Motivation and background

In this document

1. Background

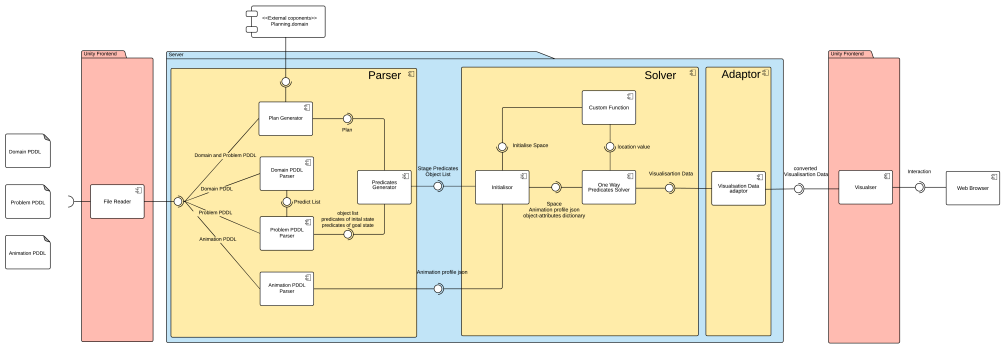
Planimation is an openSource framework to visualise sequential solutions of planning problems specified in PDDL. The framework was built by a team of University of Melbourne students in the context of the course SWEN90013 and under the lead of Professor Nir Lipovetzky and it has been continually supported by students and contributors. The project has important awards and recognitions.

The planimation goals are:

1. help to debug PDDL code for Online PDDL Editor users
2. increases user understanding of planning problems
3. showing planning solutions to non-technical audiences

The system uses a domain PDDL script, a problem PDDL script and a Animation profile script to produce visualisation.

The architecture of the system is composed of a backend server and a frontend user interface. The backend is an Apache/Python Django and the frontend is a graphic interface built with Unity graphic engine for webGL. The frontend has the task to provide user interface, pass PDDL script to the backend, display and control (interaction) visualisation and export files. Likewise the backend can parse PDDL scripts to solve the problem and produce a series of steps (or plan) in the format of a Visualisation File (VFG) that is later read by the frontend to generate the visualisation.



Planimation architecture

Key concepts:

Planning is a subtopic of the field of Artificial Intelligence, which is a model-based approach to autonomous behavior, based on finding an action sequence that produces desired state, a plan [1].

PDDL (Planning Domain Definition Language) is a computer language built to describe formal planning problems. In theory, a correctly described solvable problem can be solved efficiently by any solver. PDDL is the de-facto standard language for describing planning problems.[2]

2. Vision

Client:

The client is Dr. Nir Lipovetzky, Senior lecturer and researcher in Artificial Intelligence at University of Melbourne and Planimation curator (https://nirlipo.github.io/)

Users:

Planimation allows to user to:

1. Check errors in their PDDL code
2. Create general animation for every problem of an specific domain
3. Encode new domain animations

As identify by the client the main user profiles are:

1. Researchers in planning models
2. Students learning AI planning modelling
3. Industry partners looking for showcase solutions using planning modelling

The main goal is:

- Improve project efficiency, maintainability and extendability of Planimation software

The specific goals are:

- replace the current frontend programed in C# with the Unity engine, for a Javascript and PixiJS framework
- decrease the loading times produced by the Unity engine
- maintaining its current features
- integrate with current development tools (Online pddl Editor plugin)
- develop a new plugin for Visual Studio Code

High-level project scope:

Planimation frontend:

The main function of the frontend is to send information (PDDL scripts) to the backend, read VFG files, display and control visualisations and export them in multiple formats.

The Unity Engine is a cross-platform graphical framework that provides graphical outputs. Its main purpose is to design videogames. In the context of a planner visualiser the Unity is a heavy tool capable to manage many graphical models and shaders and translate them to webGL(http://benchung.com/unity3d-vs-three-js/). However, this procedure seems to be an overkill for the purpose of the visualiser, making it hard to maintain, change or integrate the code with other technologies. Also it generate certain inefficiencies as the application needs to run the engine first, process the files in C# and then translate it to JavaScript to run it with webGL.

Unity Engine Loading screen

On the other hand, PixiJS is a Javascript rendering openSource library that allow to create rich, interactive graphics, cross platform applications, and games without having to dive into the WebGL API or deal with browser and device compatibility (https://github.com/pixijs/pixijs). It focuses on efficiency, running complex graphics quite fast. It integrates seamlessly with HTML5 and other web technologies. Which seems to be more suitable for Planimation in the client's eyes, to achieve the project goals.

VSCode plugin

Visual Studio Code is an Integrated Development Environment (IDE) developed by Microsoft and one of the most popular developing tools [3] . The software integrate a large number of tools and can help to build code in numerous languages. It has the capacity to extender their functionalities by user coded plugins. The software has a an Extension API that allows to simplify the plugin creation.

Expected changes:

Stakeholder	Impact
Developing team	Gain experience in software development, good mark
Client	More control, maintainability and attractiviness
Supervisor	Accomplish course objectives
Users	Faster loading times, development from VSCode
Future contributors	Easy to extend and maintain the current code

---

3. Business Case

Business options

Alternatives include:

- Keep working in the current frontend, improving code transparency and documentation.
- Write frontend in plain JavaScript
- Write frontend with other JavaScript framework as threeJS

Expected benefits

Stakeholder	Financial	Non-financial
Developing team	Possibility to access better job positions	Gain experience, increase prestige as contributors
Client		Increase in Planimation prestige and increase in future contributors
Supervisor	Fulfill salary duties	Guide future software developers
Users	Save money building their own visualisation for their specific problems	Quickly develop projects visualisations
Future contributors		Easy to integrate, understand and manipulate Planimation

Timescale

The project needs to be finish by October 24. The development time is split in 2 sprints of 4 weeks each.

