

<<utility>> FuncionesAuxiliares
+ DIFICULTADES: dict + RUTA_PUNTAJES: str
+ cargar_casilla(c: Casilla): str

Trampa
+ activas: list
+ __init__() + permite_jugador(): bool + permite_enemigo(): bool

Muro
+ permite_jugador(): bool + permite_enemigo(): bool

Tunel
+ permite_jugador(): bool + permite_enemigo(): bool

Camino
+ permite_jugador(): bool + permite_enemigo(): bool

Liana
+ permite_jugador(): bool + permite_enemigo(): bool

<<abstract>> Casilla
+ __init__() + permite_jugador(): bool + permite_enemigo(): bool

Mapa
- matriz: list[list[Casilla]]
+ __init__(matriz: list[list[Casilla]]) + obtener(f: int, c: int): Casilla + dimensiones(): tuple[int, int]

GeneradorMapa
- prob_camino: float = 0.60 - prob_muro: float = 0.20 - prob_liana: float = 0.10 - prob_tunel: float = 0.10 + salida_fila: int + salida_col: int
+ __init__() + generar(): list[list[Casilla]] - _generar_matriz_basica(): list[list[Casilla]] - _elegir_salida_borde(): tuple[int, int] - _hay_camino(matriz: list, fi: int, ci: int, ff: int, cf: int): bool + elegir_casilla(): Casilla

VentanaJuego
- juego_terminado: bool = False - jugador: Jugador - modo: str - dificultad: str - vel_enemigo: int - regen_rate: int - costo_correr: int - max_trampas: int - penalizacion: int - cantidad_enemigos: int - enemigos_eliminados: int = 0 - enemigos_atrapados: int = 0 - enemigos_escapados: int = 0 - penalizacion_total: int = 0 - mapa: list[list[Casilla]] - s_fila: int - s_col: int - j_fila: int - j_col: int - e_fila: int - e_col: int - objetivo_fila: int - objetivo_col: int - tam: int = 40 - tiempo_inicio: float - canvas: tk.Canvas - barra_energia: tk.Canvas - contador_trampa: int = 0 - trampas_activas: list[tuple[int, int]] - contador_activo: bool = False - corriendo: bool = False - enemigos: list[Enemigo] - jugador_grafico: list
+ __init__(nombre_jugador: str, modo_de_juego: str, dificultad: str) - eliminar_trampa(nf_eliminar: int, nc_eliminar: int): void - bajar_contador_trampa(): void - poner_trampa(event: tk.Event): void - cazador_palma(enemigo: Enemigo): void - contador_reaparicion(enemigo: Enemigo): void - respawn_enemigo(enemigo: Enemigo): void - dibujar_camino(x1: int, y1: int, x2: int, y2: int): void - dibujar_salida(x1: int, y1: int, x2: int, y2: int): void - dibujar_mapa(): void - dibujar_jugador(): void - dibujar_enemigo(enemigo: Enemigo): void - jugador_es_cazador(): bool - enemigo_es_cazador(): bool - puede_moverse_a(nf: int, nc: int): bool - enemigo_moverse_a(nf: int, nc: int): bool - calcular_puntaje(tiempo: int): int - actualizar_barra_energia(): void - regenerar_energia(): void - correr(event: tk.Event): void - soltar_correr(event: tk.Event): void - mover(df: int, dc: int): void - mover_arriba(event: tk.Event): void - mover_abajo(event: tk.Event): void - mover_izquierda(event: tk.Event): void + si hay alguien leyendo esto, acomoda postura, descansa un rato y ve por agua :3 - mover_derecha(event: tk.Event): void - siguiente_paso_hacia(origen_f: int, origen_c: int, dest_f: int, dest_c: int): tuple - mover_enemigo(enemigo: Enemigo): void

Enemigo
+ fila: int + col: int + estado: str = "activo" + tiempo_reaparicion: int = 0 + grafico: list
+ __init__(fila: int, col: int) + mover(df: int, dc: int): void

Jugador
+ fila: int + col: int + nombre: str + modo_de_juego: str + energia: int = 100
+ __init__(fila: int, col: int, nombre: str, modo_de_juego: str) + mover(df: int, dc: int): void + correr(): void + gastar_energia(cantidad: int): bool

Puntajes
<<static>> - RUTA_PUNTAJES: str = "datos/puntajes.json"
+ __init__() + agregar(nombre: str, puntaje: int, modo: str): void + extraer_top_5(modo: str): list[dict]

Registro
+ validar_registrar(nombre: str, label_error: tk.Label): bool

VentanaRegistro
+ nombre_jugador: str - entry: tk.Entry - label_error: tk.Label
+ __init__(parent: tk.Widget) + registrar(): void

VentanaElegirTop
+ __init__(parent: tk.Widget) + abrir_top_escaladores(): void + abrir_top_cazadores(): void + mostrar_top(datos: list, titulo: str): void

VentanaDificultad
+ dificultad: str
+ __init__(parent: tk.Widget) + seleccionar(diff: str): void

Ventana_modo_de_juego
- nombre_jugador: str + modo_elegido: str
+ __init__(parent: tk.Widget, nombre_jugador: str) + modo_escapa(): void + modo_cazador(): void

VentanaMenu
- nombre_guardado: str
+ __init__() + abrir_registro(): void + abrir_top(): void + abrir_juego(): void