User manual (Sprint 2)

1. System overview

In Sprint 2, we build an end-to-end program which can visualise two domains; **Blocks, and Grid** using our own constraint solver. It does not include an animation profile editor or external constraint solver. Instead, it includes a detailed user guide to the Animation Profile Language and a detailed documentation to explain the algorithms of constraint solver. It is an executable Unity architecture in a browser. It interacts with the planning.domains API to receive files (solution, domain, problem files). It takes an animation profile file from the client's local disk, and animates the solution. Any Blocks problem with under 10 blocks, and any Grid problem with under 50 objects should be fully visualisable in the program.

What was delivered:

- Working Visualiser and Visualiser File Generator for Blocks and Grid
- Flexible, extensible Animation Profile
- Flexible, extensible Visualiser
- Custom-builtin constraint solver for Blocks and Grid
- User documentation

Known issues

- Landing page not yet implemented
- Smooth motion issue
- Speed slider not yet implemented
- Help button
    - Does not work
    - Should also be present on the Select Visualiser File screen
- Ungraceful failing for unsolvable problems (no solution from planning.domains). Some of the pddl problem files in the included ZIP are unsolvable by planning.domains and therefore cannot be visualised

Updated based on Sprint 1:

- Modernised GUI
- Visualiser
    - At each step, show readable step information
    - Highlight the step in the list
- Improve syntax for Animation Profile
    - Predicate rules updated, add properties (text, sprite, scale) and custom objects (e.g claw, ground, table)
    - Animation Profile language documentation
- Domains
    - Fully functional:
        - Blocks
        - Grid

2. Getting Started

There are two ways of running the application:

1. **Use the link (Recommended)**
   Open the link: https://planning-visualise.herokuapp.com/index.html

   Use the following Test Files (Domain, Problem, and Animation Profiles): 

   . These can also be found in the bitbucket repository under Test/testfile.

2. **Test at local server (For Development)**

- Install dependencies
    - Unity
        - Install the 2017 series from https://unity3d.com/
        - When installing, select "Include WebGL"
    - Django REST
        - Follow the tutorial at http://www.django-rest-framework.org/tutorial/quickstart/
- Clone the project from https://bitbucket.cis.unimelb.edu.au:8445/projects/SWEN90013/repos/swen90013-2018-pl/browse
- Run the server with the command 'python manage.py runserver' in the directory of the file manage.py (/swen90013-2018-pl/server)
- Open the project in Unity.
    - The Unity application will communicate with the local server
- Test files are available under Test/testfile
    - These include Domain, Animation Profile, and Problem files for Grid and Blocks

3. Using the system

- Select files for the problem, domain and animation profile, then click “” button to see the visualisation.
    - If the page shows "loading", please wait a few seconds. (It should be no more than 1 minute in normal cases, if not, please refresh the page.)
    - The Animation Profile and Problem File should match the domain file
- The visualisation page has four main parts:
    - Steps Panel: Shows all the steps in the solution
    - Step Information Panel: show detailed information for each step, including actions.
    - Animation Panel: displays the animation
    - Control Panel: buttons to control the animation
- Step forward, backward:
    - Click to show next step (previous step)
- Play & Pause:
    - Play the animation automatically from start to finish
    - Or, pause to show the current step of the animation
- Replay:
    - Show the initial state of the blocks
- Play each step by clicking step button:
    - Clicking one step button, the animation panel will show the animation of that step.

4. Extending the system

Our system is modular and can be extended in multiple ways.

4.1 Extending Domains

The primary form of extension is adding new domains. This can be done (for many simple domains) with **no modification to the system**.

To add a new domain:

1. Write or obtain a Domain PDDL file for the problem domain
    a. Many domains can be found at https://bitbucket.org/planning-researchers/classical-domains/src/208a850d2ff2a27068329ad578ad99af9ec7e5c5/classical/?at=master
2. Write an Animation Profile which corresponds to the domain file
    a. Our documentation for writing Animation Profiles is a work in progress, because our Animation Profile syntax is a work in progress
    b. By Sprint 3, we hope to have a finalised Animation Profile syntax which is relatively easy to read and write.
    c. Our documentation for the Animation Profile Syntax Design (similar to the final design) can be found at Animation Profile Documentation
        i. An Animation Profile written in this format needs to be converted to JSON to be usable in the current sprint
        ii. For examples see, the attached ZIP file of Animation Profiles above.
3. Write or obtain a Problem PDDL file for a specific problem
    a. Many can be found at https://bitbucket.org/planning-researchers/classical-domains/src/208a850d2ff2a27068329ad578ad99af9ec7e5c5/classical/?at=master

With these three files, simple new domains can be visualised.

For more complicated domains or problems, modifications may need to be made to the application. These are detailed as follows:

4.2 Extending the Visualisation File Generator

The Visualisation File Generator (VFG) decides where objects are on the screen and what they look like, based on the three files listed above.

Most domain-related modifications to the system should be made to the VFG.

Extending the Visualisation File Generator requires only building the project in Django (as above). Python scripts outlined in our architectural documentation can be extended by adding or modifying new modules or functions.

4.3 Extending the Visualiser

Modifying the visualiser is only required when the current visualiser is incapable of adequately rendering the domain on-screen.

It does not concern the logic of object layout.

For extending the visualiser, see below:

Works for MacOs, Windows.

1. Setup Unity
    a. Download Unity from https://unity3d.com/get-unity/download
    b. Make sure to download the 2017 series
    c. When installing, select "include webGL"
2. Setup Bitbucket
    a. Clone the project from:
        i. https://bitbucket.cis.unimelb.edu.au:8445/projects/SWEN90013/repos/swen90013-2018-pl/browse
3. Open project in Unity