Cost

The main cost are the human resources and Infrastructure cost. No pay technology is budgeted.

People/hours

	Weekly hours	Number of weeks	Market reference	Total
5 Students	20	8	$38.46[4]	$30778
Supervisor	2	8	$64.10[5]	$1025

Total				$31803

Additional cost: power, hardware depreciation, internet.

Risk

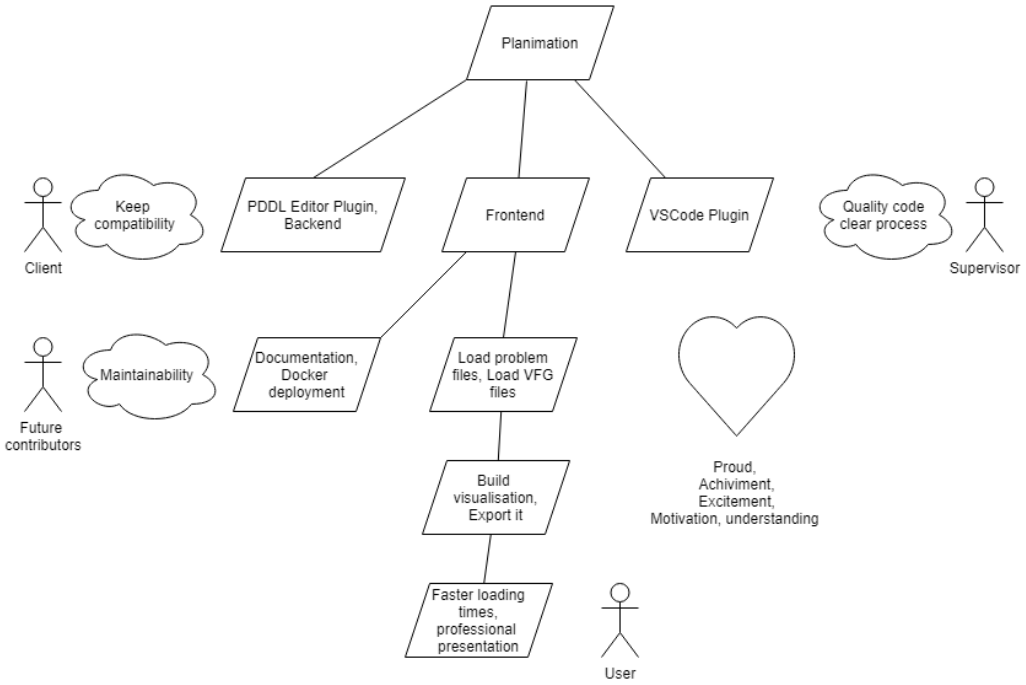
Risk ID	Risk Type	Description	Probability (0-1)	Impact (1-5)	Justification
1	Project risk	Given the comunitary nature of the framework choose by the client, the framework may have poor support, or be abandoned in favor of other frameworks in future	0.1	3	Despite beign a stable framework with 5 major releases, open source proyect comunities could become impredictable or have quite uneven support
2	Business risk	Contributors cannot extend the code provided due to design problem or poor documentation	0.3	4	
3	Project risk	Design and implementation may be inefficient due to inexperience of the development team	0.2	5	It is a fact that team members have limited experience in software development and Javascript frameworks, which may lead to some errors or inefficient code/functionality. While the probability is relatively high, the impact is low because the learning curve for the proposed technologies is low.
4	Product risk	Proposed framework PixiJS cannot deliver the desired features	0.2	5	Choosing the wrong framework might end up in some important requirements not being delivered. This has a high impact because if the risk occurs, the team might have to change the framework which would cause a big delay and effort loss
5	Business risk	The backend or OPE plugin cannot integrate to the new frontend.	0.2	5	If the code does not integrate with the current project it maybe necessary to change the code of other parts, increasing complexity
6	Project risk	The team may experience delays in the original schedule due to unexpected issues	0.8	3	The development team has no experience in project management nor in Javascript development. So it is very likely that original estimations of the schedule will not be met due to unexpected technical issues and optimistic schedules

4. Motivational Model

DO/BE/FEEL

Who	Do	Be	Feel	Concerns
Development team		Quality deliver	Achievement	Inexperienced
Client	Planimation Module Plugin, VS Code Plugin, Use PixiJS framework	Trackless	Proud	High expectations
Supervisor		Transparent process and quality of the delivered product		
Users	Load and use the Planimation module, Build visualisation from Problem files, Build visualisation from solution VFG file ,Export animation feature	Fast loading times for visualisation, professional presentation	Excited	
Furture developers	Documentation, Dockerised deployment	Interesing, easy read, inviting	Intriged, motivated, Understanding	

Model Chart



References:

1 Lipovetzky, N. (2020). Lecture 1. Short Introduction to Automated (AI) Planning [Lecture notes]. Retrieved from https://canvas.lms.unimelb.edu.au/courses/89089/pages/lecture-3?module_item_id=2215058

2. Lipovetzky, N. (2020). Lecture 3. Introduction to Planning How to Describe Arbitrary Search Problems.[Lecture notes]. Retrieved from https://canvas.lms.unimelb.edu.au/courses/89089/pages/lecture-3?module_item_id=2215058

3. https://insights.stackoverflow.com/survey/2021#section-most-popular-technologies-integrated-development-environment

4. https://au.talent.com/salary?job=junior+developer

5. https://au.talent.com/salary?job=Senior+Developer