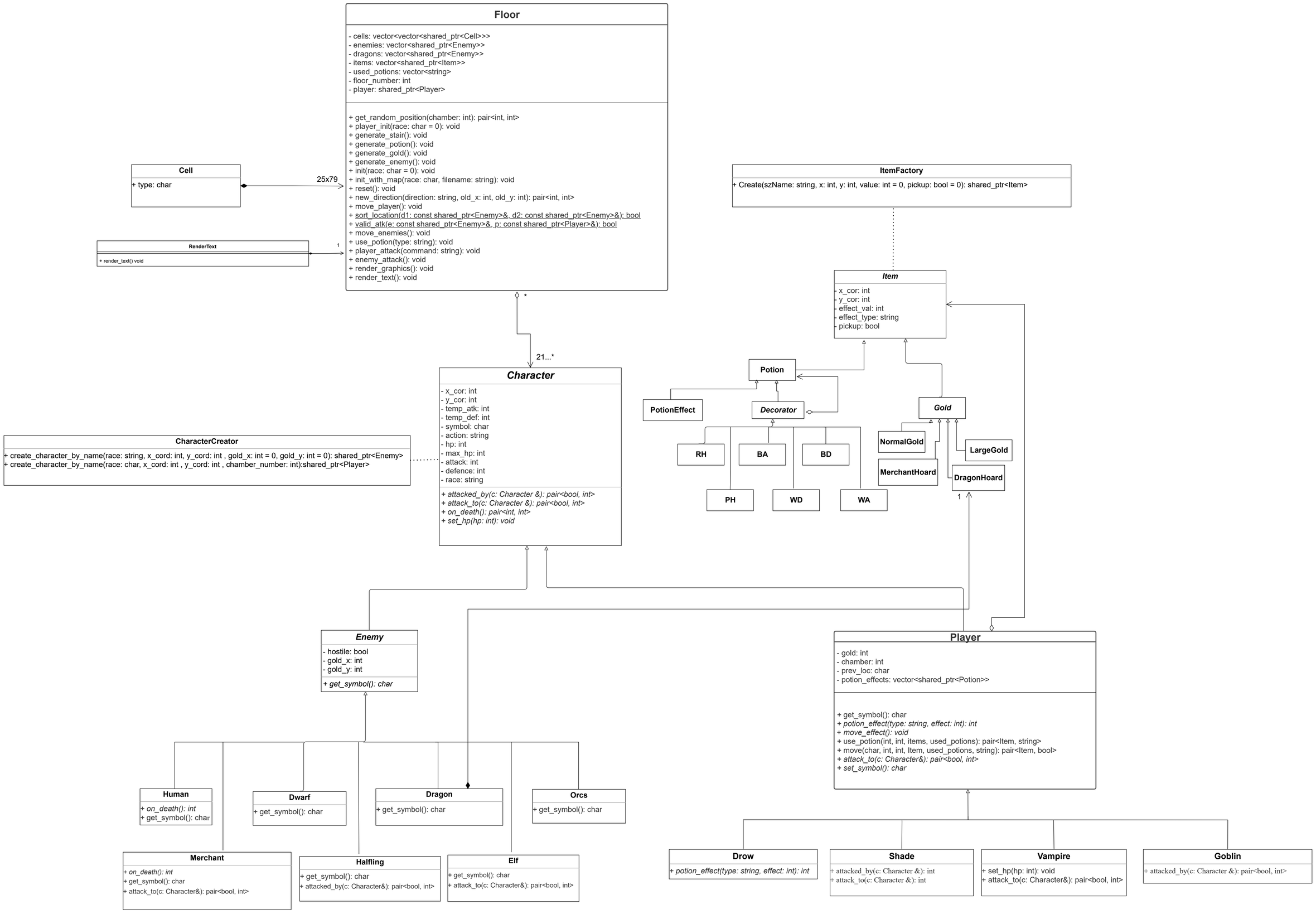
Floor
- cells: vector<vector<shared_ptr<Cell>>>
- enemies: vector<shared_ptr<Enemy>>
- dragons: vector<shared_ptr<Enemy>>
- items: vector<shared_ptr<Item>>
- used_potions: vector<string>
- floor_number: int
- player: shared_ptr<Player>
+ get_random_position(chamber: int): pair<int, int>
+ player_init(race: char = 0): void
+ generate_stair(): void
+ generate_potion(): void
+ generate_gold(): void
+ generate_enemy(): void
+ init(race: char = 0): void
+ init_with_map(race: char, filename: string): void
+ reset(): void
+ new_direction(direction: string, old_x: int, old_y: int): pair<int, int>
+ move_player(): void
+ sort_location(d1: const shared_ptr<Enemy>&, d2: const shared_ptr<Enemy>&): bool
+ valid_atk(e: const shared_ptr<Enemy>&, p: const shared_ptr<Player>&): bool
+ move_enemies(): void
+ use_potion(type: string): void
+ player_attack(command: string): void
+ enemy_attack(): void
+ render_graphics(): void
+ render_text(): void
Cell
+ type: char
25x79
RenderText
+ render_text() void
1
*
21...*
ItemFactory
+ Create(szName: string, x: int, y: int, value: int = 0, pickup: bool = 0): shared_ptr<Item>
Item
- x_cor: int
- y_cor: int
- effect_val: int
- effect_type: string
- pickup: bool
Potion
PotionEffect
Decorator
RH
BA
BD
PH
WD
WA
Gold
NormalGold
MerchantHoard
LargeGold
DragonHoard
1
Character
- x_cor: int
- y_cor: int
- temp_atk: int
- temp_def: int
- symbol: char
- action: string
- hp: int
- max_hp: int
- attack: int
- defence: int
- race: string
+ attacked_by(c: Character &): pair<bool, int>
+ attack_to(c: Character &): pair<bool, int>
+ on_death(): pair<int, int>
+ set_hp(hp: int): void
CharacterCreator
+ create_character_by_name(race: string, x_cord: int, y_cord: int , gold_x: int = 0, gold_y: int = 0): shared_ptr<Enemy>
+ create_character_by_name(race: char, x_cord: int , y_cord: int , chamber_number: int):shared_ptr<Player>
Enemy
- hostile: bool
- gold_x: int
- gold_y: int
+ get_symbol(): char
Player
- gold: int
- chamber: int
- prev_loc: char
- potion_effects: vector<shared_ptr<Potion>>
+ get_symbol(): char
+ potion_effect(type: string, effect: int): int
+ move_effect(): void
+ use_potion(int, int, items, used_potions): pair<Item, string>
+ move(char, int, int, Item, used_potions, string): pair<Item, bool>
+ attack_to(c: Character&): pair<bool, int>
+ set_symbol(): char
Human
+ on_death(): int
+ get_symbol(): char
Dwarf
+ get_symbol(): char
Dragon
+ get_symbol(): char
Orcs
+ get_symbol(): char
Merchant
+ on_death(): int
+ get_symbol(): char
+ attack_to(c: Character&): pair<bool, int>
Halfling
+ get_symbol(): char
+ attacked_by(c: Character&): pair<bool, int>
Elf
+ get_symbol(): char
+ attack_to(c: Character&): pair<bool, int>
Drow
+ potion_effect(type: string, effect: int): int
Shade
+ attacked_by(c: Character &): int
+ attack_to(c: Character &): int
Vampire
+ set_hp(hp: int): void
+ attack_to(c: Character&): pair<bool, int>
Goblin
+ attacked_by(c: Character &): pair<bool, int>