ShaloRequirementsDocumentation.png

Document Purpose

* *Propose the entirety of instructions and details to each task and element of the game ‘Shalo’.*

*Task occupation notation*

Ryan - R

Dan - D

Chris - C

*Links to Other Shalo Guide(s)*

* [*Shalo Guide Sheet #1*](https://docs.google.com/document/d/1czzCOBjBPzIROnnSujUPH-P0WhtU5sMFDpR-PODa9hE/edit)

*High-Level To Code-Based-Level*

Each high level will link to the appropriate task identification number (TIN), searching the TIN will get you to the low level, code based description of the task

* For example “Equipment Menu (TIN: 101)”

*Each task will be appointed a difficulty labeled 1-5 on construction of the task.*

|  |  |
| --- | --- |
| Difficulty | Meaning |
| 1 | An easy 5-10min task |
| 2 | A minor task that will require 10-30min. |
| 3 | A medium level task, bugs and difficulty are expected here. These task take 30min-2hr |
| 4 | A hard task, large effort and or time will be required here, collaboration is expected for this type of tasks. This task will be heavily tested. These tasks take 2+hr |
| 5 | This level suggests that the developer has no idea what is going on and research is required as well as team collaboration. This level of difficulty should be discussed immediately. |

General Gameplay

General Gameplay pertains to: Description of game modes, Character movement, Zone transitions, Character Interaction with NPCS, Character Interaction with Shops, Character Interaction with leveling and operating eco-skills and com-skills, Character equipment handling.

Character Movement (TIN: 100)

* Consists of two modes

    i. Free roam

        Player moves around moving to and from, NPCS, Zone transitions, and hearthstones

    ii. Battle mode (Search: Battle System)

Game Modes (TIN: 401)

* Consists of three modes. Game data is stored on the local computer, and uploaded when connected.

    i. Campaign mode

        The player completes a series of quests which *do not change* each time it’s played.

        The player’s account is saved on his or her computer.

    ii. World Mode (Multiplayer / Online / Dungeon Mode)

        The player brings his or her local account to the server; once uploaded, the player can fight

        alongside other players in dungeons, skilling, and quests. These online dungeons are

        intended to be procedurally generated, however based on time constraints, may be

        static, or completely non-existent

    iii. Challenge mode (Battle / Task / Trial Mode)

        The player **does not** bring his or her own player to this game mode. Challenges (easy to

        impossibly difficult) are completed; upon completion, unique items, equipment, abilities,

        experience, or other unlockables (such as concept art, behind the scenes, or cool game

        data information) are unlocked, and or added to the selected character file saved on the

        l Inteocal computer.

Character Interaction with leveling (TIN: 203)

* This section contains the proposed ideas from the initial board meeting for Shalo

i. The **Dot System**

Players will place Stat Points, received when leveled, and place them into a statistic of        their choosing. When the dots have reached max capacity - the skill will level up and increase the actual statistic in which it represents.

ii. The **Square Grid**

Similar to FFX’s sphere grid, the square grid contains a large area for which players are

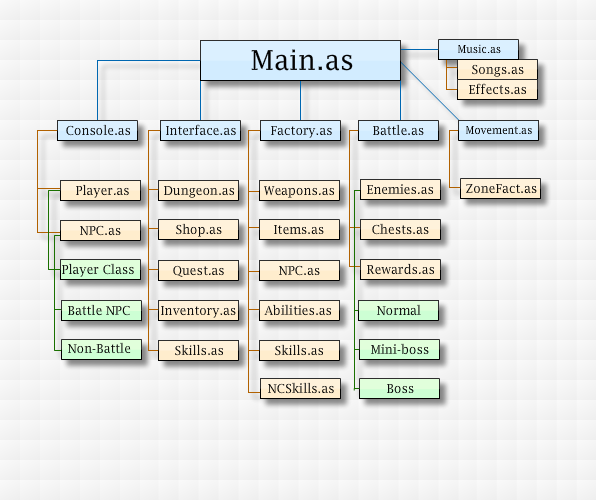
able to spend (something named similarly to) [Square Points (SP)]. Once spent, [SP] allows players to increase (not limited to): Each Skill, Eco-Skills, Abilities, Effectiveness of passives and eco-skill efficency.

iii. The **Ability Tree**

Abilities **specific to each class**(Warrior, Mage, and Ranger, [tec]), are stacked atop each

other (represented as squares connected by lines in a tree pattern). Going a specific way will cause other abilities to be ungetable.

