

### MV4025 Lab 3 Deliberate Attack Planning: Assault Position Selection

The primary goal of this exercise is to find good assault points for a platoon attack on a squad similar to those studied in Severson (2019) and Harder (2016), and on the same terrain. Separate points must be chosen and displayed for the main effort (ME), i.e. the squad that will be the first to move into the enemy position, and the two supporting efforts (SE1 and SE2). For the purposes of this exercise, the disposition of the enemy combatants is assumed precisely known, and unlike in Severson (2019) the axis of the ME attack will be controlled by the user. Unless extra work is done, we will not generate an attack position for blue (essentially, we are treating blue's initial position as the attack position).

0. Study the provided starter code. You may make any changes you require that fit the intent of the lab, but document them. If you detect problems with the starter code early on, please notify me, so that I consider whether I should provide a patch for the whole class.

1. Adjust the movement costs (from terrain and enemy observation), i.e. the observation penalty factor "moveObserverPenalty" and the function that determines speed as a function of grade. Also, decide how big a square is necessary to safely fit a maneuvering squad and set "unitWidth" to that value. Scan the navigation graph (leave the parameters alone), then press the "Compute Observe Data", "Average Over Unit Size", and "Recompute Costs" buttons in sequence. Then generate a cache of your navigation graph (do not allow rescanning!). I should not need to regenerate the navigation graph when I test your project, as the correct data should load from the cache.

2. Decide on the filtering and planning factors you will use to plan the main effort assault position. Anything you think appropriate and are capable of implementing can be included. You do not need to use the factors named in the starter code, discussed in class or in Severson (2019). Implement the filtering and factor computation in the code that runs when the "ME Analysis" button is pressed. For each ME planning factor, add a weight and a visualization button to the GUI. After the analysis button is pressed, pressing one of the visualization should result in the display of one colored square at each navigation node that passes the filter, appropriately colored by the value of the factor (blue is good, red is bad). When the "ME Assault Position" button is pressed, the weighted sum of the planning factors should be displayed at each filtered node, and a tactical path should be drawn from the blue starting position to the position.

3. Follow the same steps for SE1, and then for SE2. You may add separate planning factor weights for SE1 and SE2 if you find it necessary.

4. Provide a README document with the names of your group, a description of your filtering algorithm, and the names and intents of all planning factors used. Provide planning factor algorithms (in English and or pseudo-code) as needed for me to understand your work. Be sure I understand how to properly test your work, particularly if the any deviation from the simple test protocol described below is necessary, and provide an estimate of run time for any operation that takes over a few seconds. Also, please make clear which parts of the lab have been completed.

5. Zip up and submit your code (Asset and Project Settings subdirectories) plus your README and submit via Sakai.

If you have a good solution for the steps above and would like to do more, you can consider the following possibilities.

A. Generate an attack position for blue. Make sure that any consideration of enemy observation take into account the larger size of the platoon. Visualize the position and the platoon's path to the attack point.

B. Analyse the generated paths and determine the most dangerous segments of the paths. Determine the red units causing the danger. Then find a position for the light blue unit to provide supporting fire for the segment assuming that it is a direct fire weapon (like a heavy machine gun). Visualize your work.

C. Add an "Execute" button that triggers the blue units to follow the plan by moving along the pre-planned routes with appropriate timing. Be aware that when the simulation starts, some data objects may be regenerated, so you might need to store your plan in static objects, or preserve it in some other way.

## **Appendices**

### *Test Protocol*

To test your lab, I will press the buttons in the left column in sequence, starting with "ME Analysis" and ending with "SE2 Assault Position" to observe all factors and resulting plans.

### *New Interface Elements*

The terrain can now be textured (colored) in various ways.

There are now three cameras in the scene. The RTS camera is the default, but there is also a top down camera and a FPS camera for walking the battlefield. The minus, left square bracket, and apostrophe keys switch between cameras. Shift left clicking activates the FPS camera in the location clicked on.