

A Qualitative Model for Reasoning about Path and Support*

Abhishek Jaiswal, Zoe Falomir

Umeå University, Sweden

{abhishek.jaiswal,zoe.falomir}@umu.se

Abstract

Spatial reasoning abilities correlate strongly with performance in STEM fields. Games offer a compelling medium for training these critical skills in developing children who have a natural proclivity for play. However, to facilitate human-like tutoring and player guidance, these games require an AI agent capable of making commonsense inferences from spatial events. Qualitative reasoning (QR) models appear to be a suitable framework for these application domains. As these models reason in symbolic representations, they can seamlessly translate game states into interpretable feedback for human-like player guidance. This paper introduces a hybrid qualitative model designed for Camelot Jr., a block-puzzle game that requires constructing multi-level bridges to connect two avatars stationed on separate towers. The game poses a challenge for the player, who must make platforms stable, plan their path, and ensure they use all the provided blocks. To handle the precise physics required by the domain, we integrate a mathematical center-of-mass stability logic to guide our qualitative solver. Our work facilitates spatial skill training in Camelot Jr. and contributes to the development of human-centric, explainable game-playing agents.

1 Introduction

Qualitative Spatial and Temporal Representations and Reasoning (QSTR) [Cohn and Renz, 2007; Ligozat, 2011; Sioutis and Wolter, 2021] models and reasons about *time* (i.e., coincidence, order, concurrency, overlap, granularity) and also about properties of *space* (i.e. topology, location, direction, proximity, geometry, intersection, etc.) and their evolution in time between continuous neighboring situations. Maintaining the consistency and constraints in space and time are the basics in qualitative reasoning when solving spatial problems (i.e. path finding, orientation, relative position, etc.) and temporal problems (i.e. constraint satisfaction, schedule optimisation, precedence) [Guesgen and Bhatt, 2009]. As a result, well-defined qualitative models and reasoning techniques have appeared in the literature which can deal with imprecise and incomplete knowledge on a symbolic level. Spatio-temporal reasoning has been proved to be successful in many areas and applications such as: robotics [Mast *et al.*, 2016; Falomir *et al.*, 2011, 2013], computer vision [Suchan, 2022], ambient intelligence [Falomir

and Olteşanu, 2015], architecture and design [Li and Schultz, 2024; Bhatt and Freksa, 2015], geographic information systems [Wei *et al.*, 2024; Clementini and Cohn, 2024; Moratz *et al.*, 2025], safety analysis [Grigoleit *et al.*, 2017], autonomous driving [Monsen *et al.*, 2025] and education [Kragten *et al.*, 2024; Santamarta *et al.*, 2023]. The most relevant calculi in Qualitative Spatial Reasoning (QSR) was reported by Dylla *et al.* [2017].

Furthermore, qualitative representations are believed to be closer to the cognitive domain, as shown in cognitive models of sketch recognition [Lovett *et al.*, 2006], spatial problem solving tasks (i.e. visual oddity tasks) [Lovett and Forbus, 2011], paper folding reasoning tests [Falomir, 2016; Falomir *et al.*, 2021], analogy in Raven’s Progressive Matrices [Lovett and Forbus, 2017], logic composition of qualitative shapes when solving cognitive tests [Falomir *et al.*, 2020; Pich and Falomir, 2018], modeling 2D spatial abstraction during mental rotation tasks [Lovett and Schultheis, 2014], solving 3D rotation tasks [Ejarque and Falomir, 2026], solving the cube comparison test (CCT) [Falomir *et al.*, 2026; Falomir, 2025].

Therefore, novel models which combine QSTR, cognitive spatial thinking and common sense are a challenge which envision further advances in Artificial Intelligence and its applications.

Moreover, spatial cognition studies have shown that there is a strong link between spatial abilities and success in Science, Technology, Engineering and Math (STEM) disciplines [Newcombe, 2010; Wai *et al.*, 2009]. For example, children at 4 years old have already informal awareness of spatial relations such as parallel relations for two dimensional shape identification and description before they are properly taught about them [Sinclair *et al.*, 2013]. For this reason, researchers in US and Canada study the actualities and possibilities of spatial reasoning in contemporary school mathematics [Sinclair and Bruce, 2014], also because spatial learning and reasoning can be taught easily using visual and kinetic interactions offered by new digital technologies [Highfield and Mulligan, 2007].

