Color notation
Written, tested
Written, not tested
Partially written, not tested
Not written, not tested

contains
provides building kind data
provides profession data
provides stat bonuses
lives in
works in
participates in
starts
ends in (provides items and resources)
one per turn

TypeObject
#name: std::string
#description: std::string
+TypeObject()
+~*TypeObject()*
+get_name(result: std::string&): size_t
+get_description(result: std::string&): size_t
+set_name(value: char_t*): size_t
+set_name(value: std::string): size_t
+set_description(value: char_t*): size_t
+set_description(value: std::string): size_t
+what(out: std::string&): size_t

IngameStorage
+items: std::vector<Item>
+professions: std::vector<TypeProfession>
+buildings: std::vector<TypeBuilding>
#db_name: std::string
+IngameStorage()
+IngameStorage(db: std::string)
+~IngameStorage()
+get_item(index: size_t, result: Item&): size_t
+get_profession(index: size_t, result: TypeProfession&): size_t
+get_building(index: size_t, result: TypeBuilding&): size_t
+get_db_name(result: std::string&): size_t
+set_item(value: Item): size_t
+set_profession(value: TypeProfession): size_t
+set_building(value: TypeBuilding): size_t
+set_db_name(value: std::string): size_t

TypeProfession
#consumation: std::vector<size_t>
#can_slave: bool
+TypeProfession()
+~*TypeProfession()*
+get_consumation(result: std::vector<size_t>&): size_t
+get_consumation(index: size_t, result: size_t&): size_t
+get_can_slave(result: bool&): size_t
+set_consumation(value: std::vector<size_t>): size_t
+set_consumation(inde: size_t, value: size_t>): size_t
+set_can_slave(value: bool): size_t
+what(out: std::string&): size_t

TypeBuilding
#max_employees: size_t
#producable: std::vector<bool>
#resources: std::vector<size_t>
#cost: std::vector<size_t>
#building_time: size_t
#living_space: size_t
+TypeBuilding()
+~*TypeBuilding()*
+get_max_employees(result: size_t&): size_t
+get_living_space(result: size_t&): size_t
+get_producable(result: std::vector<bool>&): size_t
+get_producable(index: size_t, result: bool&): size_t
+get_resources(result: std::vector<size_t>&): size_t
+get_resources(index: size_t, result: size_t&): size_t
+get_cost(result: std::vector<size_t>&): size_t
+get_cost(index: size_t, result: size_t&): size_t
+get_building_time(result: size_t): size_t
+set_max_employees(value: size_t): size_t
+set_living_space(value: size_t): size_t
+set_producable(value: std::vector<bool>): size_t
+set_producable(index: size_t, value: bool): size_t
+set_resources(value: std::vector<size_t>): size_t
+set_resources(index: size_t, value: size_t): size_t
+set_cost(value: std::vector<size_t>): size_t
+set_cost(index: size_t, value: size_t): size_t
+set_building_time(value: size_t): size_t
+what(out: std::string&): size_t

