**CORE MODULE INFO**

RESEARCH SYSTEM

* Most recipes in Ancient Warfare require research before they are craftable. The entire research system may be disabled in the core config file – in which case, no research will be required and all Ancient Warfare recipes will be craftable in the normal workbench.
* Research is done at the Research Station, which may be crafted in the normal workbench.
* Research requires a Research Book, which is also craftable in the normal workbench.
* Research is stored per-player. Any player may contribute to any others’ research by obtaining a copy of their research book and using that in their research tables.
* You may use other players research for crafting by obtaining a copy of their research book and using it in your Engineering Station.
* You may only progress on a single research goal at a time. All research stations with a copy of your research book will all contribute towards your selected goal.
* To begin research first craft yourself a Research Book. Right click the book once in order to assign it to your player. From now on whenever that book is placed into a research station, that research station will reference your research progress.
* Research goals have both a time and materials component. The needed materials may be viewed on the tooltip of the goal in the research-queue gui, or in the ‘needed’ box on the main research progress gui. Once these materials are present in the ‘input’ box, the materials will be used and research will automatically start progressing.
* In order to make progress on a research goal you must do one of three things:
  + Leave the research station GUI open.
  + Assign an npc researcher to work at the station.
  + Right click on the research station with a ‘Quill’ item equipped.
* Multiple Research Stations may be setup to make faster progress towards a single goal. They will each need researcher NPCs assigned to them, or have players using quills / viewing their GUIs.
* The Research Book may be right-clicked after it is assigned to view a GUI with details about each research goal and researchable recipe.

CRAFTING SYSTEM

* Most Crafting in Ancient Warfare is accomplished at the Engineering Station.
* The Engineering Station may be crafted in the normal workbench.
* The Engineering Station can craft any vanilla recipe in addition to researched recipes.
* The Engineering Station has a slot in the upper-left to place a Research Book. The research from the books’ owner will be used to determine what is craftable at that Engineering Station.
* If no book is inserted, the Engineering Station will only be able to craft vanilla recipes or the few Ancient Warfare recipes that do not require research.
* There are 18 resource storage slots in the Engineering Station. When crafting, items will automatically be pulled from these slots to refill any items in the input-matrix that would have run out.

RECIPES

* To view recipes, use NEI.
* There is a built-in NEI plugin that allows viewing of all researched recipes – it should list the crafting type as “AWCrafing” or similar. They do not yet list the required research in the NEI recipe lookup, but that capability should be added in the near future.

TEAMS

* Most Entities and tiles in Ancient Warfare use the vanilla scoreboard teams system to determine a players’ team status.
* Team status may effect your ability to control or interact with an entity or block. If you cannot interact with something – check to see if the owner can. If you are the owner and cannot interact with something you own, please file a bug-report.

OPTIONS MENU

* The options menu may be accessed via the F7 key (configurable via the Minecraft options->keybinds config gui).
* Any client-side options that are adjustable in-game will be listed and adjustable here. These options are client-side only, and have no effect on the server or other clients.
* There is an additiona control to access the Keybinds configuration GUI.
  + Custom Ancient Warfare keybinds may be reconfigured in this screen.
  + These keybinds may share keys with vanilla keybinds or other mods keybinds – it is up to you the user to resolve in-game conflicts. I felt it was more compatible in the long run to allow for sharing of keybinds and duplicate assignments.

CONFIG FILES

* Each module has its own config file for module-specific options. The core module also has its own config file for global/shared options.
* Please pay attention to the default values and comments in the config file, these should explain everything necessary to make adjustments.

ITEM USE

* All clickable items have been reworked to use keys for most of their functionality. This resolves issues of not being able to click the item on a block that also has interactive properties.
* The default key for item-interaction is : Z (can be changed through the options menu). This is used in many items that previously used left/right click functionality to select blocks.
* Other keys available for item actions are: X, C, V, B – these are used on some of the more complex items such as the NPC command baton, construction tool, and structure scanner. See the item tooltip for directions as to what key does what function.

