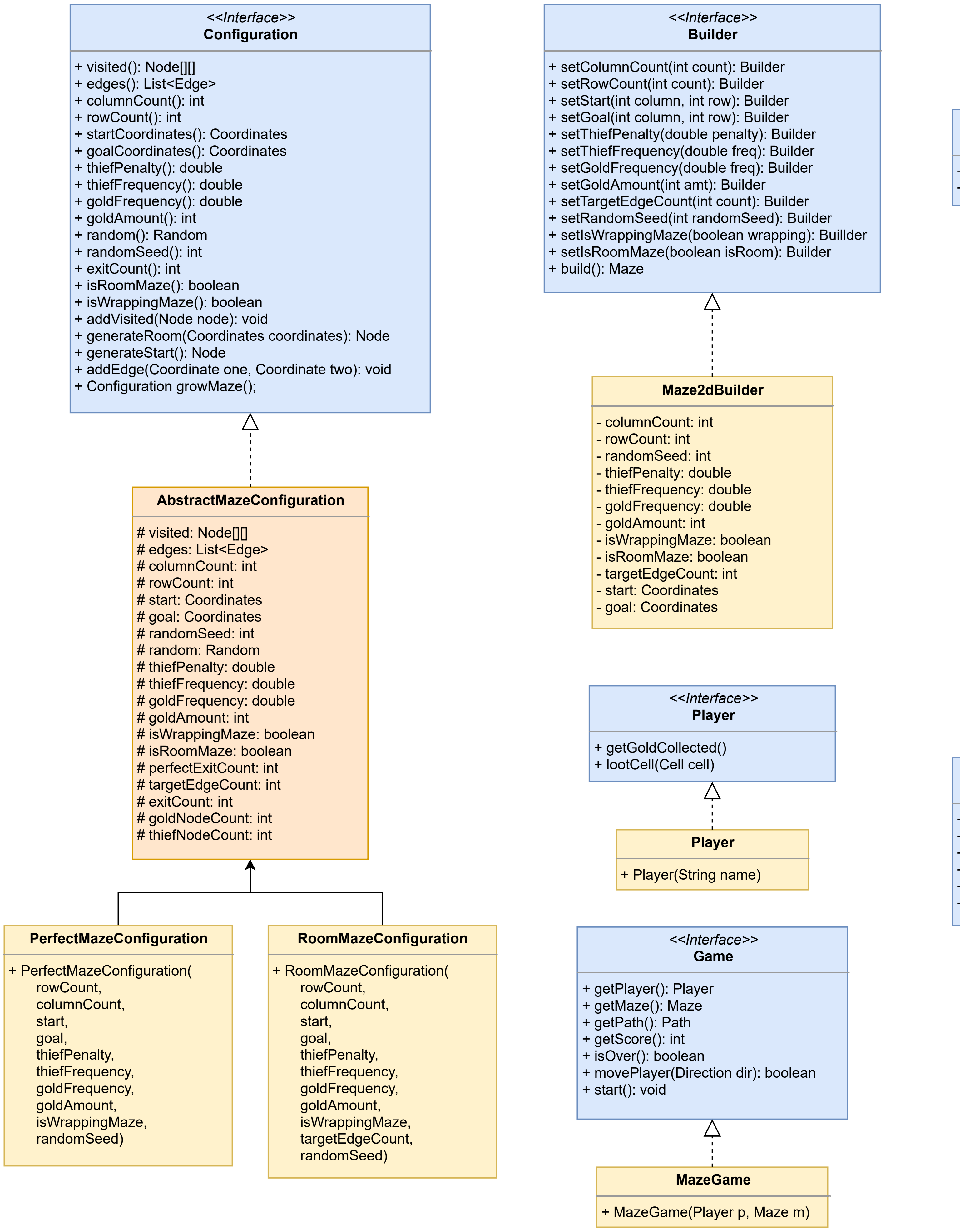
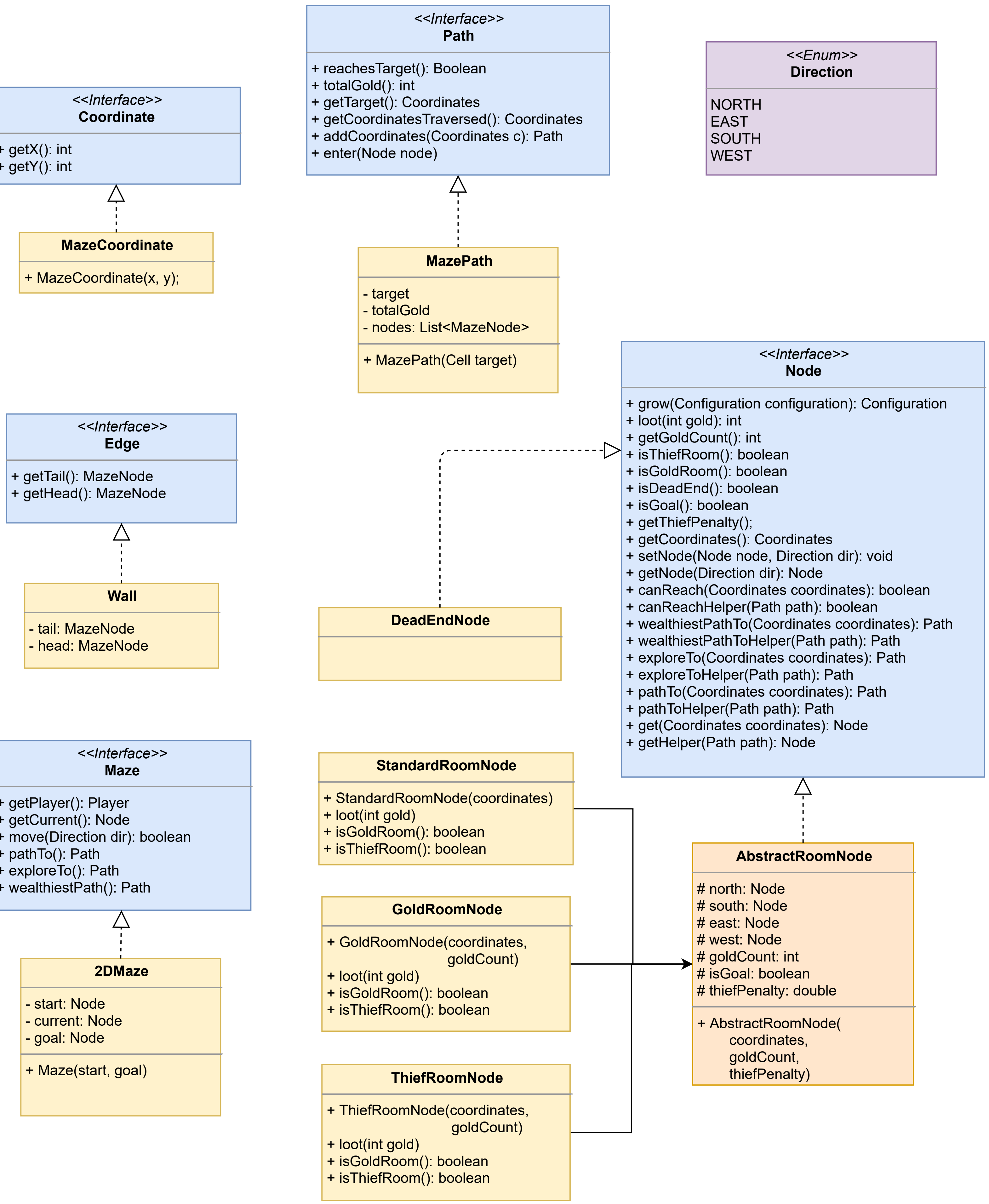

<<Interface>>
Configuration
+ visited(): Node[][]
+ edges(): List<Edge>
+ columnCount(): int
+ rowCount(): int
+ startCoordinates(): Coordinates
+ goalCoordinates(): Coordinates
+ thiefPenalty(): double
+ thiefFrequency(): double
+ goldFrequency(): double
+ goldAmount(): int
+ random(): Random
+ randomSeed(): int
+ exitCount(): int
+ isRoomMaze(): boolean
+ isWrappingMaze(): boolean
+ addVisited(Node node): void
+ generateRoom(Coordinates coordinates): Node
+ generateStart(): Node
+ addEdge(Coordinate one, Coordinate two): void
+ Configuration growMaze();
<<Interface>>
Builder
+ setColumnCount(int count): Builder
+ setRowCount(int count): Builder
+ setStart(int column, int row): Builder
+ setGoal(int column, int row): Builder
+ setThiefPenalty(double penalty): Builder
+ setThiefFrequency(double freq): Builder
+ setGoldFrequency(double freq): Builder
+ setGoldAmount(int amt): Builder
+ setTargetEdgeCount(int count): Builder
+ setRandomSeed(int randomSeed): Builder
+ setIsWrappingMaze(boolean wrapping): Buillder
+ setIsRoomMaze(boolean isRoom): Builder
+ build(): Maze
Maze2dBuilder
- columnCount: int
- rowCount: int
- randomSeed: int
- thiefPenalty: double
- thiefFrequency: double
- goldFrequency: double
- goldAmount: int
- isWrappingMaze: boolean
- isRoomMaze: boolean
- targetEdgeCount: int
- start: Coordinates
- goal: Coordinates
AbstractMazeConfiguration
