DIABLO CONSOLE GAME

DIABLO CONSOLE GAME is a remake RPG game based on story from game Diablo II.



*Illustration of fighting Diablo, taken from Diablo II game*

This game allows you to develop your game character, increase level, purchase and equip item, gain gold coins and fight dozen of enemies including boss battles. Diablo is your last boss battle, if the player defeats it, the game is finished and you once again save Tristam City from obliteration.

Let the story begin…

# Game Concept

You are a *player* of a *character* in a Diablo world. This *character* has 4 **primary attributes**.

## Primary Attributes

1. Strength (**STR**)
2. Endurance (**END**)
3. Agility (**AGI**)
4. Dexterity (**DEX**)

These four attributes are *Integer*. Whenever your character gains level, three new points are given to be spent on these four attributes to make your character stronger.

## Secondary Attributes

Each of this attribute corresponds to **secondary attributes**.

1. **Damage (DMG)**: base damage of this character. Base value is 3 (without any weapon equipped). Depends only on the equipped weapon. If character equips both left and right weapon then the total damage is the sum of both weapons.
2. **Chance to Hit (CTH)**: how likely the character hits enemy. Base value is 80%. Depends on:
   * Dexterity: each Dexterity increases CTH by 2%.
3. **Evade (EVA)**: how likely the character avoids enemy attack. Base value is 0%. Depends on:
   * Agility: each Agility increases EVA by 2%.
4. **Speed (SPD)**: how fast the character moves. This increases the character likelihood to do more attack than enemy. Base value is 100. Depends on:
   * Agility: each Agility increases SPD by 5.
   * Dexterity: each Dexterity increases SPD by 3.
5. **Max Health (MHP)**: maximum health point. Base point is 0. Depends on:
   * Strength: each Strength increases Max Health by 5.
   * Endurance: each Endurance increases Max Health by 10.
6. **Max Stamina (MST)**: maximum stamina point. Base point is 0. Depends on:
   * Strength: each Strength increases Max Stamina by 3.
   * Endurance: each Endurance increases Max Stamina by 10.
   * Dexterity: each Dexterity increases Max Stamina by 2.
7. **Armor (AMR)**: armor that character has, depends only on the equipment the character currently wears, e.g. *helm, boot, shield, and armor*. Base point is 0. The higher the armor, the higher the stamina cost for doing attack (see Battle System).

All of these attributes are *Integer*. These attributes usually are modified directly by equipments the character currently wears.