**AUTOMATION MODULE INFO**

WORKSITES

* Worksites form the heart of the Ancient Warfare automation system.
* Each worksite is devoted to a single type of work. There are worksites for the following types of resource-related tasks:
  + Crop Farm – Plants and harvests wheat, carrot, potatoes, melons and pumpkins. Blocks to be tilled/planted/harvested are configurable via the control GUI.
  + Tree Farm – Plants and harvests vanilla saplings. Should also work with any mod-added saplings that properly extend ItemSapling. Blocks to be planted/harvested are configurable via the control GUI.
  + Mushroom Farm – Plants and harvests red and brown mushrooms as well as netherwart. Appropriate planting conditions must be met – e.g. low light and dirt for red/brown mushrooms, low light and soul-sand for nether wart. Blocks to be planted/harvested are configurable via the control GUI.
  + Reed/Cactus/Cocoa Farm – Plants and harvests the ‘odd’ vanilla plantables. Appropriate planting and growing conditions must be met – cocoa must have jungle wood to grow on, cactus must have sand and clear area, reeds must have either sand or dirt adjacent to water.
  + Quarry – The traditional pit-style quarry. Dig a giant hole in the world.
* In order to place a worksite, you must first denote what area it will be working. This is done by equipping the worksite block and pressing the ‘Z’ key on the first block position and then again on the second block position. Position may be offset by side hit by sneaking/holding shift (this allows for selecting air blocks). The work bounds will be rendered on screen while holding the worksite item. After both bounds have been set, you may place the block through right-click. A worksite control block must be placed directly adjacent to its work bounds (may be placed diagonally in either or both horizontal/diagonal directions, but it must touch the work-bounds with at least one corner).
* Worksites have at least one inventory area – the output. This is at the top of the GUI.
* Some worksites have a second inventory area. This is the Input area. Any resources needed by the worksite should be inserted here.
* A few worksites have a third inventory area. If present, this area is used for special resources – notably bonemeal for crop and tree farms.
* Inventory side accessibility is adjustable for all worksites – click the Inv. Setup. button in their GUI to access the side-control GUI.
* Players may manually ‘work’ a worksite by right-clicking on the block with a ‘Hammer’ equipped that is in ‘Work’ mode.

AUTO CRAFTING

* The auto-crafting station is a special block dedicated to automating the crafting of items.
* It has a research book slot to enable the crafting of AW researched recipes, upper-left corner.
* It has an input/layout grid to determine what recipe to craft.
* It also has a resource input area to store the resources needed to craft the denoted item.
* Finally, it has a resource area for the crafted output.
* Inventory side accessibility is adjustable – click the Inv. Setup button in the GUI to access the side-control GUI.
* Crafting may be activated by pressing the ‘Craft’ button in the GUI, or right-clicking on it with a ‘Hammer’ equipped that is in ‘Work’ mode.
* The auto-crafting station is also a worksite – you may assign a ‘Craftsman’ type worker to work at it, and he will craft an item on every work tick.

WAREHOUSE

* The warehouse is a special work-site dedicated to storing items. Lots of items. Any items, all items.
* It needs its work-bounds set in the same fashion as the Crop Farm, Quarry, etc. Work bounds may be up to 16w x 16l x 4h.
* The warehouse is a multi-block structure that requires more than just the controller block to function. The additional blocks needed are: at least one Warehouse Storage block (any size), and at least one Warehouse Interface.
* The warehouse bounds may be filled with any number and type of Warehouse Storage block or Warehouse Interface block.
* Warehouse Storage blocks add additional storage space to the warehouse. Inventory is stored –in- these blocks. Removal or destruction of these blocks will cause the ejection or destruction of any contained items. Each storage block may be right-clicked to set filters to determine what items may should be stored in that block. An empty filter list denotes that any item may be stored there. A filter with an empty item accepts nothing.
* Warehouse Interface blocks allow for interfacing NPCs and other mods’ with the warehouse. Each interface block has a configurable filter list in their GUI. For each item in this filter list the warehouse controller block will attempt to keep the interface stocked with the specified amount of the specified item. Any items not matching any filters in the interface block will be marked for moving into a storage block. Thus the interface block functions as both an input and output block for the warehouse.
* The warehouse controller block may itself be interacted with to display the warehouse GUI. This GUI displays all items stored in the storage blocks controlled by this warehouse controller.