# visited: Node[][]
# edges: List<Edge>
# columnCount: int
# rowCount: int
# start: Coordinates
# goal: Coordinates
# randomSeed: int
# random: Random
# thiefPenalty: double
# thiefFrequency: double
# goldFrequency: double
# goldAmount: int
# isWrappingMaze: boolean
# isRoomMaze: boolean
# perfectExitCount: int
# targetEdgeCount: int
# exitCount: int
# goldNodeCount: int
# thiefNodeCount: int
<<Interface>>
Player
+ getGoldCollected()
+ lootCell(Cell cell)
Player
+ Player(String name)
PerfectMazeConfiguration
+ PerfectMazeConfiguration(
rowCount,
columnCount,
start,
goal,
thiefPenalty,
thiefFrequency,
goldFrequency,
goldAmount,
isWrappingMaze,
randomSeed)
RoomMazeConfiguration
+ RoomMazeConfiguration(
rowCount,
columnCount,
start,
goal,
thiefPenalty,
thiefFrequency,
goldFrequency,
goldAmount,
isWrappingMaze,
targetEdgeCount,
randomSeed)
<<Interface>>
Game
+ getPlayer(): Player
+ getMaze(): Maze
+ getPath(): Path
+ getScore(): int
+ isOver(): boolean
+ movePlayer(Direction dir): boolean
+ start(): void
MazeGame
+ MazeGame(Player p, Maze m)


<<Interface>>
Path
+ reachesTarget(): Boolean
+ totalGold(): int
+ getTarget(): Coordinates
+ getCoordinatesTraversed(): Coordinates
+ addCoordinates(Coordinates c): Path
+ enter(Node node)
<<Enum>>
Direction
NORTH
EAST
SOUTH
WEST
<<Interface>>
Coordinate
+ getX(): int
+ getY(): int
MazeCoordinate
+ MazeCoordinate(x, y);
MazePath
- target
- totalGold
- nodes: List<MazeNode>
+ MazePath(Cell target)
<<Interface>>
Node
+ grow(Configuration configuration): Configuration
+ loot(int gold): int
