**Technical Report**

## 1. Introduction

DevFortress is a project that is developed under Scrum development process. This product is splited into two sprints, 4 weeks length each sprint. This project is developed using JAVA programming language. The purpose of this project is create a simulation game, allow users to control and manage a software company with ability to hire/fire employee, sign contract and buy item to support employee.

**2. Difficulties and solution**

As this is quite a large project, we always saw problems along the way. All of them, whether they are minor or major, provided us much experience when attempting to solve. Most significant difficulties that we encountered are:

* There are many events, which results in many sprites for the event animations. However, there is obviously no sprite collection that fit all events. Moreover, it is not guaranteed that all events have a corresponding sprite collection. Therefore, we decided to combine different sprite collections for each event. In the worse cases when no sprites found, we used sequences of static images for the event animations.
* Changes are foundation of every project, so is this one. In fact, we had to update our application model frequently to match with the backlogs. Nevertheless, it makes it difficult to test the project because the business logic may change at any time. Thus, we have no choice other than keep watching for new revision and update the tests accordingly.
* Last but not least, group working is an remarkable factor that make the development hard to manage. Particularly, we had many problems when pulling from or pushing to GitHub repositories, and in allocating workload as well. So far, the best solution that we have applied is to conduct a series of meetings and encourage communication between team members. By negotiating, our team was able to reduce the level of conflicts and make everyone happy.

**3. Tools**

**a. Netbeans**

Since this project is developing in Java, we use Netbeans IDE to edit the source code for some reasons:

- Our team members are familiar with Netbean as it’s one of the main IDEs that RMIT recommended us.

- Netbeans IDE is free and has good supports for Java developer with a large community (18 million downloads and over 800,000 participating developers - from <http://netbeans.org/about/index.html>) so we can get help easily if we have trouble while using Netbean.

- Netbeans now supports Git which we are using to develop the project. With Git version control client is already plugged into Netbeans IDE it’s very convenient and easy to use Git without downloading GitHub program for Window users.

**b. Git and Github**

We are using Git to share source code and control the flow of the project over Github.

- Git is much faster comparing to SVN.

- Github is a web-based hosting service with many functions that developers can use to control and track their project’s process such as the project’s network graph which is used to follow the project’s process as well as all the branch the team is working on.

**c. Google Drive**

- Google Drive is a great tool to share project’s resources like reports or backlogs document.

- With Google Drive team members can write and edit documents together at home.

- As a web-based office suite Google Drive only require a browser to use and free for all users who has internet access.

- With auto saving function we can work without worrying about electric problems.

**d. Skype**

- One of the best free tools for communication when work in team.

- Skype’s conference chat and voice call are great to work together at home.

**e. TeamViewer**

- TeamViewer is a nice remote control tool.

- It’s very helpful when we need to show others how the program works or help them solving their problems especially when being used along with Skype.

**4. Changed feature**

**a. Event list**

We improve the list of all event in requirement so it could fit well with our strategy pattern, that is:

+ Remove some events that have similar effect.

+ Divide list to 3 categories: easy, medium, difficult.

**b. Game start**

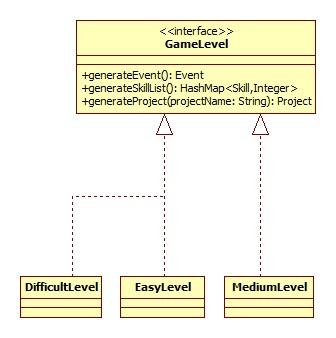
Specification state that at the beginning of the game we allow player to hire 7 employees and can train them. However, we decide that player can only hire provide trained employees.

**c. End game condition**

We only have 1 end game condition, that is company run out of money, compare to 2 end game condition as we see it is more realistic.

**5. Applicability of design patterns**

**a. Strategy pattern**

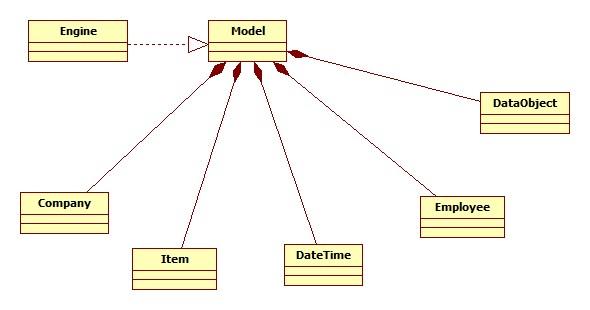


This pattern is considered as the most important pattern of our game. The purpose of this pattern is to decide the difficulty of the game through the game play. The longer player use the game, the more difficult project will be auto-generated for player to sign contract. This pattern also generate Event that will randomly occur. In difficult mode, there will be some event that can cause high damage to user.