In cognitive psychology, games like *Upside Down World* are used to evaluate spatial skills of students who are challenged to recreate buildings composed of multilink cubes in their upright orientation and use spatial language to describe the composition of the buildings to their colleagues to build accordingly [Sinclair and Bruce, 2014]. Also, a test of the German Academic Foundation to find out children with gifted brains consists in finding out the consistent view/projection for a 3D object usually corresponding to a technological drawing¹. In previous works by the same author, a qualitative model for 3D object description was

*Correspondence to: Abhishek Jaiswal. E-mail: abhishek.jaiswal@umu.se

¹Test der Studienstiftung: Gehirnjogging für Hochbegabte, see *Spiegel Online*: <http://www.spiegel.de/quiztool/quiztool-49771.html>

developed and promising results are obtained [Falomir, 2015].

QSR methods present a unique opportunity to develop game-playing agents that follow a human-centric logic. While this paradigm requires frameworks grounded in symbolic representations, existing research on spatial games like *Angry Birds* suggests the need to complement pure symbolic abstractions by mathematical knowledge of physical phenomena Haase and Wolter [2021]. In this work, we investigate a hybrid spatial reasoning framework using the bridge-building puzzle game *Camelot Jr.* (illustrated in Figure 1), where the core objective is to construct a continuous path connecting a starting position to a goal point. To address this, we propose a solver model grounded in human-like commonsense logic that combines qualitative reasoning modules with physical stability calculations.

Concretely, this paper explores the challenge of formulating a model that can serve two objectives:

- Help people understand how to solve games involving spatial stability and path planning, so that they can improve their spatial skills and therefore enhance their success in STEM; and also,
- Equip artificial intelligence agents with the symbolic and physical reasoning mechanisms to solve spatial problems in wooden puzzle games so that they can reason about the spatial stability and path finding.

The rest of the paper is organized as follows. Section 2 outlines the specifics of the *Camelot Jr.* game. Section 3 introduces the Qualitative Descriptor model for the game. Section 4 describes the game constraints for the qualitative model to comply with. Section 5 details the numerical centre-of-mass (COM) stability logic. Section 6 describes the graph traversal logic used by the agent in the game. Section 7 explains the path-finding algorithm using the previously discussed ingredients. Section 8 concludes with a discussion on the current scope and future directions.

2 The Camelot Jr. Game: Background

This section presents the various game elements, how they interact, and the rules governing their behavior.

2.1 Game Elements and Definitions

All gameplay in *Camelot Jr.* occurs on a vertical 2D **Grid** that represents the castle layout, where the **Ground** serves as the base of the grid and acts as the primary immutable support for all structures. The core task revolves around two animated characters, the **Knight** (K) and the **Princess** (P), who must be connected by a path comprising various block pieces across the layout. The environment contains fixed **Towers** (T), which are solid, permanent structures positioned at the start of a challenge. To bridge structural gaps, the player deploys pieces from a limited inventory of **Blocks** (B), consisting of rectangular building blocks that serve as stable, walkable platforms and stair-blocks, which combine a solid block with a ramp side that allows diagonal movement up or down. The ramp side occupies exactly one grid cell. The rectangular blocks can be of sizes 1 to 3. The stair-block has one ramp side combined with rectangular blocks of size one or two (see Figure 2 for sample stair-block orientations).

The entire environment is governed by a global **Gravity** constraint, ensuring that every placed block structure is physically viable under static equilibrium to be considered a valid solution. Together, these elements dictate the available **Walkable Space**, defined as the set of empty grid cells located immediately above stable structures or the ground. Crucially, for stair-blocks, the

ramp face itself constitutes the active walkable surface rather than the empty grid cell above it. A schematic overview of the game is illustrated in Figure 1.

2.2 Game Mechanics and Objectives

The primary objective in *Camelot Jr.* is to use a limited set of blocks to construct a path for the Knight to reach the Princess. This task is governed by three environmental constraints. First, no game element can extend beyond the castle grid. Second, towers stay fixed; they cannot be moved or rotated. Third, every inventory block given at the start must be used in the final solution.