ActionScript Package Solution Draft # 1



**Interface Design and Menus**

Interface Design and Menus pertains to: Global menu design; look and feel, a detailed explanation of each menu within the game as well as their corresponding components.

**Global Menu Design**

* Blue Based
* Sub-hover menus are blue and see through
* Simple Square based design

**Menus & Interface**

* Inventory
  + Each interface slot is stackable (amount is variable). 16 - 32 Slots total.
* Shop Menus
  + Items
* Equipment interface
  + Displayed horizontally. From left to right: [Weapon Image] [Weapon Name].
  + On Hover: Show [equipment] stats and attributes or functionality
* Quest (Help Menu / Initiate Menu)
  + Quests will be grouped by level (e.g. 1-3, 4-7, 8-10). Each quest *can* be related in story, or not. Additionally, **only a few quests from each section are required for completion of Shalo**. Completion of additional quests (which may or may not contain skill requirements) will boost the rate at which the game quests can be completed; moreover, extra-difficult quests can then be rendered available upon completion of some side-quest.
  + Story is unique to different areas. Ultimately, Ryan has proposed that the primary chain of quests **move** **from place to place.** The player then has the choice to pick up other interesting quests along the way. Special dialogue, good plot containing antagonists whose motives may or may not be unclear will be included. Intensity of stories can vary; e.g. a story can be as simple as collecting some materials for a farmer NPC, and can range up to fighting the most epic dragon in a [dungeon].
* Dialogue Interface
  + Dialogue will be similar to RuneScape. Left to right: Image of who the user is talking to, Name of NPC, [4] lines of dialogue. **Click to Continue is important in advancing dialogue**. Both, players who are skipping, and actually reading feel as though they are *turning the page* in a novel; this is the most simple form of progress and while they don’t know it will appreciate it greatly. The user is getting stuff done (even though it’s not their laundry or their taxes, it’s a similar *moving-forward* progress which makes me feel progress is happening).





Quest

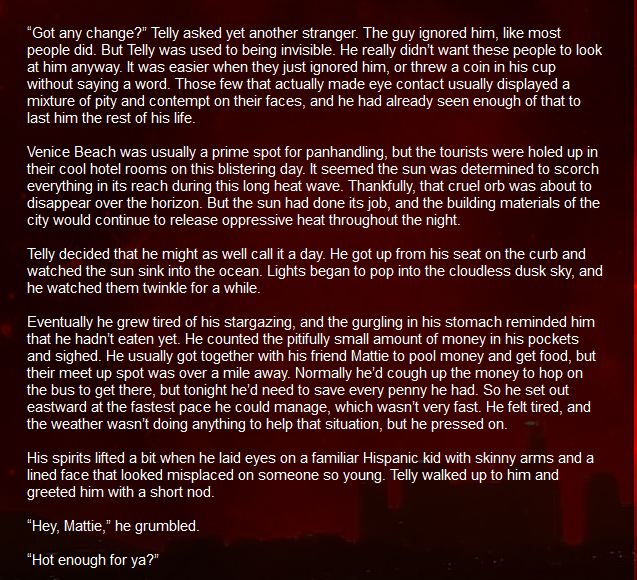
The story of Shalo can be experienced through campaign-mode. Each play-through of the game will be the exact same every time. Please experience the table, containing information regarding how quests will function, feel, and reward:

