## **UML Class Diagram**

## PathOutput

- \_AStarAlg: AStarAlgorithm
- + DisplayPath(std::ostream&): void
- + setAStar(AStarAlgorithm): void
- getCharForPt(int): char



## A\_star\_algorithm

- + outputPath: int [][]
- + \_arena: Arena
- storeG Cost: int
- currentPtX: int
- currentPtY: int
- TargetPtX: int
- TargetPtY: int
- + SampledPts: int [][]
- + compute(): void
- + getCost(int, int, int, int): double
- + isPtInObstacle(int, int): bool
- + isPtSampled(int, int): bool
- + setPtAsSampled(int, int)
- + setArena(Arena): void



## Arena

- + gridArena: int [][]
- + startPtX: int
- + startPtY: int
- + endPtX: int
- + endPtX: int
- + MaxX: int
- + MaxY: int
- + getWorkspace(int gridArena[][]):void
- displayWorkspace(): void
- getArenaSize(): void
- getStartPt(): void
- getEndPt(): void
- setStartPt(int, int): void
- setEndPt(int, int): void
- setMapSize(): void