

**Bicol University**

**Bicol University Polangui Campus**

**Polangui, Albay**

**Name:** Rod B. Rañola  **Subject:** Data Structures and Algorithms

**Course/ Year/ Section:** BSIS/ 2/ A  **Professor:** Prof. Khristine Botin

**Title:**

**HEX WARS**

**PROJECT OVERVIEW:**

Hex Wars is an interactive text based RPG or role playing game based and inspired from the TV series Arcane. The story of the game starts when the Noxians who are the power hungry people of Noxus discovered the creation of hextech, advanced technology that uses hex cores which are made of condensed magic. They found out that the hextech bellow the Hexgates which are gates that gives the ability of instant travel, is enough to empower every single warrior of Noxus. If successful in acquiring that technology it will eventually lead to the invasion of Noxus throughout Runetera.

In this game you play as a general organizing, leading and commanding your troops for war. Your task is to select the best soldiers you have based from power level and have them fight the enemy’s soldiers in a 5v5 match.

**KEY FEATURES:**

1. **Gameplay Elements**:

* **Army Selection**: Players choose soldiers from a pool of randomly generated Hextech-powered units with varying power levels.
* **Heap Management**: Soldiers are organized using either a Max-Heap or Min-Heap structure, simulating resource allocation and strategy.
* **Battle Simulation**: A turn-based battle system where players' soldiers compete against randomly generated enemy units. The Heap is used randomly so that the player’s and enemy’s soldiers are ordered randomly. Either Max or Min heap
* **Binary Tree Traversal**: Post-battle, players can explore their army configuration using Preorder, Inorder, and Postorder traversals.
* **Sound/Background:** Background music for having that in-depth feeling while playing.

1. **Player Actions**:

* **Soldier Management**: Add, delete, and search for specific soldiers in the army.
* **Strategic Decision-Making**: Players decide on army configurations and proceed with battle sequences.
* **Replayability**: Players can fight multiple battles with the ability to revise strategies.

**RUNNING THE CODE:**

1. Open Code::Blocks, go to **File → New → Project**, and select **Console Application**.
2. Choose **C++** as the language and create the project by following the prompts.
3. Open the main.cpp file, delete the default code, and paste the **HEX WARS** code into it.
4. Place the bg.wav file in the same folder as the .cbp project file.
5. Go to **Settings → Compiler → Linker Settings** and add winmm as a library.
6. Click **Build and Run** (gear and play icon) or press F9 to compile and play the game.

**DESCRIPTION OF EACH FUNCTION:**

**1. Display Army**

The display function prints the soldiers' power levels in a given army along with the army's name. It provides a clear view of the current state of either the player’s or the enemy's army.

**2. Heapify and Build Heap**

The heapifyMax and heapifyMin functions maintain the properties of a max-heap or min-heap by rearranging elements. The buildMaxHeap and buildMinHeap functions turn a given array into a heap to prioritize the strongest or weakest soldiers based on the gameplay.

**3. Tree Traversals**

The code includes three traversal functions: preorderTraversal, inorderTraversal, and postorderTraversal. These traverse the heap in different orders and display the nodes, which represent the soldiers' power levels.

**4. Search Soldier**

The search function checks if a soldier with a specific power level exists in the army. This functionality helps the player locate particular soldiers quickly.

**5. Delete Soldier**

The deleteSoldier function removes a soldier from the army based on their power level. After deletion, the heap is rebuilt to maintain its structure, and the player can insert a replacement.

**6. Battle Simulation**

The battle function compares the power levels of the player’s army with the enemy’s army, soldier by soldier. It announces the outcomes of each fight and determines the overall winner based on the number of victories.

**7. Typing Effect**

The typingEffect function adds dramatic flair by simulating a typing effect when displaying text. It’s used for storytelling and creating an immersive game experience.