Beyond these operational boundaries, the player must follow the rules of structural placement and stability. The support rule requires that every block must rest on solid ground, a fixed tower, or another stable block. Under the global stability rule, any assembly of inventory blocks must remain viable under gravitational torque without toppling. The inclined side of a stair-block cannot support another block (i.e., no block can be placed on top of the ramp for stability). However, if a horizontal stair-block is inverted with its flat bottom on top (*HNI* and *HMI* in Figure 2), it provides a valid horizontal support surface. A valid solution gives a walkable path, defined as a sequence of connected grid cells, from the Knight to the Princess.

3 Qualitative Description

3.1 Grid Environment ($\mathcal{G}$)

The grid is a set of cells $c_{x,y}$ where $x \in [0, W)$ and $y \in [0, H)$:

$$c_{x,y} = \begin{cases} 0 & \text{Empty Space} \\ 1 & \text{Tower } (T) \\ id \geq 10 & \text{Placed Block } (B) \end{cases}$$

$K = (x_k, y_k)$ (Knight) and $P = (x_p, y_p)$ (Princess).

3.2 Block Definition

A block B is defined as a 4-tuple representing its intrinsic properties and current state:

$$B = \{N, id, \mathcal{O}, \chi\}$$

Where:

- $N \in \mathbb{Z}^+$ is the **Scalar Size**, representing the length of the block segment.
- $id \in \mathbb{Z}$ is the unique identifier used for tracking grid occupancy and enforcing placement order.
- $\mathcal{O}$ is the **Orientation State Vector**, stores the particular orientation of the block to be used for grid placement.
- $\chi \in \{0, 1\}$ is the **Stair Indicator**, defining whether the block possesses ramp side ($\chi = 1$) or is a standard flat block ($\chi = 0$).

A placement is a mapping $L : B \rightarrow (x, y)$, where (x, y) represents the anchor coordinate (the bottom-left cell) of the block in the grid $\mathcal{G}$. We define $B_{x,y}$ as the block instance whose anchor point is situated at (x, y) .

3.3 The Orientation State Vector

The vector $\mathcal{O} = (\text{Axis}, f, \iota)$ governs the spatial projection and parity of the block via three qualitative operations:

- **Axis** $\in \{H, V\}$: Determines if the block is Horizontal or Vertical.

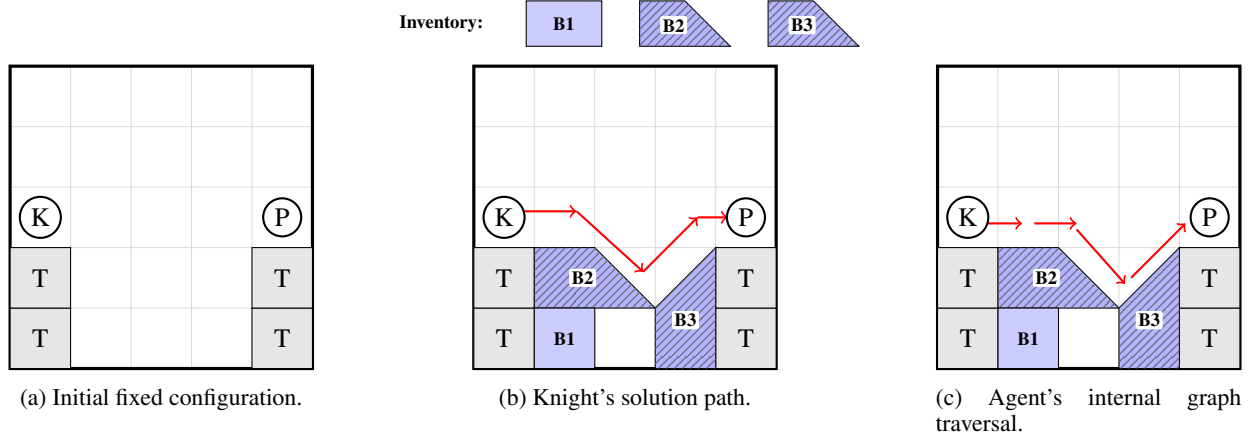


Figure 1: Qualitative discrete representation of a *Camelot Jr.* game challenge layout: (a) initial problem state, (b) a stable solution path with stair-blocks placed in horizontal and vertical orientations, and (c) the agent's internal path traversal graph. The agent's moves are: (*move-right*), (*move-right + move-diagonal-down*), and (*move-diagonal-up*).