MAILBOX

CHUNKLOADERS

TORQUE POWER SYSTEM

**NPC MODULE INFO**

INTERACTION

* A Player Owned NPCs inventory may be accessed via right-clicking on the NPC.
* A Player Owned NPC may be commanded to start or stop following a player by sneak-right-clicking on the NPC.

TOWN HALL

C:\git_source\git\AncientWarfare2\src\main\resources\assets\ancientwarfare\textures\blocks\npc\town_hall_side.png

* Functions as an auto-assigned upkeep point for same team / owner npcs.
  + If a same-teamed npc is within range and does not have an upkeep point assigned, they will auto-assign the town-hall as their upkeep point.
* Has an inventory for storage of food for npc upkeep.
* Tracks deaths of nearby npcs, player owned priests may resurrect these npcs

COMMAND BATONS

C:\git_source\git\AncientWarfare2\src\main\resources\assets\ancientwarfare\textures\items\npc\stone_command_baton.pngC:\git_source\git\AncientWarfare2\src\main\resources\assets\ancientwarfare\textures\items\npc\iron_command_baton.pngC:\git_source\git\AncientWarfare2\src\main\resources\assets\ancientwarfare\textures\items\npc\gold_command_baton.pngC:\git_source\git\AncientWarfare2\src\main\resources\assets\ancientwarfare\textures\items\npc\diamond_command_baton.png



* functions as a sword, with no block ability
* right click to add or remove an npc from the commanded npc list, which are displayed in the upper-left and also highlighted in red
* commands are issued to the highlighted block or entity, which is displayed in the upper-right
* use one of the following keys to issue commands to the current selected npcs (reassignable through keybinds gui (f7))
  + Z -- clear current command (such as guard or follow)
  + X -- attack target (entity target), move to target while looking for things to attack (block target)
  + C -- follow/guard target (entity targe), move to target while ignoring potential attack targets (block target), mount target (horse target)
  + V -- set home point / (sneaking) clear home target
  + B -- set upkeep point / (sneaking) clear upkeep point

TEAMS

* All teams are derived from owning players team.
* If owning player is not on a team, or npc is not owned, they are considered NEUTRAL.
* Neutral will not attack other teams, but will defend themselves vs attacks and attack 'bandits'.
* Bandits / hostile NPCs will not be part of the team system and will have customized AI
* Combat npcs will attack other combat NPCS if the attacker has a team and the target has a team, and they are not the same team
* Npcs whos owner are not on a team may only be commanded by their owner. If their owner is no a team, they may also be commanded by team members.

NPC INVENTORIES

* All npcs have full inventory accessibility for equipment slots
  + main-hand slot determines work/combat type/inventory-size for couriers/bonus effectiveness for bards
  + armor slots are used for actual armor
  + upkeep slot is for upkeep orders -- where to get the food
  + orders slot is for work/combat/route orders -- where to work / what to fight / what items to move

* All npcs have an 'upkeep order' slot
  + simple work-order that denotes a single point for npc upkeep.
  + allows setting the amount of upkeep to withdraw
  + allows setting the block-side to withdraw from (in the case of sided inventories)

* all npcs have an 'orders' slot
  + worker = work order, denotes what sites and in what order a workers should work
  + courer = routing slip, denotes what sites and what items a courier should move
  + combat = combat order, denotes a patrol route for combat-based npcs.

ORDERS ITEMS

C:\git_source\git\AncientWarfare2\src\main\resources\assets\ancientwarfare\textures\items\npc\upkeep_order.pngC:\git_source\git\AncientWarfare2\src\main\resources\assets\ancientwarfare\textures\items\npc\work_order.pngC:\git_source\git\AncientWarfare2\src\main\resources\assets\ancientwarfare\textures\items\npc\routing_order.pngC:\git_source\git\AncientWarfare2\src\main\resources\assets\ancientwarfare\textures\items\npc\combat_order.png

* Orders items all function in the same way.
* Press the primary item-use key (Z key by default) to assign a point/add a point.
* Right-click to open the GUI. From here you can alter settings, setup filters (for courier route slips), and change the order of points.
* While holding a combat/upkeep/route/work order item the route and blocks that it is referring to will be highlighted.

