



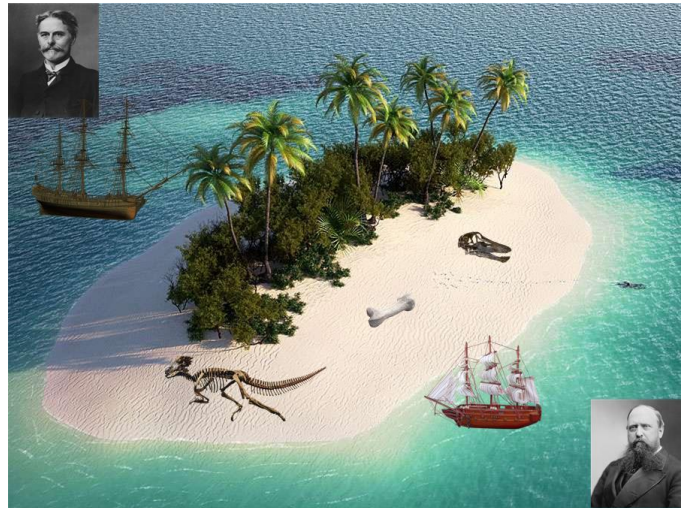
## Middle East Technical University Northern Cyprus Campus

### CNG242 Programming Language Concepts Assignment 4 (C++): The Bone Wars: An Archeological Challenge

**Deadline: 31 May 2020, Sunday 23:55**

The Bone Wars (aka Great Dinosaur Rush<sup>1</sup>) was a period famous for intense and ruthlessly competitive fossil hunting and discovery in America during the late nineteenth century. The period witnessed a particularly heated rivalry between Edward Drinker **Cope** (of the Academy of Natural Sciences of Philadelphia) and Othniel Charles **Marsh** (of the Peabody Museum of Natural History at Yale). In this assignment, we are going to focus on the implementation of a board game that relies on an **imaginary** scenario, which includes these two famous archaeologists.

Similar to many other board games we are going to have counters or pieces which are moved or placed on a specifically designed surface or "board", according to a set of rules. The players would certainly require strategic skills, however, we are also going to include the element of chance through the use of a dice. Some of the well-known strategy board games are as follows: Risk, Game of Thrones: The Board Game, Puerto Rico and Diplomacy.



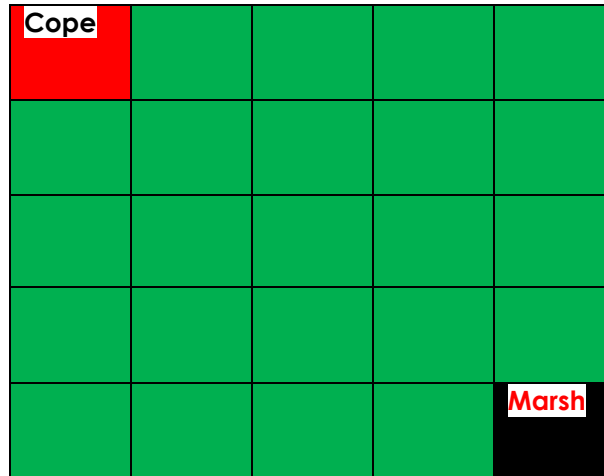
In this assignment, you will write a program to play a simplified, custom board game called **"The Bone Wars"**. In this game, we will assume that there are two players Cope, and Marsh (the computer will not play the game) that are positioned on opposite ends of an island in the ocean, as shown in the figure below. The island is a square shape platform made of  $n \times n$  elements. The archeologists strive to find all of the hidden bones. At the end of the game, the player with more bone score wins the game. **The players start the game with the same energy.** With each move, **they spend one energy and require supplements such as food and water.** Furthermore, in this initially peaceful looking island, unfortunately there are some hidden booby traps that would decrease the energy of the players. The energy of the player keeps on reducing in case they jump on a cell with a trap. **Once the player hits a trap, in addition to usual one energy spent, an additional 0.25 unit is reduced until the player finds medical supplies.** If one of the players dies (energy reaches 0) s/he can still win the game. The other player may still need to collect bones for a better score. Of course, in case one of the players collect more than half of the available bone score, the game ends with victory for this player.