Label	State Name	(Axis, f , ι) if $\chi = 1$	Functional Affordance
HN	Horizontal Normal	$(H, 0, 0)$	Flat structural support + Downward ramp from left to right
HM	Horizontal Mirror	$(H, 1, 0)$	Flat structural support + Upward ramp from left to right
HNI	Horizontal Inverted	$(H, 0, 1)$	Flat structural support
HMI	Horizontal Mirror Inverted	$(H, 1, 1)$	Flat structural support
VN	Vertical Normal	$(V, 0, 0)$	Upward ramp from left to right
VM	Vertical Mirror	$(V, 1, 0)$	Downward ramp from left to right
VNI	Vertical Normal Inverted	$(V, 0, 1)$	Inverted Pillar (Illegal placement)
VMI	Vertical Mirror Inverted	$(V, 1, 1)$	Inverted Pillar (Illegal placement)
RB	Rectangular Solid Block	$\chi = 0$	Flat structural support

Table 1: Qualitative orientation labels and the supported affordances for inventory blocks

- **Reflection** ($f \in \{0, 1\}$): Governs horizontal parity (Mirroring).
- **Inversion** ($\iota \in \{0, 1\}$): Governs vertical parity (Flipping).

The solver navigates the orientation space using three discrete operations:

- **Reflect** (R): A Horizontal Flip (Reflection across the vertical axis),
- **Invert** (I): A Vertical Flip (Reflection across the horizontal axis),
- **Transpose** (T): A 90° Anti-Clockwise Rotation.

Table 1 describes the various orientations for a stair-block, along with the affordance of the orientations. Figure 2 shows a schematic of the various ramp orientations. Horizontal Normal refers to the base orientation of a stair-block with the ramp occupying the rightmost size (1st diagram in Figure 2).

For a rectangular block denoted as **RB**, only two distinct orientations exist - Horizontal and Vertical and we use only the Axis label to define them.

3.4 State Transition Matrix

Table 2 defines how orientation changes through various qualitative operations for a stair-block. The solver utilizes this to explore valid placements during backtracking. For rectangular blocks only to-and-fro Horizontal-Vertical state transitions exist. From subsequent discussion, we consider only the legal placements of stair-blocks.

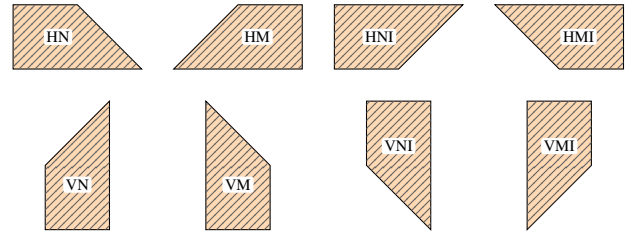


Figure 2: The eight qualitative orientation states for a stair-block ($N \geq 2$). Arrangements like VNI and VMI are illegal as per the game rules.

4 Block Placement Constraints for the Solver

4.1 Spatial Restrictions

The effective dimensions (w_e, h_e) are emergent properties derived from the Scalar Size N and the **Axis** state:

$$(w_e, h_e) = \begin{cases} (N, 1) & \text{if Axis} = H \\ (1, N) & \text{if Axis} = V \end{cases}$$

A placement $L : B \rightarrow (x, y)$ is valid only if every grid cell c within the block's effective dimensions (w_e, h_e) is currently empty (value = 0).

$$\forall j \in [0, w_e), \forall i \in [0, h_e) : c_{x+j, y+i} = 0$$

This rule ensures that blocks do not collide with fixed Towers or previously placed inventory items. Additionally, the placement must satisfy the Boundary Constraint:

$$x + w < W \quad \text{and} \quad y + h < H$$

Initial State	Reflect (R)	Invert (I)	Transpose (T)
HN	HM	HNI	VN
HM	HN	HMI	VNI (Illegal)
HNI	HMI	HN	VM
HMI	HNI	HM	VMI (Illegal)
VN	VM	VNI (Illegal)	HMI
VM	VN	VMI (Illegal)	HM

Table 2: State Transition Matrix for qualitative block orientations. Pruned states are marked as Illegal.

