

**Project CSci115 Spring 2020: Shortest-Path & Procedural Terrain Generation**  
**(Group project – 2 students maximum)**

**Submission deadline: Week 15 (5/5/2020)**

**Project Presentation: during the lab on 05/05/2020 (or after if lack of time)**

**Programming language: C++**

Group project: Yes (maximum 2 students)

**Learning outcomes:**

- Software development with C++
- Object Oriented Programming with C++
- Data structures: arrays, lists, matrix, graph, shortest path algorithms
- Group project

**Project definition**

- The project is a 2D application where the user select a character in the environment and a target location.
- The view is from the top, where a map is displayed. The map is represented by a matrix where each cell contains an image corresponding to a type of terrain.
- The character will move cell by cell towards to target location by using the shortest path.
- There are multiple types of characters who are present on the map:
  - the human who can just walk and not go in the water or cross the high mountains
  - the bird that can fly over mountains but that cannot go in the desert
  - the fish that must stay in the water
  - the frog that can go everywhere except on the mountains and the desert.
- Rule 1: The human, bird, fish, and frog cannot be on the same cell on the map.
- We consider the environment represented as a matrix. The environment contains different types of cells:
  - 1) high mountain, 2) middle mountain, 3) foothills, 4) plain, 5) forest, 6) desert, 7) beach, 8) ocean.
  - It means that you need 8 different images to represent these blocks which can be displayed on the map.
- Rule 2: In relation to the terrain, the character can take more time to move to one vertex to another vertex in the graph.
- Rule 3: The terrain should include special features such as tunnels in the mountains.
- The terrain should be created randomly by using a procedural function to generate the terrain with quadtrees.
- The terrain (the map) can loaded from a text file where the characters represent the different elements to be displayed on the map.
- Controls:

- Click on the character to select the character
- Click on a target to select the target, then the character will move towards the target
  - Include a pause of 500 ms after cell that is visited to show the different steps.
  - You may display change the color of each cell that should be visited to highlight the path that will be taken

Example of cost for the edges:

Row: From, Column: To

From/to	High mountain	Middle mountain	Foothills	Plain	Forest	Desert	Beach	Ocean
High Mountain	2	1	1	1	1	1	1	10
Middle Mountain	3	2	1	1	1	1	1	10
Foothills	4	3	3	2	3	2	4	4
Plain	5	4	3	2	3	2	2	2
Forest	4	4	4	2	3	2	2	5
Desert	10	5				2	2	3
Beach	10	5	3	2	3	2	2	1
Ocean	10	5	4	3	3	3	2	2

You can adjust the cost matrix so it feels right for each character.

**Example:** it costs 10 points to go from a high mountain to the ocean.

## Code constraint

Your code must not contain any **break** or **continue** inside loops.

## Provisional marking scheme:

1. The project is submitted with a readme file that explains what it contains, who has done the project, and it tells how to use the files.
2. The project compiles.
3. The project is commented (e.g., head of the functions)
4. The project runs and it is possible to test the application
5. The project contains classes.
  1. A class for the map
  2. A class Character (for the human, bird, fish, and frog)
6. Main functions
  1. Display a maze based on the values from a matrix
  2. Generate a random terrain with quadtrees
  3. Load a terrain from a file
  4. Shortest path function to move a character to a target destination
    - Implementation of a data structure for a graph
    - Algorithm for the shortest path
  5. User control to select a character and move to a destination
7. A 1-page report to explain how to use the application.
8. Effort for the creation of the map
9. Effort for the creation of textures for each element to be displayed on the map
10. Possibility to select different maps.
11. Possibility to load levels from a file.
12. Personal touch to improve the project, by adding relevant functionalities.
13. Ability to explain the project during the presentation

For the graphical representation, you can inspire yourself from games such as Wargrove, Advance Wars, Fire Emblem, Langrisser, ...

Example:



Selection of a character + selection of a target: display the path to the target.