|  |
| --- |
| **- Dialogue -**  > The NPC’s which players will interact with will have unique dialogue. The main story and its sub stories (relevant to specific areas) will be fully written with intricate plot which users **can** invest his or her time in.  6_zpsfbea76e1.png |
| **- Feeling -**  Think about: RuneScape Questing / World of Warcraft Dialogue.  > When I receive a quest, I should know where to go immediately.  > As a reader of dialogue, I should be more motivated to help out with conflict, by doing x. |
| **- Reward -**  > ***Reward is extremely important.*** **The player MUST feel good about what they’ve accomplished.**   * Gold * Experience (dots) * New quests * New Areas * New Abilities?   dailyquest.jpg |

**The Writing Process with Video Games**

Writing fiction with an intricate story can be complicated. Shalo should have a story which should be pretty epic in its result. Thinking of *Mega Man X (Rockman and Forte)*, the player experiences dialogue explaining who they were in the game, and what their role was. Through improving their character, they were able to enhance effectiveness, and ultimately improve; By being able to defeat those who, at the beginning, defeated them with ease, the player will notice their improvement through the antagonists in the story acknowledging their growth in power. In Shalo, this acknowledgement of growth is important in the result of this story. The player has to have made a positive or negative impact on the lives of the NPCs involved in their own little world.

**Excerpt from Ramsey Isler’s** [**Remortal**](http://ramseyisler.com/remortal/sample.html)



Economy

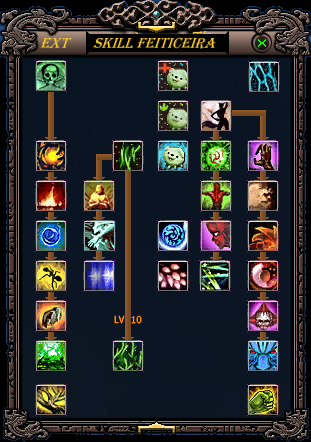
Money will run this game. Items will be bought with gold, and improvements in battling will come as a result. Upon completion of quests, gold will be one of the rewards, ultimately improving the user’s gaming experience.

**Ecosystem summary**

|  |  |  |  |
| --- | --- | --- | --- |
| **ECO-SKILLS**   * Mining * Orb Making * Book Making * Fishing * Cooking’ * Smithing * Crafting * Woodcutting * Woodworking | **SHOPS**  > Shop items refill over-time (online and offline).  > Currency: gold coins | **Why Skill?**  > Skilling is an important part of the game. Major influences for skilling include:   * Quest eco-skill requirements. * Unique ways to receive materials for crafting / making weapons. * Skill Mastery. * Game Mastery. * It’s fun, yo. | **Why Shop?**  > Buying items is important for improving the user’s character. When the player equips a new item, he or she will feel empowered and **be more capable** to complete higher level dungeons.  > Using items as materials. Materials are required to make weapons, potions, or other things. These will **benefit the player in battle and money making.** |