4.2 Foundational Restrictions

To optimize the search process, the solver evaluates a block’s immediate foundational support before committing to a placement. For any block $B_{x,y}$ not at ground level ($y > 0$), there must be at least one valid point of support contact with the structure below. Support is validated by inspecting the cells directly beneath the block’s footprint. For these supporting blocks, we have:

For $j \in [0, w_e]$ and $(x_{\text{sup}}, y_{\text{sup}}) \in c_{x+j, y-1}$,
ValidSupport($x_{\text{sup}}, y_{\text{sup}}$) = True if:

1. $c_{x_{\text{sup}}, y_{\text{sup}}} \neq 0$ i.e., $x_{\text{sup}}, y_{\text{sup}}$ must be part of a Block B_{sup}
2. $\chi = 0$ i.e., B_{sup} is not a stair-block.
3. If $B_{\text{sup}} \in \{\text{HNI}, \text{HMI}\}$ (i.e., $\chi = 1$ and $\iota = 1$), it is an inverted stair-block providing flat structural support.
4. If $B_{\text{sup}} \in \{\text{HN}, \text{HM}, \text{VN}, \text{VM}\}$ (i.e., $\chi = 1$ and $\iota = 0$), the support is invalid if $(x_{\text{sup}}, y_{\text{sup}})$ is the non-inverted ramp cell, i.e.,
$$(x_{\text{sup}}, y_{\text{sup}}) = \begin{cases} (x_s + N - 1, y_s) & \text{if label}(B_{\text{sup}}) = \text{HN} \\ (x_s, y_s) & \text{if label}(B_{\text{sup}}) = \text{HM} \\ (x_s, y_s + N - 1) & \text{if label}(B_{\text{sup}}) \in \{\text{VN}, \text{VM}\} \end{cases}$$

where (x_s, y_s) denotes the structural anchor (the bottom-left reference cell) of the supporting block B_{sup} , and N represents the dimension length of the block.

This support logic works because any non-stair horizontal or vertical block allows walking over itself. Similarly, if the stair is inverted ($\iota = 1$), the block acts as a flat top surface, and support is valid throughout its width. For a normal stair-block, the support is valid as long as the cell $x_{\text{sup}}, y_{\text{sup}}$ is not the ramp cell of the stair-block.

5 Recursive Stability Formalization

The solver verifies structural integrity by iterating through all placed pieces in descending order of their vertical position (y). This top-down sequence follows real-world centre-of-mass (COM) physics and ensures that loads from higher blocks are propagated to those below.

Recursive Load Aggregation

For each block B_i in the set of placed pieces (sorted by descending y -coordinate), we define the total weight W_i and the resultant moment M_i :

$$W_i = m_i + \sum_{k \in \text{Loads}(i)} f_k$$

$$M_i = (m_i \cdot cx_i) + \sum_{k \in \text{Loads}(i)} (f_k \cdot pos_k)$$

The effective center of mass for the system including B_i is then:

$$COM_{res} = \frac{M_i}{W_i}$$

The Convex Hull Constraint

A block is considered stable if, and only if, its resultant center of mass is contained within the bounds of its support geometry:

$$s_{min} \leq COM_{res} \leq s_{max}$$

Where s_{min} and s_{max} represent the extreme horizontal boundaries of all supporting cells $(x, y - 1)$ in contact with B_i .

Bridge Distribution (The Lever Rule)

In cases of multiple supports (e.g., a bridge), the load W_i is distributed between the two outermost supports s_1 and s_2 . The force w_2 allocated to s_2 is determined by:

$$w_2 = W_i \cdot \frac{COM_{res} - cp_1}{cp_2 - cp_1}$$

Where cp represents the center point of the respective support. The remaining force $w_1 = W_i - w_2$ is transmitted to s_1 . This recursive propagation continues until the forces are absorbed by the immutable ground or towers.