NPC UPKEEP

C:\git_source\git\AncientWarfare2\src\main\resources\assets\ancientwarfare\textures\items\npc\upkeep_order.pngC:\git_source\git\AncientWarfare2\src\main\resources\assets\ancientwarfare\textures\items\npc\food_bundle.png

* Player owned NPCs may have their upkeep point set through the use of the 'Upkeep Orders' item
* Alternatively, if no upkeep item is present, an NPC will auto-assign an upkeep point to the nearest in range friendly town-hall.
* A player may use the command-baton to override this 'auto assigned upkeep point' to one of their choosing however while it does not require a upkeep-orders item ,using the command baton does not give the option to set the amount of upkeep to withdraw or the side of the inventory to withdraw from.
* If an NPC is hungry he will attempt to move to his upkeep point if he has one, otherwise he will return to his assigned home point.
* Hungry NPCs will not work. Combat NPCs will still retaliate but will not seek out new targets.

NPC WORKERS

C:\git_source\git\AncientWarfare2\src\main\resources\assets\ancientwarfare\textures\items\npc\work_order.pngC:\git_source\git\AncientWarfare2\src\main\resources\assets\ancientwarfare\textures\items\core\iron_quill.pngC:\git_source\git\AncientWarfare2\src\main\resources\assets\ancientwarfare\textures\items\core\iron_hammer.png



* Worker work points may be set via work-orders item.
* If no work-orders item is present, NPC will work at whatever nearby worksites have work that are appropriate for his sub-type.
* Workers work-type is set by the item in their weapon/primary hand equipment slot. Worker types are:
  + Hoe = Farmer
  + Axe = Lumberjack
  + Pickaxe = Miner
  + Quill = Researcher
  + Hammer = Craftsman
* With no item equipped, a worker has no ‘work type’ and cannot work at any worksites – equip them first!
* A workers ‘effectiveness’ stat is determined by a combination of their level for their current work type and the material quality of their currently equipped tool. This effectiveness stat determines how much work they do for each work ‘tick’ at a worksite – the highest levels and tools result in a net ~2x increase in work output.

COMBAT NPCS

C:\git_source\git\AncientWarfare2\src\main\resources\assets\ancientwarfare\textures\items\npc\spawner_combat.pngC:\git_source\git\AncientWarfare2\src\main\resources\assets\ancientwarfare\textures\items\npc\combat_order.png

* Combat NPCs type is determined by their primary equipped weapon:
  + Sword = Soldier
  + Bow = Archer
  + Baton = Commander
  + Axe = Medic
  + Hammer = Engineer
* If no weapon (or unrecognized item) is equipped the NPC uses the ‘Soldier’ AI routines.
* Attack damage is set via the equipped weapons’ ‘Attack Damage’ item attribute – you should be able to equip soldiers with mod-added weapons as long as the modder made the weapon correctly. (On-hit stuff might not function appropriately, but damage bonuses should apply correctly)
* Combat NPCs may be issued a patrol route through the use of a ‘Combat Orders’ item. Set the item up appropriately and place it into the work orders slot.
* Soldiers have a simple default melee attack.
* Archers have a simple default ranged arrow attack.
* Medic NPCs will attempt to heal any nearby friendly injured NPCs (or players). They also use
* Commanders give a slight (2-point) damage bonus to nearby friendly combat NPCs.
* Engineers will attempt to repair nearby friendly mechanical units (gates, vehicles)

COURIER NPC

C:\git_source\git\AncientWarfare2\src\main\resources\assets\ancientwarfare\textures\items\npc\spawner_courier.pngC:\git_source\git\AncientWarfare2\src\main\resources\assets\ancientwarfare\textures\items\npc\routing_order.pngC:\git_source\git\AncientWarfare2\src\main\resources\assets\ancientwarfare\textures\items\core\backpack.png

* Courier NPCs move items between inventories.
* Couriers need a ‘Routing Orders’ item in their orders slot in order to function. This item denotes what inventories to move between and what action to take at each stop.
* Couriers’ main-hand item is a ‘Backpack’. They MUST be equipped with a backpack item in order to have storage inventory to store the items they are moving.
* You may remove a couriers’ backpack at any time to view/alter the items that are in it or replace it with a higher capacity.