**Eco-Skills**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Skill Name** | **Skill Description** | **How is it done?** | **What do I get?** | **Discussion** |
| **Mining** | Mine ore to obtain minerals | Go to ore, mine for time, get material/xp. | Ores for Smithing. | Can/should ores be used for other things? |
| **Orb Making** | Create Orbs | Go to shrine. Use time, get materials/xp | Orbs for Books. | Where do we get the orbs from? |
| **Book Making** | Use orbs and materials | Combine orbs, get books. | Books used for spells. | How do players use these? Do they expire? |
| **Fishing** | Use fishing rods to obtain a variety of fish | Go to pool, use time, get materials/xp | Fish for Cooking. | How long does fishing take? What are good alternatives to these time skills? |
| **Cooking** | Cook raw food and create foods based on recipes | Random chance displayed to user near fire. Chance increased for each level. | Food used for battle. Unique colored fire which can be used as AoE totems. | Why are we cooking? What can we gain aside from better heals which most characters aside from tanks won’t be able to use? Temp Buffs? |
| **Smithing** | Use ores and other materials to create weapons and armor | Combine ores, make bars, make equipment. | Equipment used for battling. | How do the equipment tiers differ? How is equipment unique in its build pattern. |
| **Crafting** | Combine materials to create various goods | Use a plethora of materials to create and fuse raw and non-raw materials. | Weapons, equipment and other items used for battle. | What the fuck is crafting for? is this enchanting? if we have wood working and smithing, why are we crafting? |
| **Woodcutting** | Cut trees to obtain a variety of wood materials | Travel to type of wood, use time to get materials/xp | Wood used for woodworking | Where do we find the trees? |
| **Woodworking** | Use wood and other materials to create bows, staffs and other items | Use wood from inventory to make wood items, such as bows, crossbows, etc. | Weapons used for equipment and weapons. | How can we implement a system that is different from, cooking, crafting, fishing, oring |
| **Farming (Opt.)** | Use seeds to grow herbs over (longer, > 10mins) time. | Use herbs and other materials for potions, and special consumables. | Consumables used for battle buffs, dungeon buffs, and xp gains. | What else can farming be useful for  for the user? |



Battle system

Shalo’s fundamental battle system is similar to that of Final Fantasy VIII. Fighting will not be turn based, however will have elements of a fast pace system with the perks of the ABT (Active Battle Time) bar.

|  |  |  |
| --- | --- | --- |
| **- Movement -**  > ABT (Active Battle Time) bar available for player and each NPC. The bar fills over time, and movement, techniques, and other things will consume **subdivisions** of this bar. | **- Using Abilities -**  **>** Into each dungeon, the player (in preparation) **can bring only up to [6] abilities** from his or her pool of learned abilities and spells. This will boost the strategy factor when preparing for known dungeons. Abilities will apply the classes resource constraints (Mana, Rage, Stacks) | **- Health -**  > Health is little. The way it was original (as best as I can remember) is that HP began at around 3 or 4 bars. Damage was calculated as [1.00]. I believe that **we can increase that to 10**, allowing for some flexibility in the early levels. |
| **- Chest -**  > Dealing with chests [involves locksmith skill?]. Each dungeon has chests which, upon opening, **give the player items** and materials for other things. These chests may be surrounded by difficult NPCs, and can be avoided if the primary goal is to just complete the dungeon. | **- Dungeon Drop Tables (TIN : 101) -**  **>** Monsters, bosses and chests have pre-set drop-tables. Upon killing a monster, the loot can be roll for and a player will or will not receive some pretty neat and possibly **very rare pick-ups.** | - **Location of NPCs -**  **>** Subject to change, NPCs will be **in the same place every time the instance of the dungeon is ran**. This is helpful for rerunning the dungeons; players will know exactly when to use AoE Moves and heal when a large mob is after them. |
| **- Ability Tree -**  > Each time a character levels up, they will receive a sphere point. The sphere grid is where character specialization occurs and where sphere points are spent. For instance, a warrior could have the option between spending a point further into his ‘Offensive’ tree branch in which he already has points or to balance himself out to increase his the ‘defence’ tree branch making him tanky as well has offensive. ‘Pures’ however will choice to only increase points in a single tree branch. |  |  |

**Combat Class Design**

Classes are chosen at the beginning of the game when you select a new character

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Class Name** | **Description** | **Strengths** | **Weaknesses** | **Other Column** | **Ease of Play** |
| Fighter | Physical dmg dealer. Moderate HP/dmg/AoE. | Robust.  Reliable damage  Sustainable. | Lacks burst. Predictable. Easily kited/zoned. | Good at specializing; AoE, single target, HP, stuns | Easy to  Moderate |
| Rogue | Mobile, physical dmg dealer. Low HP, high damage, single target. | Mobility.  Burst.  Resets.  Mixed damage. | No healing sources.  Little CC  Low defense. | Needs to be able to be outplayed, yet not completely defenseless. | Moderate to difficult |
| Mage | Ranged magic dmg. Diverse abilities. | Ranged, high utility, AoE and single target. Heals. | High cooldowns. Low defense. Slow ATB/MS | Targets need to be able to take advntge of big CD’s | Moderate |
| Ranger | Ranged utility, physical dmg  Set-ups | Ranged crits.  Zone control. utility. | Auto attack reliant. | Needs weakness besides “low HP/resist” | Easy to difficult |

