|  |
| --- |
| **A Text/Turn Based Game** |

* Abdullah Raheel 18K-0170
* Mahad Khalid Tarar 18K-0187
* Ammar Bin Nasir 18K-1037
* Jahanzeb Raza 18K-0202
* Abdullah Muzaffar 18K-0170

|  |
| --- |
| **Objective**: To learn and implement the concepts of OOP in depth and to set our footing in the world of game development. |

# Short Description

|  |
| --- |
| Character: The user will be controlling the player. There will be different options that the user can select from, the options will be displayed along with the information about the character.  **Character** will have the following attributes:   1. Gold ( Gold dropped for enemies) 2. Level 3. Health 4. Mana 5. Stamina 6. Experience 7. Armor 8. Magic Resistance 9. Inventory (items dropped for enemies)   The **Players** will have non-combat abilities like:   1. Move    1. Forward    2. Backwards    3. Left    4. Right 2. Rest 3. Interact   The **Characters** will have combat abilities like:   1. Physical Attack 2. Cast Spell 3. Drink Potion 4. Defend 5. Ultimate 6. Escape   **Inheritance** will be applied here as there will be several classes created from character.  **Polymorphism** will be used here to deal with same name functions.  Player P;  Enemy E;  Character \*character1;  Character \*character2;  character = &P;  character = & E;  character1->setarmor(100);  character2->setarmor(23);  Using polymorphism we can only display what is necessary that way we hide our implementation so we are also using **Abstraction**  Everything will be private (armor, health, ect) and we will be using mutators and friend functions to access these variables we are using **Encapsulation** |
|  |

# Classes

|  |
| --- |
| CHARACTER {}; //Atributes defined abovePLAYERS: Protected Character {};ENEMY: Protected Character {};Example Objects:PLAYER (our Character)Mini BossFinal BossBanditsectItems {int Damage;int heal;};SHOP: Protected ITems {int value;};Example Objects:Sword (Can create life-steal by COMBINING heal/damage)Health PotionMana potionectpolyMorphism will be applied here as we will be dealing with functions of the same name |

# Other Concepts that may be used

|  |  |
| --- | --- |
| * Mutators * Inline * Destructors * Static variables/functions * Virtual functions * Over-Riding   **Display (Main Menu):**   1. New Game 2. Continue 3. Load 4. Quit   **Display (In Game Menu):**   1. Perform action 2. Inventory 3. Main Menu 4. Quit   **Display (Perform Action):**   1. Move 2. Rest 3. Interact   **Display (Combat Menu):**   1. Physical Attack 2. Cast Spell 3. Drink Potion 4. Defend 5. Ultimate 6. Escape | * Friend functions /classes * Operator Overloading * Constructers * Constant * Chain function calling |

# Case Diagram For Character:

# 

# Case Diagram For Items:

# 