Product Planning

Programming Life
Team Dynamites & Butterflies

Table of Contents

Introduction

The following is a report on how Dynamite & Butterflies will create an interactive genome graph browser. A product created by 5 engineering students of the TU Delft. In said document the product and its features will be specified, a release schedule and roadmap of said features will be made as well as the user stories that come with those features.

Product

In the following section is an explanation of the product as the goal is to make it. It involves the product backlog, a section dedicated to the features of the product. Besides that this section involves a roadmap as a general planning for the project and how the the product development is going to come by.

High-level product backlog

This section contains some high-level requirements for the product.

- The application is able to correctly parse and save data from .gfa files
- An interactive user interface allows the user to interact with the data presented
- The data that is read should be visualized and presented to the user in a graph
- The user should be able to pan and zoom through the data
- The application should keep track of the current position in the graph
- The application should make use of semantic zooming
- The user should be able to choose subsets of the genome to display
- The user should be able to annotate the presented graph
- The user should be able to hide certain information he deems unnecessary
- All of the above should scale into (very) large datasets

Roadmap

The product will be developed with Agile programming using the SCRUM methodology. In sprints of one week working products and releases will be presented. Together with a release comes a sprint retrospective, reviewing the previous weeks work. And a sprint backlog, creating a plan for the coming week to reach the next milestone.

Date	**Goals**
12-05	- Setup project with tools - Parse files - Basic UI - Some visualization
19-05	- Parse and load large datasets - Visualize graph based on centre node and radius - Move centre point query
02-06	- Layout the graph with no overlapping edges - Enable panning of graph - Enable zooming of graph
09-06	- Search/filter - Conditional colouring of nodes and edges - Edge thickness of weight attribute - Edge colour highlights specific samples
16-06	- Highlighting paths - Creating bookmarks in the graph - Highlight subsections of the graph
23-06	- Semantic zooming (mutations in bubbles etc.) - Stacking of multiple overlapping annotation paths
28-06	- Eye-candy - Bug fixes - Codebase is clean

Product backlog (first version with estimates and prioritized user stories)

In this section we will discuss the elements that should be implemented during the course of the project.

User stories of features

As a user,
When I start the application,
And I have not yet done anything,
I should be able to choose a .gfa to load into the application.

As a user,
When I load a .gfa file into the application,
I should be able to choose a centre node and radius to be displayed.

As a user,
When I am viewing the genome,
I should be able to add annotations to the graph.

As a user,
When I am viewing the genome,
I should be able to pan and zoom through the graph.

As a user,
When I am zooming in or out on the graph,
The application should make use of semantic zooming.

As a user,
When I am viewing the genome,
I should be able to gain more information on nodes and edges by pressing them.

As a user,
When I am interacting with the application,
I want the application to feel intuitive.

User stories of technical improvements (if applicable)

As a programmer,
When reading the code,
I want clear code in which I can quickly identify the functionality of different methods and variables.

As a programmer,
When wanting to contribute to the project,
I want to be able to clone the repository and create a pull request easily.
As a programmer,
When wanting to run the application,
And after cloning the master branch,
I want it to build without failures.

User stories of know-how acquisition

As a development team,
When we make a decision in a field,
And we are not too knowledgeable in that field,
We will read literature to ensure we make a knowledgeable decision.

As a designer,
When designing the user interface,
I design it so to ensure it is intuitive.

Definition of Done (backlog items, sprints, releases)

This section describes when something is done. This is to ensure all project members are on the same line when rounding out certain aspects of this project. We will define finished for the following three subjects.

- Features
- Releases
- Sprints

Feature

A feature is considered done when:

- The feature is reviewed by 2 group members who did not work on the feature
- The feature is thoroughly tested and has a test coverage of 80+%
- The code is good quality. It is well readable by the entire team (and outsiders). We ensure this by use of analysis tools such as Checkstyle and comments in the code
- The feature meets the requirements set in the user story

Releases

A release is considered done when:

- It is a successful build on Travis
- An application can be started on all 3 operating systems

Sprint

A spring is considered done when:

- A sprint backlog has been made for coming sprint
- A sprint retrospective has been made reviewing the sprint and its accomplishments
- Most (if not all) open pull requests have been reviewed and merged into the master branch

Glossary