## Dynamic Attributes

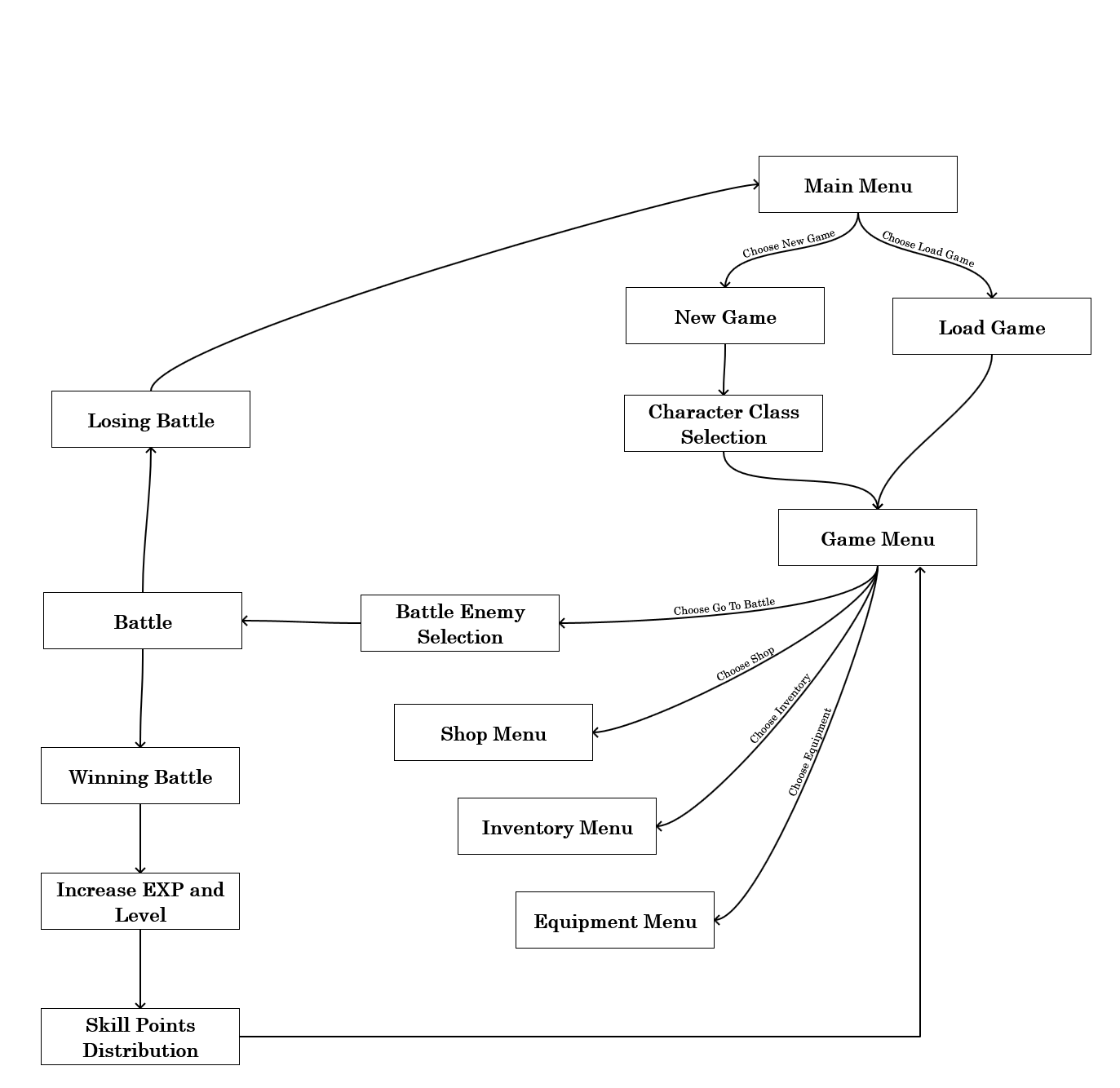
There are also **dynamic attributes** that are fast changing.

1. **Health (HP)**: current health of character (at first HP = MHP). Will automatically restored after battle to MHP.
2. **Stamina (STA)**: current stamina of character (at first STA = MST). Will automatically restored after battle to MST.
3. **Gold Coin (GC)**: currency in this game used to buy item. Initial value is 2000 G.
4. **Experience (EXP)**: current player experience. Experience starts from 0.
5. **Level (LV)**: current player level (starting from 1).

All of these attributes are *Integer*. All dynamic attributes cannot be modified by primary attributes or equipment status effect.

# Game State Concept Diagram

The flow of the game can be seen in diagram below:



*The game state flowchart*

# Main Menu Screen

There are two possibilities, when starting the application, before starting the game, it will display the following main menu:

1. New Game : start a new game
2. Load Game : load the previously saved game
3. Quit Game : exit the game

When the player already enter game then there are 2 additional options (resume & save game) when user access this main menu. The main menu screen will become:

1. New Game
2. **Resume Game** : will resume and back to game menu
3. **Save Game** : save the current game progress to file
4. Load Game
5. Quit Game

# New Game Menu

When player choose New Game, the screen changes to this New Game Menu. There are three things in New Game Screen:

1. Input player’s character unique name (validate between 6-25 chars, only alphabet or numeric character and no space allowed). Valid name for example: “vaynard89”,”bornTokillZzz339”.
2. Choose from three different classes: Assassin, Paladin, Barbarian.
3. Confirmation screen: displaying the description of the selected class.