Human
#name: std::string
#profession: TypeProfession&
#skills: std::vector<size_t>
#saga: PersonalSaga
#combat_stats: CombatStats
#gender: bool
#house_id: signed size_t
+Human()
+~Human()
+get_name(result: std::string&): size_t
+get_profession(result: TypeProfession&): size_t
+get_skills(result: std::vector<size_t>&): size_t
+get_skills(index: size_t, result: size_t&): size_t
+get_saga(result: PersonalSaga&): size_t
+get_saga_killed(result: size_t&): size_t
+get_saga_raids(result: size_t&): size_t
+get_saga_looted(result: size_t&): size_t
+get_saga_enslaved(result: size_t&): size_t
+get_combat_stats(result: CombatStats&): size_t
+get_combat_stats_wounds(result: std::pair<size_t, size_t>&): size_t
+get_combat_stats_wounds(index:size_t, result: size_t&): size_t
+get_combat_stats_melee(result: size_t&): size_t
+get_combat_stats_ranged(result: size_t&): size_t
+get_combat_stats_defense(result: size_t&): size_t
+get_combat_stats_initiative(result: size_t&): size_t
+get_gender(result: bool&): size_t
+get_house_id(result: signed size_t&): size_t
+set_name(value: std::string&): size_t
+set_profession(value: TypeProfession&): size_t
+set_skills(value: std::vector<size_t>): size_t
+set_skills(index: size_t, value: size_t): size_t
+set_saga(value: PersonalSaga): size_t
+set_saga_killed(value: size_t): size_t
+set_saga_raids(value: size_t): size_t
+set_saga_looted(value: size_t): size_t
+set_saga_enslaved(value: size_t): size_t
+set_combat_stats(value: CombatStats): size_t
+set_combat_stats_wounds(value: std::pair<size_t, size_t>): size_t
+set_combat_stats_wounds(index: size_t, value: size_t): size_t
+set_combat_stats_melee(value: size_t): size_t
+set_combat_stats_ranged(value: size_t): size_t
+set_combat_stats_defense(value: size_t): size_t
+set_combat_stats_initiative(value: size_t): size_t
+set_gender(value: bool): size_t
+set_house_id(value: signed size_t): size_t
+what(out: std::string&): size_t
+consume(): size_t
+add_kill(): size_t
+add_raid(): size_t
+add_loot(value: size_t): size_t
+add_enslaved(value: size_t): size_t
+wound(): size_t
+raise_melee(): size_t
+raise_ranged(): size_t
+raise_defense(): size_t
+raise_initiative(): size_t
+raise_skill(index: size_t): size_t

CombatStats
+wounds: std::pair<size_t, size_t>
+melee_attack: size_t
+ranged_attack: size_t
+defense: size_t
+initiative: size_t

PersonalSaga
+enemies_killed: size_t
+raids_count: size_t
+gold_looted: size_t
+people_enslaved: size_t

Item
#name: std::string
#description: std::string
#cost: std::vector<size_t>
#bonuses: CombatStats
+Item()
+~*Item()*
+get_name(result: std::string&): size_t
+get_description(result: std::string&): size_t
+get_cost(result: std::vector<size_t>&): size_t
+get_cost(index: size_t, result: size_t&): size_t
+get_bonuses(result: CombatStats&): size_t
+get_bonuses_wounds(result: size_t&): size_t
+get_bonuses_melee(result: size_t&): size_t
+get_bonuses_ranged(result: size_t&): size_t
+get_bonuses_defense(result: size_t&): size_t
+get_bonuses_initiative(result: size_t&): size_t
+set_name(value: std::string): size_t
+set_description(value: std::string): size_t
+set_bonuses(value: CombatStats): size_t
+set_bonuses_wounds(value: size_t): size_t
+set_bonuses_melee(value: size_t): size_t
+set_bonuses_ranged(value: size_t): size_t
+set_bonuses_defense(value: size_t): size_t
+set_bonuses_initiative(value: size_t): size_t
+set_cost(value: std::vector<size_t>): size_t
+set_cost(index: size_t, value: size_t): size_t
+what(out: std::string&): size_t

Inventory
#items_id: std::vector<Item*>
+Inventory()
+~Inventory()
+get_items_id(result: std::vector<signed size_t>&): size_t
+get_items_id(index: size_t, result: signed size_t&): size_t
+get_melee_bonus(result: size_t&): size_t
+get_ranged_bonus(result: size_t&): size_t
+get_defense_bonus(result: size_t&): size_t
+set_items_id(value: std::vector<signed size_t>): size_t
+set_items_id(index: size_t, value: signed size_t): size_t
+what(out: std::string&): size_t