TRADER NPCS

C:\git_source\git\AncientWarfare2\src\main\resources\assets\ancientwarfare\textures\items\npc\spawner_trader.png

* Trader NPCs work as configurable villagers regarding trades.
* Each trader has access to a random selection of the trades specified for their type and level.
* The trader trade-list may be specified in the ‘trades.cfg’ file located at: .minecraft/config/AWConfig/npc/trades/trades.cfg
* An entry in the trades file is specified such as:
  + trade:
  + npcTypes=trader,bandit.trader,viking.trader,pirate.trader
  + npcLevels=0-10
  + outputItem=stick
  + outputQuantity=2
  + outputMeta=0
  + inputItem1=stick
  + inputQuantity1=1
  + inputMeta1=0
  + :endtrade
* There is a single input item, and up to 9 output items.
* Each entry in the trade list denotes a single trade. The fields are:
  + npcTypes field may contain multiple npc types, comma separated.
  + npcLevels field accepts a range of levels, min and max, separated via a minus sign.
  + outputItem field should contain the registry-name of a single item
  + outputQuantity field should contain the quantity obtained from the trade
  + outputMeta field is the damage value of the item obtained from the trade
  + inputItem1…9 should contain the registry item name for the input for the trade
  + inputQuantity1…9 should contain the needed quantity for the input item for the trade
  + inputMeta1…9 should contain the metadata needed for the input item
* Currently no wildcards are allowed, and each input item must be specified exactly (e.g. you must specify ‘white wool’ rather than ‘any wool’ via the damage number). This will likely be adapted and expanded in the future to include metadata wildcards for input items as well as matching of intput-item nbt tags for using nbt-based items in the trades.
* Trade list config files must match between client and server or VERY STRANGE things will happen.

BARD NPCS

C:\git_source\git\AncientWarfare2\src\main\resources\assets\ancientwarfare\textures\items\npc\spawner_bard.png

* Bard NPCs are capable of playing user specified sounds at configurable intervals.
* The bards sounds GUI may be accessed from his inventory GUI via the ‘Adv. Gui’ button.
* Sounds must be specified as they would be in the code – e.g.: “random.levelup”, “eating”, “mob.horse.land”, “game.tnt.primed”, “random.orb”, “random.pop”
* A minimum play frequency may be specified, in ticks, via the ‘Play Delay’ value.
* A random chance to play the selected sound may be specified in the ‘Play Chance’ value, this is an integer value between 0-100, higher values = more likely to play.
* The length of the tune may be specified in the ‘Play Length’ value – checks to see if the bard should play again will be delayed by at least this number of ticks, to ensure that the previously started sound was finished playing.
* Custom sounds may be added to the available list through the creation of a custom resource pack and sounds.json file.
  + ---TODO add implementation details regarding adding sound packs

PRIEST NPCS

C:\git_source\git\AncientWarfare2\src\main\resources\assets\ancientwarfare\textures\items\npc\spawner_priest.png

* Player owned priest NPCs serve a single purpose – resurrecting their fallen comrades.
* This is done at a town-hall – if you do not have a town hall setup, a priest has no function.
* The list of dead/resurrect-able NPCs is stored and viewable in the town-hall GUI.
* If there are resurrect-able NPCs in the dead NPC list, the priest should move towards the town hall and begin his ritual. Once complete, the dead NPC will be brought back to life at half-health with no equipment (it was dropped when they died). They must be re-equipped and issued new orders (if applicable). They will retain any levels, experience, custom name, and custom texture references they had at the time of death.

NPC HORSE RIDING

* All player-owned NPCs may be ordered to mount and ride a horse while performing their duties.
* An NPC may be commanded to ride a horse via the command baton by selecting the NPC and issuing a move/guard/mount command (default C key) while targeting the horse to be mounted. The horse must be tamed and saddled (the saddle will disappear while the NPC is riding, but is still present)
* To dismount an NPC that is riding a horse, select the NPC with the command baton and issue another move/guard/mount command while targeting the horse he is riding.

NPC LEVELING

* All npcs have a base (shared) level and experience value.
* Base level determines max hit-points.
* Sub-type level determines bonus damage or work-effectiveness.
* Sub-typed npcs also store level and xp per sub-type.
  + For npcs that can change subtypes (combat,worker), current level is looked up per-subtype;
  + Each work/combat type will return a different level/experience value.
* When an NPC gains experience, it is applied to both his base-level/xp, as well as the sub-type level/xp
* Npcs with no subtypes will have the same value for their base and (single) subtype level/xp values.