6 Cell Affordances for Agent Actions

Once the entire block inventory is placed on the grid and verified as structurally stable, the agent attempts to discover a valid path connecting the Knight to the Princess. Starting from the Knight’s position, the agent operates within a Directed Acyclic Graph (DAG) framework, with the movement primitives *move-right*, *move-diagonal-up*, and *move-diagonal-down* (Table 3). The agent initiates a *move-right* motion from an idle position; however, depending on the structural affordances of the terrain where the agent lands, a *move-diagonal-up* or *move-diagonal-down* adjustment may follow automatically depending on where the agent lands.

Action Label	Coordinate Transition
<i>move-diagonal-up</i>	$(i, j) \rightarrow (i + 1, j + 1)$
<i>move-diagonal-down</i>	$(i, j) \rightarrow (i + 1, j - 1)$
<i>move-right</i>	$(i, j) \rightarrow (i + 1, j)$

Table 3: Permissible agent actions during path exploration from the Knight to the Princess.

Let the intentional horizontal target cell be $c_{x,y}$ at grid position $(x, y) = (i + 1, j)$, and let $(x_{\text{sup}}, y_{\text{sup}}) = (i + 1, j - 1)$ be the underlying support cell provided by block B_{sup} . Let (x_s, y_s) denote the bottom-left anchor point of the block under evaluation (i.e., the support block if the target cell is empty, or the block occupying the target cell otherwise). A transition is valid if it satisfies any of the following conditions:

1. **Flat Foundation** ($\Delta y = 0$): A horizontal move is valid if the cell is empty and the agent is supported by a flat boundary. This occurs if the support cell directly below is ground ($j - 1 = 0$), a tower, or a block with flat structural support. For inventory block support, we require: $c_{x,y} = 0 \wedge c_{x_{\text{sup}}, y_{\text{sup}}} \in \{\text{RB}, \text{HNI}, \text{HMI}\}$
2. **Stair Platform** ($\Delta y = 0$): A horizontal move is still valid if the agent walks along the flat surface of an up-right stair-block. Only HN and HM blocks present this spe-

cific combination of a flat walking surface and a ramp cell:
 $c_{x,y} = 0 \quad \wedge \quad c_{x_{\text{sup}},y_{\text{sup}}} \in \{\text{HN}, \text{HM}\}$ such that

$$(x_{\text{sup}}, y_{\text{sup}}) \neq \begin{cases} (x_s + N - 1, y_s) & \text{if label}(B_{\text{sup}}) = \text{HN} \\ (x_s, y_s) & \text{if label}(B_{\text{sup}}) = \text{HM} \end{cases}$$

3. **Geometric Descent** ($\Delta y = -1$): An automatic diagonal downward transition is triggered when the horizontal movement causes the agent to land over an empty cell whose underlying support footprint corresponds exactly to a downward-sloping ramp cell.
 $c_{x,y} = 0 \quad \wedge \quad c_{x_{\text{sup}},y_{\text{sup}}} \in \{\text{HN}, \text{VM}\}$ such that

$$(x_{\text{sup}}, y_{\text{sup}}) = \begin{cases} (x_s + N - 1, y_s) & \text{if label}(B_{\text{sup}}) = \text{HN} \\ (x_s, y_s + N - 1) & \text{if label}(B_{\text{sup}}) = \text{VM} \end{cases}$$

4. **Geometric Ascent** ($\Delta y = +1$): A diagonal upward transition automatically follows if the agent lands on the ramp cell of an ascending stair. This is the only case in which the landing cell for *move-right* is not an empty cell ($c_{x,y} \neq 0$), and validity requires the entry point to match the ramp cell.
 $c_{x,y} \neq 0 \quad \wedge \quad c_{x,y} \in \{\text{HM}, \text{VN}\}$ such that

$$(x, y) = \begin{cases} (x_s, y_s) & \text{if label}(B_{x,y}) = \text{HM} \\ (x_s, y_s + N - 1) & \text{if label}(B_{x,y}) = \text{VN} \end{cases}$$

Figure 1c illustrates the agent’s motion as it executes two consecutive *move-right* actions from the Knight’s initial position. It lands on a cell above the downward ramp, which automatically triggers a *move-diagonal-down* action. Subsequently, it lands on an upward ramp, triggering a *move-diagonal-up* action and enabling the agent to successfully reach the Princess. Although the agent’s path differs slightly from the Knight’s traced path due to stair-block traversal mechanics, both approaches follow the same block-placement structure.