Building
#employees_id: std::vector<signed size_t>
#production_queue: std::vector<signed size_t>
#living_space: size_t
#kind: TypeBuilding&
+Building(kind: TypeBuilding&)
+~Building();
+get_employees_id(result: std::vector<signed size_t>&): size_t
+get_employees_id(index: size_t, result: signed size_t&): size_t
+get_production_queue(result: std::vector<signed size_t>&): size_t
+get_production_queue(index: size_t, result: signed size_t&): size_t
+add_to_queue(value: signed size_t): size_t
+remove_from_queue(): size_t
+remove_from_queue(index: size_t): size_t
+get_living_space(result: size_t&): size_t
+add_inhabitant(): size_t
+remove_inhabitant(): size_t
+set_employees_id(value: std::vector<signed size_t>): size_t
+set_employees_id(index: size_t, value: signed size_t): size_t
+set_production_queue(value: std::vector<signed size_t>): size_t
+set_production_queue(index: size_t, value: signed size_t): size_t
+set_living_space(value: size_t>): size_t
+what(out: std::string&): size_t
+produce(): size_t

Village
#villagers: std::vector<Human*>
#buildings: std::vector<Building*>
#construction: std::vector<ConstructionSite*>
#resources: std::vector<size_t>
#items: std::vector<Item*>
#raid: Raid*;
+Village()
+~Village()
+get_villager_by_id(index: signed size_t, result: Human*&): size_t
+get_building_by_id(index: signed size_t, result: Building*&): size_t
+get_construction_site_by_id(index: signed size_t, result: ConstructionSite*&): size_t
+get_resources_by_id(index: signed size_t, result: size_t&): size_t
+get_villagers(result: std::vector<Human*>&): size_t
+get_buildings(result: std::vector<Building*>&): size_t
+get_construction(result: std::vector<ConstructionSite*>&): size_t
+get_resources(result: std::vector<size_t>&): size_t
+get_raid(result: Raid*&): size_t
+get_raid(): Raid&
+start_raid(terms: std::vector<size_t>): size_t
+end_raid();
+add_villager(if_male: bool): size_t
+add_slave(if_male: bool): size_t
+remove_villager(index: signed size_t): size_t
+add_contruction(kind: TypeBuilding&): size_t
+remove_construction(index: signed size_t): size_t
+add_resource(index: signed size_t, amount: size_t): size_t
+remove_resource(index: signed size_t, amount: size_t): size_t
+turn(): size_t
+what(out: std::string&): size_t

RaidEvent
#name: std::string
#description: std::string
#difficulty: size_t
#skill_id: size_t
#raid_turns: size_t&
#raid_resources: std::vector<size_t>&
#raid_participants: std::vector<Human*>&
#raid_loot: std::vector<Item*>&
+RaidEvent()
+RaidEvent(name: std::string, description: std::string, difficulty: size_t, raid_turns: size_t&, raid_resources: std::vector<size_t>&, raid_participants: std::vector<Humanm*>&, raid_loot: std::vector<Item*>&)
+~*RaidEvent()*
+get_name(result: std::string&): size_t
+get_description(result: std::string&): size_t
+get_difficulty(result: size_t&): size_t
+get_total_misc_stat(result: size_t&): size_t
+get_total_misc_stat(): size_t
+get_save_data(result: prototypes::RaidEventTable&): size_t
+set_name(value: std::string): size_t
+set_description(value: std::string): size_t
+set_difficulty(value: std::string): size_t
+what(out: std::string&): size_t
+execute()
+make_roll(): size_t
+make_roll(result: size_t&): size_t
+make_roll(die: size_t): size_t
+make_roll(die: size_t, result: size_t&): size_t