+ getGoldCount(): int
+ isThiefRoom(): boolean
+ isGoldRoom(): boolean
+ isDeadEnd(): boolean
+ isGoal(): boolean
+ getThiefPenalty();
+ getCoordinates(): Coordinates
+ setNode(Node node, Direction dir): void
+ getNode(Direction dir): Node
+ canReach(Coordinates coordinates): boolean
+ canReachHelper(Path path): boolean
+ wealthiestPathTo(Coordinates coordinates): Path
+ wealthiestPathToHelper(Path path): Path
+ exploreTo(Coordinates coordinates): Path
+ exploreToHelper(Path path): Path
+ pathTo(Coordinates coordinates): Path
+ pathToHelper(Path path): Path
+ get(Coordinates coordinates): Node
+ getHelper(Path path): Node
<<Interface>>
Edge
+ getTail(): MazeNode
+ getHead(): MazeNode
Wall
- tail: MazeNode
- head: MazeNode
DeadEndNode
<<Interface>>
Maze
+ getPlayer(): Player
+ getCurrent(): Node
+ move(Direction dir): boolean
+ pathTo(): Path
+ exploreTo(): Path
+ wealthiestPath(): Path
2DMaze
- start: Node
- current: Node
- goal: Node
+ Maze(start, goal)
StandardRoomNode
+ StandardRoomNode(coordinates)
+ loot(int gold)
+ isGoldRoom(): boolean
+ isThiefRoom(): boolean
GoldRoomNode
+ GoldRoomNode(coordinates, goldCount)
+ loot(int gold)
+ isGoldRoom(): boolean
+ isThiefRoom(): boolean
ThiefRoomNode
+ ThiefRoomNode(coordinates, goldCount)
+ loot(int gold)
+ isGoldRoom(): boolean
+ isThiefRoom(): boolean
AbstractRoomNode
# north: Node
# south: Node
# east: Node
# west: Node
# goldCount: int
# isGoal: boolean
# thiefPenalty: double
+ AbstractRoomNode(coordinates, goldCount, thiefPenalty)