*Illustration of selecting character class*

After player selects a class, this class attributes will be the starting attributes thorough the game. Use the following data for description of classes:

## Assassin

Fast moving and high agility make Assassin hard to hit and likely to land several small damages to enemy before enemy can move. Assassin suffers for weak protective gear and will easily go down by several devastating hits, assuming if the enemy is capable of hitting the Assassin. Because Assassin favors evading enemy hit entirely, it is hard for enemy to hit a Assassin. Assassin is likely to land critical hit, thanks to its high dexterity. Assassin will often encounter difficulty when facing enemies with heavy armor because of Assassin low damage output. This is why Assassin relies heavily on critical hits, bypassing enemy armor.

|  |  |
| --- | --- |
| Primary Attribute | Value |
| Strength | **7** |
| Endurance | **7** |
| Agility | **13** |
| Dexterity | **12** |

## Paladin

Paladin is a balanced class that is good at both offensive and defensive. It can use one-handed weapon and wear heavy armor. Its main drawback is its lack of stamina due to heavy armor penalty. Paladin often cannot survive prolonged battle unless its stamina restored. Its faith in God makes paladin has the highest strength of all classes. While relatively slower than Assassin, paladin packs both high damage and ample chance of critical hit. Its heavy armor makes enemy attack feels like gentle touch.

|  |  |
| --- | --- |
| Primary Attribute | Value |
| Strength | 14 |
| Endurance | 12 |
| Agility | 6 |
| Dexterity | 8 |

## Barbarian

Very brutal at offensive while sacrificing its protection, barbarian has slow but deadly attack. It is likely to endure the longest battle, thanks to its high stamina. Barbarian has the highest endurance and hit point and can withstand devastating hits from enemy. Barbarian is able to use all kind of weapons including two-handed weapon but what a disappointment, barbarian cannot wear heavy armor.

|  |  |
| --- | --- |
| Primary Attribute | Value |
| Strength | 13 |
| Endurance | 15 |
| Agility | 5 |
| Dexterity | 6 |

# Game Screen

Display the 4 primary attributes and all secondary attributes and gold coins.

Display 5 list of actions:

1. Go To Battle
2. Visit Tristam Shop
3. View Inventory
4. View Equipment
5. Back to Main Menu

# Inventory Management

Display all items that the character already bought.

You must filter out inventory that is being equipped by the character. For example, if a player has 3 items in inventory which one of them is a weapon and is being used in equipment in left hand then this weapon should not be displayed in this list.

When you sell your item in inventory you also get gold coin to **50% of its original price**.

# Shop Management

There is only one shop in this game you can enter. In this shop you can view the item, sort, and filter, and also buy the item you want. Gold coin is currency used in this game, it is used to buy stuffs in shop.

At first, when player choose new game, the item list is loaded from file *shop.txt*. But further when player choose load game, the item list is loaded from inside save game file itself. There are total at most 100 items in shop.

*shop.txt* has format as follows:

<item name1>,<item type1>,<item price1>,<item effect1>,<restriction1>

<item name2>,<item type2>,<item price2>,<item effect2>,<restriction2>

…

<item nameN>,<item typeN>,<item priceN>,<item effectN>,<restrictionN>

Description of the format:

* **Item name** is the name of the item. (String)
* **Item price** is the buy price of the item in shop. (integer)
* **Item type** will be one of the following: *helmet*, *weapon*, *shield*, *body armor*, *boot*.
* **Item effect** will be using the following format:

**[attribute][+/-][integer value][*space*][attribute][+/-][integer value][*space*]...**

* **[attribute]** is the primary or secondary attribute (using its abbreviation).
* **[+/-]** is the addition or reduction of the attribute.
* **[integer value]** is the amount of value this item modify.
* item effect can only has AT MOST 4 tuples. (e.g. *AGI+2 DEX+3 DMG+10 EVA+10*)
* **Item Restriction** is the restriction of this item, where:
  + 0 = usable to ALL class
  + 1 = usable only to Assassin
  + 2 = usable only to Paladin
  + 3 = usable only to Barbarian

Format example:

*Sword of Aegis,weapon,12000,STR+10 DEX-3 DMG+20,1*

This means this item has a name of *Sword of Aegis*, a *weapon* which costs *12000* Gold coin to purchase. The item has restriction value of 1, this means it is only usable by Assassin class. This item has the following effects:

* *Strength (STR) + 10*
* *Dexterity (DEX) - 3*
* *Damage (DMR) + 20*

## Shop menu

1. **Filter by: [*all,helm,weapon,shield,armor,boots*]**

If user click here, the filtering is cycled from *all* to *boots* and back to *all* again.