Raid
#participants: std::vector<Human*>
#slaves: std::vector<Human*>
#resources: std::vector<size_t>
#loot: std::vector<Item*>
#turns_left: size_t
#terms: std::vector<size_t>
+Raid();
+Raid(data: prototypes::RaidTable);
+~Raid();
+get_participants(result: std::vector<Human*>&): size_t
+get_participants(index: size_t, result: Human*&): size_t
+get_participants(index: size_t): Human&
+get_participants(): std::vector<Human*>&
+get_participants_count(result: size_t&): size_t
+get_participants_count(): size_t
+get_slaves(result: std::vector<Human*>&): size_t
+get_slaves(index: size_t, result: Human*&): size_t
+get_slaves(index: size_t): Human&
+get_slaves(): std::vector<Human*>&
+get_resources(result: std:vector<size_t>&): size_t
+get_resources(index: size_t< result: size_t&): size_t
+get_loot(result: std::vector<Item*>&): size_t
+get_loot(index: size_t, result: Item*&): size_t
+get_loot(index: size_t): Item&
+get_loot(): std::vector<Item*>&
+get_loot(index: size_t, result: std::vector<Item*>&): size_t
+get_turns_left(result: size_t&): size_t
+get_turns_left(): size_t&
+get_terms(result: std::vector<size_t>&): size_t
+get_terms(index: size_t, result: size_t&): size_t
+set_participants(value: std::vector<Human*>): size_t
+set_participants(index: size_t, value: Human*): size_t
+set_slaves(value: std::vector<Human*>): size_t
+set_slaves(index: size_t, value: Human*): size_t
+set_resources(value: std::vector<size_t>): size_t
+set_resources(index: size_t, value: size_t): size_t
+set_loot(value: std::vector<Item*>): size_t
+set_loot(index: size_t, value: Item*): size_t
+set_turns_left(value: size_t): size_t
+set_terms(value: std::vector<size_t>): size_t
+set_terms(index: size_t, value: size_t): size_t
+what(out: std::string&): size_t
+distribute_loot(village_resources: std::vector<size_t>&, village_items: std::vector<Item*>&, village_people: std::vector<Human*>&): size_t

DiseaseEvent
+DiseaseEvent()
+DiseaseEvent(name: std::string, description: std::string, raid_turns: size_t&, raid_participants: std::vector<Human*>&, raid_loot: std::vector<Item*>&)
+~*DiseaseEvent()*
+execute(): size_t

GoodHuntEvent
+GoodHuntEvent()
+GoodHuntEvent(name: std::string, description: std::string, raid_turns: size_t&, raid_participants: std::vector<Human*>&, raid_loot: std::vector<Item*>&)
+~*GoodHuntEvent()*
+execute(): size_t

NoWindEvent
+NoWindEvent()
+NoWindEvent(name: std::string, description: std::string, raid_turns: size_t&, raid_participants: std::vector<Human*>&, raid_loot: std::vector<Item*>&)
+~*NoWindEvent()*
+execute(): size_t

StormEvent
+StormEvent()
+StormEvent(name: std::string, description: std::string, raid_turns: size_t&, raid_participants: std::vector<Human*>&, raid_loot: std::vector<Item*>&)
+~*StormEvent()*
+execute(): size_t

BattleEvent
+BattleEvent()
+BattleEvent(name: std::string, description: std::string, raid_turns: size_t&, raid_participants: std::vector<Human*>&, raid_loot: std::vector<Item*>&)
+~*BattleEvent()*
+execute(): size_t

RobbedVillageEvent
+RobbedVillageEvent()
+RobbedVillageEvent(name: std::string, description: std::string, raid_turns: size_t&, raid_participants: std::vector<Human*>&, raid_loot: std::vector<Item*>&)
+~*RobbedVillageEvent()*
+execute(): size_t

MonasteryEvent
+MonasteryEvent()
+MonasteryEvent(name: std::string, description: std::string, raid_turns: size_t&, raid_participants: std::vector<Human*>&, raid_loot: std::vector<Item*>&)
+~*MonasteryEvent()*
+execute(): size_t

SunkenSailorEvent
+SunkenSailorEvent()
+SunkenSailorEvent(name: std::string, description: std::string, raid_turns: size_t&, raid_participants: std::vector<Human*>&, raid_loot: std::vector<Item*>&)
+~*SunkenSailorEvent()*
+execute(): size_t