NPC HEALING

* All npcs will 'heal' over time on their own, slowly. The time/delay between heals is customizable via config file.
* Player-owned NPCs will require upkeep be met in order to heal – they will not auto-heal when hungry.

NPC SKIN PACKS

* Custom skin-packs may be placed into the config/AWConfig/npc/skin\_pack/ folder
* Skin-packs should contain only minecraft skin-format .png files and a single skin-pack.meta file
  + The skin-pack.meta file is should be specified with one entry per line, in the format of:
    - npcType.npcSubType=image\_name
  + Each npc type may be specified multiple times with separate images
  + When multiple images are found for an npc, the npc will be assigned a random skin from those found for his type/subtype
  + Examples:
    - worker=worker1.png
    - worker=worker2.png
    - worker.farmer=farmer1.png
* Custom npc types may be specified in the skin-pack.meta file, these custom npc types may be referenced in the npc gui 'custom texture' field.

NPC FACTIONS

* There are currently five factions (and the corresponding biomes they should be found in)(biome is merely a guideline and not enforced by the NPC):
  + Bandits/Brigands (plains/forests)
  + Pirates (islands/oceans/coastal areas)
  + Vikings (tundra, taiga, colder biomes)
  + Jungle Natives (jungle, tropical forests, warm/wet biomes)
  + Desert Natives (deserts, dry/hot biomes)
* There are currently ten subtypes for each faction:
  + Soldier
  + Mounted Soldier
  + Elite Soldier
  + Archer
  + Mounted Archer
  + Elite Archer
  + Leader
  + Elite Leader
  + Trader
  + Priest
* The only difference between the normal and ‘elite’ varieties is that the ‘elite’ varieties require a much higher faction standing before they will be neutral. The elite varieties are meant to serve as treasury / elite guards for areas in structures that a player is not supposed to enter regardless of faction standing (or at least not until they have very high standing).
* The ‘elite’ faction offset is configurable via the npcs config file.
* Starting faction standing for each faction is configurable via the npcs config file.
* Faction standing for each faction vs. each other faction is configurable via config file. You can have bandits be hostile vs. pirates for instance.
* Faction standing may be lost by killing any npc belonging to a particular faction. The amount lost is configurable via config file.
* Faction standing may be gained by trading with a trader belonging to a particular faction. The amount of standing gained per trade is configurable via config file.
* Faction standing is tracked per-player – a players’ team has no effect on their faction standing.

CREATIVE MODE INTERACTION AND CONTROL

* Players in creative mode have access to the inventory for all faction-based NPCs. Faction based NPCs do not use their upkeep or work orders slots, so these slots may be filled with items to act as custom-rewards for when the NPC is slain.
* Players in creative mode may command a faction-based NPC to follow them by shift-right clicking on it.
* Players in creative mode have additional controls available on all NPCs. These controls are accessible via the ‘Creative Controls’ button in the NPCs main inventory screen. The additional controls are:
  + Set Owner Name – most useful for reassigning a player-owned NPC to a different player. Has no effect on faction-owned NPCs.
  + Enable/Disable wander – turn on/off the NPCs wander AI. Temporary setting, not saved. Should be used to aid in structure or map creation and NPC positioning.
  + Attack Damage Override – if set to >=0, this value will be used for attack damage instead of the NPCs equipped weapon. Most useful for creating custom NPC bosses or subtypes.
  + Armor Value Override – if set to >=0, this value overrides the armor value from whatever armor the NPC is wearing. Most useful for creating custom NPC bosses or subtypes that should be armored but still show their texture.
  + Max Health Override – if set to >0, this value will be used to determine the NPCs max health. Can be used in combination with the other overrides to create custom NPC bosses.
  + Custom Texture Override – may specify a custom lookup type for the NPC. This custom type should exactly match the custom type specified in the skin-pack file. E.g. If the skin-pack has ‘custom.boss.1=boss1.png”, then this field should say ‘custom.boss.1’. This field/input box will be moving to the main NPC inventory/gui shortly to allow for users to specify custom textures for their player-owned npcs.