1. **Sort by: [*name,price*]**

*Note*: To sort by name, use case insensitive comparation. Sort ascending.

1. **Buy Item**

When player choose Buy Item, there is a confirmation which item that player want to buy.

When player has not enough gold there is a warning message “*you don’t have enough gold to purchase this item*”.

If the player purchase the item, the item in shop will be moved to inventory (you will no longer see the same item you already bought in shop). You can see the item you bought in Inventory Menu.

# Equipment

In equipment you can view and put down your equipment in these slots:

1. **Head**: *helm* only
2. **Left Arm**: *weapon* or *shield*
3. **Right Arm**: *weapon* only
4. **Hands**: *gloves* only
5. **Torso**: *armor* only
6. **Legs**: *boots* only.

Image below is the illustration of the equipment:



There are 3 options in this screen:

1. Equip Item
2. Unequip Item
3. Back

## Equip Item

You can choose category of item you want to equip. The item displayed is filtered from inventory list.

## Unequip Item

There will be list of category of item that already equipped by player that can be unequipped. The player can choose the item to be unequipped.

After unequip, all that item effect will be reversed. For example if the player buy and then wear a helmet that has STR+3 then the Strength attribute of the player is increased by 3. If the player then decide to unequip the helmet, the Strength attribute is reduced by 3 (nullify the effect of helmet).

# Load And Save

## Save Game

You are free to use any kind of format to save as long as it is kept in the same relative folder of your project and in just one file.

## Load Game

Data that needs to be loaded:

1. player character name
2. character class (e.g. assassin, paladin or barbarian)
3. 4 primary attributes
4. all secondary attributes
5. dynamic attributes
6. inventory item list
7. shop item list
8. equipment status
9. enemy list (*optional*)

# Battle System (optional)

## Monster Selection

When player choose battle menu, the screen will display all the available enemies to fight. This list of enemies is loaded from file “monster.txt” using following format:

<enemy name1>,<LV>,<GC>,<DMG>,<CTH>,<EVA>,<SPD>,<MHP>,<MST>,<AMR>,<EXP>,<OFF>,<DEF>

* Offense (OFF) is the attacking power of this monster.
* Defense (DEF) is the defensive power of this monster.
* Experience (EXP) is the experience your character will gain if this monster is defeated.
* Gold Coin (GC) is the amount of gold you receive when you defeat this.
* Level (LV) is the level of this monster.

Let N be the player level, the player can only see/fight monster up to level N+7. For example if current player level is 8 the monster list will contain monsters from level 1 to 15.

Offense and Defense are displayed using bars of character, for Offense we use ‘\*’ and for defense we use ‘#’. The number of bar depends on offense or defense divided by 20 (rounded up). For example if the Gargoyle Trap has 16 OFF and 41 DEF this means it has 1 bar of OFF and 3 bar of DEF.

After finish selecting monster to fight, the game will go to battle state where we can watch the character fights automatically thorough game.

## Battle Round

Each battle consists of several rounds and automatically loop until one side has reached zero Health (0 HP).

One round means one attack either your attack or enemy attack. To determine who moves first and who will move next:

1. Assume player speed is SPD1 and enemy speed is SPD2. First set both speed counter players (SPD\_CTR1 and SPD\_CTR2) to zero.
2. Next we add each player speed counter with corresponding player Speed (SPD). That is: SPD\_CTR1 = SPD\_CTR1 + SPD1 and SPD\_CTR2 = SPD\_CTR2 + SPD2.
3. Repeat step 2 until at least one player reach SPD\_CTR to 1000. Let that player move, if both player (you and enemy) reach 1000 prioritize the player (you).
4. After that player moves, the speed counter (SPD\_CTR) of that player is reduced by 1000.
5. Repeat step 2 until one side is dead.

## Stamina Calculation

Each time either the player or enemy get turn, it calculates the current stamina to attack, if it is below (10 + AMR), no attack can be performed, so the player must rest and gain 50 stamina. For example if a player has 12 Stamina left and wearing total 10 armor then the player needs 12 + 10 = 22 Stamina to perform an attack, therefore he needs to rest (gain 50 Stamina) this turn and thus skipping his turn.