---

<sup>1</sup> [https://www.wikiwand.com/en/Bone\\_Wars](https://www.wikiwand.com/en/Bone_Wars)



## Middle East Technical University Northern Cyprus Campus



The hidden elements can be one of the following:

- Bones
- Traps
- Food
- Water
- Medical supplies

The size of the board for the game will be specified by the players; however, the board is always a square shape, and the size of the board should be at least 5 x 5. For this game, the program should randomly hide  $3 \left\lfloor \frac{n^2}{25} \right\rfloor$  bones,  $2 \left\lfloor \frac{n^2}{25} \right\rfloor$  traps,  $3 \left\lfloor \frac{n^2}{25} \right\rfloor$  food,  $2 \left\lfloor \frac{n^2}{25} \right\rfloor$  water and ,

$\left\lfloor \frac{n^2}{25} \right\rfloor$  medical supply resources across the grid randomly.

The game begins by generating a random number between  $n$  and  $2n$ , inclusive which indicates the initial energy of Cope and Marsh. At each turn, the current state of the game board should be displayed as well as the total score of bones for each player.

- At the beginning of each turn, the players should choose the **x** and **y** coordinates on the board.
- In case there is any kind of resource at the selected coordinate, all the cells which involve the resource will become visible.
- At each turn, the program should also generate a random number, between 1 and 6 (inclusive). This will be our dice effect.
  - If the dice value is 6, then the 8 neighbours (can be less than 8 on the leftmost and rightmost columns and top and bottom rows. Especially in the corners, we will have 3 neighbours) will be exposed to the player and all the resources will be gathered. In case there are traps, they will not be effective.
  - If the dice value is 1, in case the player has landed on a trap, it is not going to be effective.
  - The other outcomes will not have any special effects.



## Middle East Technical University Northern Cyprus Campus

The objective of the game is to score better bone points than the other player.

On each grid, the resource types and coordinates will be assigned automatically (randomly). That is to say, in case there are 3 foods to be placed, the type (Burger, Chips, Nugets) and the location on the grid will be specified randomly. The same condition holds for types and coordinates of bones as well. The effects of each resource can be given as follows:

Resource	Effect	Size	Character to use for representing it on the board
Bone (Full Body)	+100 bone score	2	F
Bone (Skull)	+50 bone score	1	S
Bone (Ordinary)	+20 bone score	1	O
Food Burger	+4 energy	2	B
Food Nuggets	+3 energy	2	N
Food Chips	+2 energy	1	C
Water	+1 energy	1	W
Medical Supplies	Stop Trap Effect	1	M
Trap	<b>Bad news!!!</b>	1	T

After the resources have been positioned, the game proceeds in a series of rounds. In each round, each player takes a turn to announce a target cell in the grid. The computer then announces the outcome of the random numbers and the actions to be followed. The grid is then updated accordingly.

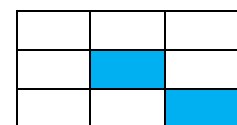
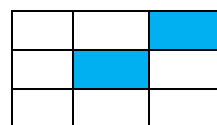
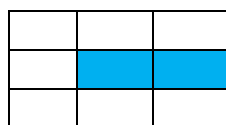
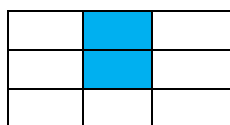
### Implementation

Create a base abstract class "**Resources**" which will have member variables for size and representing character. There are five sub-classes of resources (trap included) for five distinct types (Food and bones are further divided, but you are free in implementing these). The resources gathered by each player will be held in a structure (You can use any structure e.g. linked list, array, vector etc.). In each turn, the structure used to keep the resources will be traversed to calculate the total score, energy and current condition (is the player hurt from the trap and in need for medical attention or not). **For each three Food items in the structure, the energy can be increased one point in each round since the archaeologist with a good diet would develop a stronger immune system to survive.**

An "**Archeologist**" class will be created for each player. Each player is going to have a list of gathered resources. A "Grid" class will also be created, which will be associated with the most up to date state of the island. Each player object will have access to the same grid object.

### Deployment of Resources

The `deploy_resources` function of the grid will randomly deploy resources at the beginning of the game. The resources may be placed vertically, horizontally or diagonally as in the following examples.





## Middle East Technical University Northern Cyprus Campus

### Gathering

Each player is going to gather in turns. Before and after the choosing the coordinates, the last version of grids will be shown. The program should not allow the players to gather from the same coordinates more than once.

If one of the player collects more than half of the available bone scores, the game will end and the program will show the winner and loser with the details of the resources gathered.

### Grading Policy

The assignment will be graded as follows:

Item	Mark (out of 100)
Archeologist Class	10
Grid Class	25
Resources Class	10
Resources Subclasses	25
Game Playing	30

### Submission Rules

- You need to compress your C++ project (including header files and C++ implementation files)
- You need to submit the compressed file to ODTU-CLASS.
- Header files should contain class definitions with prototypes only, no function implementation.
- You are not allowed using `std::string` for this assignment.