FIGHTER

* Guardian - Tank. AoE focused. Sustain lies in regeneration in healing.
* Decimator - Single target berserker. Offensive capabilities increase as HP decreases. Little to no CC or utility. Sustain lies in lifesteal/vampirism.
* [R] Voidblade - Buffs self by sapping target stats. Dot+Hot necrotise.

ROGUE

* Assassasin - stealth + burst. low hp. single target focused. “resets”
* Renegade - Blink/snap to enemies. contagious DoT. increased dmg if target is alone/bleeding/lowHp, increased MS/ATB while he causes a bleed

MAGE

* [R] Venomancer - strategic AoE debuffs and DoTs with vampiric mechanics. great team synergy
* Cleric - Healer. buffer. HoTs
* Archwizard - Combo focused. all spells make each other stronger in different ways; different combinations yield varied utility to self/team

RANGER

* Sharpshooter - Deals higher damage at max range. less damage as target comes closer.
* Hunter - Set-up based. Traps, totems, buffs, etc.

****

**Combat Statistics (TIN : 103)**

Global | Class Specific

|  |  |
| --- | --- |
| **Statistic** | **Purpose** |
| Hitpoints (HP) | Player life, when 0 player dies |
| Tempo (TP) | Rate at which ABT refills |
| Resistance (Res) | Reduce a percentage of total damage taken |
| Dodge (Dog) | Percent chance to completely dodge a physical attack |
| Magic (Mag) | Increases damage of magical attacks |
| Strength (Str) | Increases damage of physical melee attacks |
| Movement Speed (MS) | The number of units one can move per ABT bar |
| Critical (CRIT) | The chance that an ability will deal 1.5-2.0 |
| Manapoints (MP) | Is consumed when player uses a special ability |
| Rage (RG) | Increases when dealing damage, consumed to use special abilities |
| Stacks (SK) | Every time an ability is landed, increases. Consumed when certain abilities are used |

i. The **Dot System**

Each skill (Attack, Strength, Defence, Magic, Range, Hitpoints, etc) has a integer

representing how many dots (%) away from completion of that level for that skill.

(e.g. - - - / 5 would represent 3 / 5 or 60% completion for this level)

ii. The **Square Grid**

Similar to FFX’s sphere grid, the square grid contains a large area for which players are

able to spend (something named similarly to) [Square Points (SP)]. Once spent, [SP] allows players to increase (not limited to): Each Skill, Eco-Skills, Abilities, Effectiveness of passives and eco-skill efficency.

iii. The **Ability Tree**

Abilities **specific to each class**(Warrior, Mage, and Ranger, [tec]), are stacked atop each

other (represented as squares connected by lines in a tree pattern). Going a specific way will cause other abilities to be ungetable

**How it was?**

I think it is very important to include how dungeons functioned on paper in order to better realize what it will play like in game. I remember being assigned the task of writing the level 8 dungeon, titled the “Cave of Terrors.” I was in charge of making the *original copy* of this dungeon, since we were doing a *world map* expansion, I was sure to make my lines, monsters, and loot text very clear. I began with the title, and grid. A total of three pieces of paper were used for this map. The design was relatively intricate, as the other two pages were used for the upstairs portion, and the downstairs portion of the map. In the downstairs portion were many chests, guarded by a strategically place amount of mages, rangers, and close-range type monsters. Only a crew of 2 or more players would be able to handle this room. The benefit was the loot within the chest was worth it. The upstairs area contained the mini-boss for this advanced level. I know that experience was *divided* between players. Making sure that I added the necessary drop % for each monster, its strength levels, hp, and weaknesses, I added around 5 new NPCs unique to this specific dungeon. I knew this would be difficult for us. Afterward, I believe I added the loot drop tables for chests, and the final boss as well as how it worked (and how it would be uniquely difficult). Ultimately, making the dungeon instances was a timely process, as it took a lot of creative aspect; but I distinctly remember completing the dungeon time after time was well worth it and made me feel my farming was put to good use.