7 Solver - the pathfinding agent

The agent’s solving algorithm is implemented using a recursive backtracking protocol. Algorithm 1 describes the step-by-step working of the agent. It searches the permutation space of the inventory $\mathcal{I}$ to identify a final structural arrangement that satisfies two independent evaluation criteria: *Static Stability* and *Path Connectivity*.

Mirroring a human player, the agent places inventory blocks one after another onto the grid layout, evaluates their structural validity once the inventory is fully deployed, and subsequently evaluates path connectivity. Similar to an intelligent player who intuitively rules out any piece configuration unlikely to reach the target destination, our solver implements geometric pruning strategies to cut off impractical search branches that cannot connect the start and the end points with the remaining inventory.

7.1 Search Criteria

To evaluate a candidate grid configuration $\mathcal{G}$, the solver verifies two primary conditions:

- **Static Stability:** This condition is satisfied if every placed inventory block conforms to the convex hull stability constraint (Section 5), verifying that no component of the structural assembly will tilt or collapse under gravity. Integrating this quantitative physics metric ensures that the qualitative model accurately respects the domain’s physical constraints.
- **Path Connectivity:** This condition is satisfied if the pathfinding module successfully extracts a continuous route from the Knight’s starting location K to the Princess’s

Algorithm 1 Recursive Backtracking Search for Stable Path Solutions

Require: Inventory $\mathcal{I}$, Current grid configuration $\mathcal{G}$, Placed blocks set $\mathcal{U}$

Ensure: A stable grid configuration with a valid path, or failure

```

1: if  $|\mathcal{U}| = |\mathcal{I}|$  then
   {Base Case: Inventory deployed}
2: if not IsEntireStructureStable( $\mathcal{G}$ ) then
3:   return false {Reject: unstable structure}
4: end if
5:  $Path \leftarrow \text{FindValidPath}(\mathcal{G})$ 
6: if  $Path \neq \emptyset$  then
7:   SaveSolution( $\mathcal{G}, Path$ )
8:   return true {Solution confirmed}
9: end if
10: return false {Reject: unreachable goal}
11: end if
12:  $B \leftarrow \text{SelectUnusedBlock}(\mathcal{I} \setminus \mathcal{U})$ 
13: for each Orientation  $O \in \text{GetOrientations}(B)$  do
14:   for each Position  $(x, y) \in \text{GridCoordinates}$  do
15:     if not HasValidSupportFootprint( $\mathcal{G}, x, y, O$ ) then
16:       continue {Prune if no support from below}
17:     end if
18:     if CanPlaceBlock( $\mathcal{G}, x, y, O$ ) then
19:        $\mathcal{G} \leftarrow \text{ApplyPlacement}(\mathcal{G}, B, x, y, O)$ 
20:        $\mathcal{U} \leftarrow \mathcal{U} \cup \{B\}$ 
21:       if SolveRecursive( $\mathcal{I}, \mathcal{G}, \mathcal{U}$ ) = true then
22:         return true {Propagate success upstream}
23:       end if
24:       {Backtrack: Revert state if path or stability fails}
25:        $\mathcal{G} \leftarrow \text{RemovePlacement}(\mathcal{G}, B)$ 
26:        $\mathcal{U} \leftarrow \mathcal{U} \setminus \{B\}$ 
27:     end if
28:   end for
29: end for
30: return false {Exhausted branches without success}

```

destination cell P based on sequential spatial affordances (Section 6).

7.2 Search Protocol and Backtracking

The solver employs a recursive backtracking approach to systematically explore the valid state space:

1. **Sequential Placement:** Blocks are selected one-by-one from the inventory $\mathcal{I}$ and assigned a local coordinate position via the mapping $L : B \rightarrow (x, y)$ alongside an orientation choice $\mathcal{O}$. Each block can serve as the starting point and all block orientations are tested recursively. Each potential placement enforces spatial Exclusion Rules and Foundational Restrictions (Section 4).
2. **Inventory Exhaustion:** The recursive engine continues branching until every asset within the inventory set has been successfully allocated to the grid.
3. **Global Validation:** Once all the inventory is used, the agent verifies the center-of-mass (COM) stability of the complete assembly. It is necessary to check stability only after placing all blocks, as sometimes counterweights can stabilize an initially unstable arrangement.
4. **Path Verification:** If the final configuration is completely stable, the agent invokes the grid-walking graph to evaluate connectivity between K and P via sequential cell affordances (Section 6).