Advantages in DevFortress project:

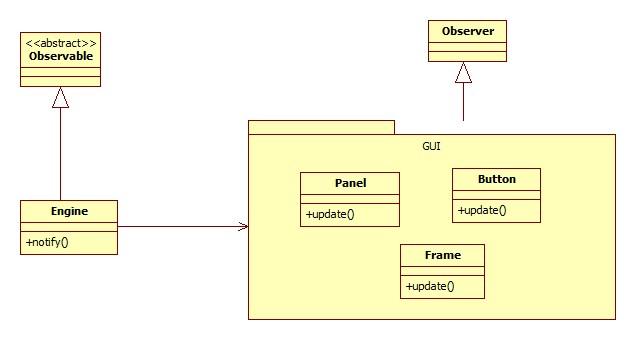
+ Allow to change behaviour of game play dung run time.

**b. Facade design pattern**



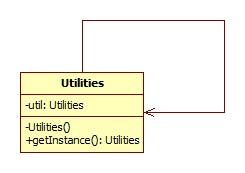
This pattern provide a simple interface for our complex system, this pattern is chosen to go with our choice of using MVC architecture design.

**c. Observer design pattern**



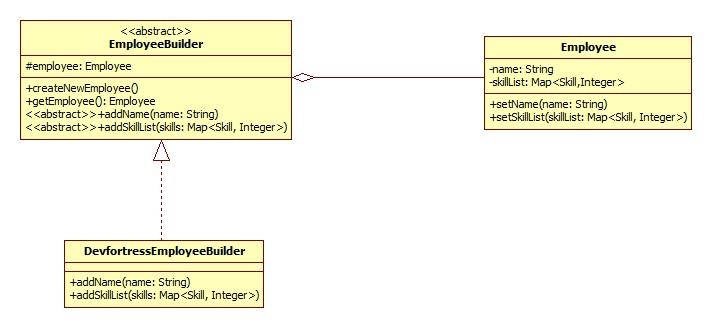
Observer design pattern allow us to communicate with the graphic user interface package without breaking coupling between classes. Moreover, this pattern also suitable for the choice of using MVC model in our project.

**e. Singleton design pattern**



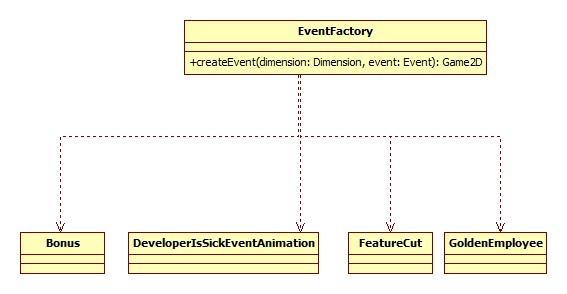
Singleton design pattern allow us to create only one instance in our project as Utilities class provides some method that could affect program data.

**f. Builder design pattern**



We use builder design pattern in our project to create employee and project object. This pattern provide us ability to create objects with ease as employee and project are every significant class in our project, and new employees and projects create every turn.

**g. Factory Method Pattern**



The Factory design pattern allow us to easily maintain a list of events, ready to be run. The Event Factory has a static method to get the instance of a pre-created event. It play an important role in de-coupling the events and the View class, allowing for later extension of event collection.

**6. Applicability of Scrum**

Our project is divided into 2 sprints, each sprint will last 4 weeks. In every week we will have a meeting to analyse what we have done in last week and what will done in next week. This meeting will be conducted in school with predetermined time. Daily meeting will be conducted through Skype

**7. Project analyse**

**a. Alternative technology**

+ Android: Android provide a good alternative to develop our project on. However, Android’s emulator quality is very bad. And some member in our team have not studied Android application development yet.

+ HTML5 + Javascript: javascript provide a great ability to interact between user and application, with the help of HTML5 canvas, this technology can provide a great interaction game play.

**b. Technology chosen**

+ Swing Framework: Swing is a very old framework to use, and it also not efficient in creating graphical user interface, However, our team have a wide experience in using swing frame work through RMIT’s assignment. Therefore we believe that using swing in our project enable us to create better UX than using other technologies.

+ Java Game2D Framework (J2DGF): This framework provides various class for creating, managing and running a 2D game in Java. J2DGF hides the complex implementation from the user, allowing us to quickly setup and display Swing animations.