Server

\*\*Note: The Development of the server will come *after* the development of the Campaign mode. A full game will be designed before the AS3 Code is then translated into Java, and socket functionality.

The server will be written in Java. The base of the server will be made up of the code best fit for TCP (call-receive) type interaction between client and server.

> **Dungeon:** Each dungeon requires a separate instance on the server. This is where the actual code which handles battle, location, items, gold, abilities, HP and MANA, and real variables for **each** player.

    For each dungeon, because it is turned based, and a grid system; sending the coordinates of each

    player will not be difficult.

> **Shop:** Items are global, the **server** will contain items for each shop and refill over time (or refill full upon server reset).

> **Out-battle:** Up for discussion. The original way movement worked outside of dungeons was a free roam style. New ideas include, but are not limited to:

* Grid system within the world map.
* ‘Zone Class’ which showed people who were only in your specific zone (as opposed to having one huge world map, with all players on the server (not in a dungeon) loaded into it).

> **PvP (Player-Versus-Player):** Once developed, the battlesystem for PvE will be developed for players on players.

**Animation & Artwork**

Animation and artwork is important in guaranteeing that the player will enjoy the great features of the gameplay. Having a certain animation style effectively increases uniqueness to a game.

Models

    Hero/Enemy models

* Warrior
  + Standing
  + Moving
  + Basic Attack
  + Ability 1
  + Ability 2
  + Ability 3
  + Ability 4
* Ranger
  + Standing
  + Moving
  + Basic Attack
  + Ability 1
  + Ability 2
  + Ability 3
  + Ability 4
* Wizard
  + Standing
  + Moving
  + Basic Attack
  + Ability 1
  + Ability 2
  + Ability 3
  + Ability 4
* Rat
  + Standing
  + Moving
  + Basic Attack
* Goblin
  + Standing
  + Moving
  + Basic Attack

**Classes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | **Location** | **Type** | **Purpose** |
| Main | Built in | Static | Initialize game |
| ZoneFactory | Global/Static | Static | Return the Zone objects |
| Zone | Base Class only | Single | Base class, contains array of all visual tiles and all npc, and characters |
| RoamZone extends Zone | Driver.as | Single | Represents the current x by x mapping of all characters and visuals in ‘Free roam mode’ |
| DungeonZone extends Zone | Driver.as | Single | Represents the current x by x mapping of all characters and visuals in ‘dungeon mode’ |
| Movable | Anything that will move | Many | This class facilitates and controls movement in all classes |
| Colidable | Any class that requires hit detection will have the class | Many | This class computes whether an object is or isn’t collidable |
| DropTableFactory | Global/Static | Static | When a dungeon is entered this object is called to retrieve the drop table that corresponds to the dungeon |
| DropTable | DuegonZone | Single | Contains the list of possible dropped items and dropped items that will be distributed at the end of the dungeon |
| Item | Anywhere loot is required - shops, drops, hero, npcs, quest | many | Base class for all items |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

TASKS

A programmatic description of each task

---------------------------------------------------------

**Character Movement: Free Roam** (TIN : 100) (Diff : 4)

    Classes

* ZoneFactory: Static class that returns a new RoamZone.as Class
  + Call ZoneFactory.zone([currentZoneNumber:Number, directionNumber:Number]) and a new RoamZone.as object will be returned. If you are just logging in call ZoneFactory.zone([currentZoneNumber:Number]) and the new RoamZone.as object will be returned.
* RoamZone: Static Class (and not Static when returned by ZoneFactory.zone() to be absorbed by Static Zone) on Main or Driver class that will contain the current order of the two dimensional array of characters as well as the two dimensional array of visuals.
  + Call roamZone.move([from:Object[x:Number, y:Number], to:Object[x:Number,y:Number]]) move will ask if moving from and to here is possible if hit collisions exists false is returned otherwise true and the character can move.
* Movable : A delegate class that is contained within a class that should be able to  move. Controls player speed and frozeness. If NPC or enemy this controls movement and character triggers character moving animation.
* Collidable : A delegate class that allows for a class to be collidable or not based on certain details.

