menu was added too [6]. The “How to Play” menu contains the story of our game, and a link to an external user manual document, which is effectively what instructs the player how to play [7]. The team concluded that the addition of a menu containing instructions clarifying the use of our software and a short piece of story would only bring benefits to our final product. As such, the *HowToPlay* class from our preceding product was merged into our current one, and added to the architecture, primarily linked to our *MainMenu* class [3-4].

Still on the main menu, a button to take the user to a leaderboard section was implemented [3], [6]. In order to carry this out, we imported two classes from our anteceding software: *LeaderboardBackend* and *LeaderboardFrontend* [3]. As one would expect, the leaderboard contains a record of the top scoring players [6]. Like the *HowToPlay* class, we felt that there was no reason not

to implement a leaderboard feature in the final product, especially considering the triviality of carrying out such implementation, which essentially involved merging a class from our team's previous implementation [6]. The two aforementioned classes involved, like the *HowToPlay* class, were added to our architecture documentation, both linked to the *MainMenu* class [4]. *LeaderboardBackend*, however, is not directly linked to *MainMenu*, but rather to *LeaderboardFrontend* [3-4].

Regarding gameplay features, the option for the players to name themselves before starting the game was included [6], not only so that players can clearly tell each other apart (which is important as every user plays on the same machine), but also for recording purposes relative to the leaderboard classes [6]. Naming is dealt with in the *Player* class, which was extended from Team Fractal's previous version of the software [3]. During gameplay, the GUI reflects the current player's name on the screen, and this is handled by the pre-existing *GameScreen* class [3-4].

Furthermore, vanishing tooltips that display the details of a tile next to mouse cursor were implemented [5]. This is also a feature from the earlier iteration of the software which was judged to be relevant for the current iteration [5]. The tooltips provide an aesthetic interface, quickly conveying useful info regarding tiles to the current player. Thus, it was collectively agreed that this is a feature that should be reimplemented for our final version of the product. The implementation was entirely carried out by extending the *Tile* class already present in the past version of the game, using the many GUI related tools from the libGDX framework [3-4].

Lastly, for consistency, the splash screen of the software was changed to our team's logo, and the tile in the title screen was updated to reflect the last assessment [3].

## References

- [1] Gandhi Inc.. *Req4.pdf*, [gandhi-inc.me](http://gandhi-inc.me). [Online]. Available: <http://gandhi-inc.me/downloads/Req4.pdf>. [Accessed: May 03, 2017].
- [2] Team Fractal. *IMPL3.pdf*, [teamfractal.github.io](https://teamfractal.github.io). [Online]. Available: <https://teamfractal.github.io/assessment3/IMPL3.pdf>. [Accessed: May 03, 2017].
- [3] Gandhi Inc.. *AT4.pdf*, [gandhi-inc.me](http://gandhi-inc.me). [Online]. Available: <http://gandhi-inc.me/downloads/AT4.pdf>. [Accessed: May 03, 2017].
- [4] Gandhi Inc.. *FinalUMLArchitectureImage.png*, [gandhi-inc.me](http://gandhi-inc.me). [Online]. Available: <http://gandhi-inc.me/downloads/FinalUMLArchitectureImage.png>. [Accessed: May 03, 2017].
- [5] Gandhi Inc.. *Change3.pdf*, [gandhi-inc.me](http://gandhi-inc.me). [Online]. Available: <http://gandhi-inc.me/downloads/Change3.pdf>. [Accessed: May 03, 2017].
- [6] Gandhi Inc.. *Impl3.pdf*, [gandhi-inc.me](http://gandhi-inc.me). [Online]. Available: <http://gandhi-inc.me/downloads/Impl3.pdf>. [Accessed: May 03, 2017].
- [7] Gandhi Inc.. *UserManual.pdf*, [gandhi-inc.me](http://gandhi-inc.me). [Online]. Available: <http://gandhi-inc.me/downloads/UserManual.pdf>. [Accessed: May 03, 2017].