## Evasion Calculation

Each time either the player or enemy moves, it attempts to attack, but sometimes the attack is failed (missed). When a player attack, its Chance to Hit (CTH) is tested against enemy Evade (EVA). The calculation is as follows:

1. Assume player CTH is CTH1, player EVA is EVA1 and opponent CTH is CTH2, opponent EVA is EVA2.
2. If now the player attacks opponent then we use CTH1 against EVA2:

CH = CTH1 – EVA2

If CH is < 10% then CH = 10% (minimum chance to hit)

1. If CH <= 100% then draw a random number between 1 to 100, if that random number is less than or equal to CH then the **normal attack** is dealt, else the **attack is missed**.
2. Else if CH > 100% then draw a random number between 1 to CH, if that random number is less than 100% a **normal attack** is dealt, else a **critical attack** is dealt.

## Damage Calculation

For normal damage, calculation as follows:

1. To get **Min Damage**, reduce DMG with opponent AMR.
2. To get **Max Damage**, multiply the attacker’s DMG with 100% + *STR* x 5% then reduce by opponent AMR.

*Note: if the Damage is negative (by strong opponent armor) then it is rounded to zero otherwise it is rounded up.*

For example: DMG player is 21, STR is 12, enemy armor is 7, then:

Min Damage = DMG – enemy AMR

= 21 – 7 = 14

Max Damage = DMG x (100% + STR x 5%) – Op Armor

= 21 x (100% + 12 x 5%) – 7

= 33.6 – 7

= 26.6 = 27 (rounded up)

So the player deals a random number between 14 and 27 damage to enemy.

For critical damage dealt by player, the damage is exact (no minimum or maximum damage) and damage is not reduced by enemy armor. The calculation is as follows:

For example: DMG player is 21, STR is 12, DEX is 10, enemy AMR is 7:

Final Critical Damage = DMG x (100% + STR x 5%)

= 21 x (100% + 12 x 5%)

= 21 x (160%)

= 33.6 = 34 (rounded up)

*Note: for monster, critical damage means double damage regardless of armor.*

## Winning Battle (optional)

If player manages to win the battle by defeating monster or reducing its HP to zero, all of its secondary attributes (except CTH and AMR) are increased by 10% (rounding up) and enemy level is increased by 1. CTH is increased by 4% per level and AMR is not increased. Gold and EXP are not increased. For example: monster Ghoul Level 1 has:

* DMG = 6,
* CTH = 90%,
* EVA = 2%,
* SPD = 50,
* MHP = 80,
* MST = 70
* AMR = 2
* GC = 200
* EXP = 200

Then Ghoul Level 2:

* DMG = 6 + 0.6 = 7 (+10%/level),
* CTH = 90% + 4% = 94% (+4% / level)
* EVA = 2 + 0.2 = 3% (+10%/level)
* SPD = 50 + 5 = 55 (+10%/level),
* MHP = 80 + 8 = 88 (+10%/level),
* MST = 70 + 7 = 77 (+10%/level).
* AMR = 2
* Gold = 200
* EXP = 200 + 20 = 220

The player gets amount of *gold coin* as stated in the list. NOTE: Gold coin does not increase every time player defeats this enemy. Player gets the same gold coin although the level of enemy is increased.

When a player wins a battle, some experience is gained (depends on the enemy killed). EXP to gain next level is current Level multiply by 1000. For example, to gain from level 1 to 2 a player needs 1000 EXP while from level 5 to 6, a player needs 5000 EXP.

If the player character is injured after battle, then the game will automatically deduct X gold coin to heal the character. The amount of gold needed is: 1G / HP. So if the player character has HP/MHP = 50/200 this means player automatically pays 150 G to heal. If player doesn’t have enough gold (negative G after healing), set the gold to zero.

## Losing Battle

If the character lose, the game ends.

## Gaining level

When a player gains enough EXP the player can increase level and thus can have three free points to distribute to primary attributes.

# NOtES

1. You are **allowed** to use standard library from Microsoft Visual Studio.
2. You are **not allowed** to download code from Internet.
3. Never replace or delete *shop.txt*, it is the initial value and store item list needed when someone restart or select new game. Also never replace or delete *monster.txt* as it is the original data of monster in this game.