    Notes

* Players, NPCs, treasures, enemies and non-static elements all **exists on a x by x two dimensional *charLocation:Array*** within the Zone Class.
* When a player or NPC or enemy moves the *charLocation:Array* current state will be updated.
* Visuals, entrances, exits, and static elements **exists on a x by x two dimensional *visuLocation:Array*** within the Zone class
* The *visuLocation:Array* remains mostly static except for changeable environments (example: a secret wall in a cave, or a switch)
* When a character walks to a door or to the end of the map we will clear the map, call ZoneFactory.zone(...) and the new Zone will be loaded and each element will be populated.
* Standing next to an npc or other character based object will cause collision detection

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

-------------------------------------------------------------------------------------

**Dungeon Drop Tables** (TIN : 101) (Diff : 3)

Classes

* DungeonZone.as extends Zone.as : Class on main.as or driver.as that represents the current Dungeon and all of its contents (enemies, heros, chests, animations)
* DropTableFactory.as : Static class that returns a drop table object based on the parameters. Call DropTableFactory.DropTable(ZoneIdentificationNumber:Number) to recieve the DropTable.as class to be insereted into the DungueonZone.as object.
* DropTable.as class contains an array list of all possible item:Item.as for this dungeon and another array *dropChance:Array* which contains the chance (summing to 100) to drop each item. Call dropTable.roll() to load the item in the *loadedItems:array* which at the end of the duengon will be distributed and will return the tier level of the chest by calling dropTable.unload() and the list of Item.as will be given
* Item.as class is the base class for all items.

    Notes

* Each ‘Dungeon Zone’ class will, upon initiation, create and hold it’s DropTable.as object. When an enemy dies or when a chest is dropped and inserting into a ‘Loot bag’ where the total types of loot is displayed, hovering over this loot bag will display the other loot tiers and the division of the total loot among the each loot tier
* The tier of each item (Item.as) is held in itself

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

-------------------------------------------------------------------------------------

**Combat statistics** (TIN : 103) (Diff : 4)

Classes

* BaseStats.as : contains a dictionary/object of Statistic.as class that represents a characters raw/base/flat statistics
* Statistic.as : contains exp, level, gain per level:function, and stat actual data for any given statistic
* AdditiveStats.as : contains an array of Additives.as class that represents all possible other statistic increase to the base statistics
* AdditiveStat.as : contains object of {elementKey (what stats are included here 00010 could be only mana),stat : statAmount, stat : statAmount, ...}
* EnhancedStats.as : contains an object array of the summation of the base stats and additive stats, also has current hp, resource, abt, etc

    Notes

* Upon creation base stats will be set, additive stats will be set, and enhanced stats will be set to the summation of the previous two (additive stats and base stats)
* When armor, buff, weapon, passive, anything is equipped/removed the following occurs…

1. it is added/removed to the additiveStats.additive array

        2. the bonuses are added/subtracted from enhancedStats.statSummation array

* All percentage boosts from passives and anything must be based on BASESTATS only, because it is static.
* When leveling up base stats, all passive bonuses must be reconfigured
* Upon creation of a new hero or enemy, their Resource bass class will be set to a child class that represents their classes required resource (mage -> mana, warrior -> rage, ranger -> stacks)
* Every Statistic.as class contains a gainPerLevel:Function, and a statisticAmount variable, this is the true base amount, the level is seperate.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_