5. **State Reversion:** If the agent fails to satisfy any of the criteria previously described, it backtracks to a previous state, clearing the cell occupancy for the most recent node allocation ($c_{x,y} \rightarrow 0$), and tests for other alternative branches.

7.3 Termination Criteria

The search terminates and yields a complete solution when both static stability and path connectivity are satisfied. The search can continue further to find multiple solutions, depending on the specific requirements of the agent solver.

8 Discussion and Future Work

This work presents a hybrid qualitative solver for the puzzle game *Camelot Jr.*, which requires understanding of structural stability and spatial path connectivity. Our solver agent leverages a symbolic paradigm to investigate the game state using common-sense logic, formalizing environmental interactions through qualitative structural affordances. Because the game mechanics involve complex physical equilibria, we integrate this symbolic layer with a numerical center-of-mass (COM) stability logic. By fusing these qualitative and numerical ingredients, the agent successfully solves arbitrary configuration challenges within the game's problem space.

We envisage several use cases and extensions for this kind of solving methodology. First, the framework can be used to train human spatial reasoning, specifically regarding structural stability and path visualization. Because the agent's decision-making process is inspired by human-centric thinking, the system can serve as a core engine for an intelligent tutoring system, providing meaningful, human-like hints tailored to a player's current solution attempt. Experiments with students in STEM fields, engaging with such interactive teaching tools, can be an effective educational aid.

Second, the solver's qualitative behavior opens up opportunities to investigate the specific structural and combinatorial parameters that increase task difficulty from a human perspective. By tracking metrics such as placement options, inventory count, and orientation and pruning efficiency across various levels, we can develop metrics that quantify how humans navigate spatial path-building games.

Building upon this foundational setup, we plan to extend this hybrid paradigm to other physical puzzle games of a similar nature. Investigating how qualitative action affordances and rigorous physical constraints interact across different game environments will help uncover more generalized logics of human-like game-playing and tutoring, providing a clear roadmap for future inquiry.

Acknowledgments

We acknowledge the funding by the Wallenberg AI, Autonomous Systems and Software Program (WASP) awarded by the Knut and Alice Wallenberg Foundation, and the funding by *Stiftelsen J.C. Kempes och Seth M. Kempes minne*, Sweden.

References

- M. Bhatt and C. Freksa. Spatial computing for design: an artificial intelligence perspective. In John S. Gero, editor, *Studying Visual and Spatial Reasoning for Design Creativity*, pages 109–127. 2015.
- Eliseo Clementini and Anthony G. Cohn. Extension of rcc*-9 to complex and three-dimensional features and its reasoning system. *ISPRS Int. J. Geo Inf.*, 13(1):25, 2024.
- A. G. Cohn and J. Renz. *Qualitative Spatial Reasoning, Handbook of Knowledge Representation*. Elsevier, Wiley-ISTE, London, 2007.
- Frank Dylla, Jae Hee Lee, Till Mossakowski, Thomas Schneider, André van Delden, Jasper van de Ven, and Diedrich Wolter. A survey of qualitative spatial and temporal calculi: Algebraic and computational properties. *ACM Comput. Surv.*, 50(1):7:1–7:39, 2017.
- Marti Ejarque and Zoe Falomir. Applying the qualitative model for reasoning about 3d perspectives (q3d) to solve a spatial test about rotations. In *International Workshop on Qualitative Reasoning (QR'26), co-located at the 35nd Joint International Conference on Artificial Intelligence (IJCAI)*, Bremen, Germany, August 2026.
- Z. Falomir and A.-M. Oltețeanu. Logics based on qualitative descriptors for scene understanding. *Neurocomputing*, 161:3–16, 2015.
- Z. Falomir, E. Jiménez-Ruiz, M. T. Escrig, and L. Museros. Describing images using qualitative models and description logics. *Spatial Cognition and Computation*, 11(1):45–74, 2011.
- Z. Falomir, L. Museros, V. Castelló, and L. Gonzalez-Abril. Qualitative distances and qualitative image descriptions for representing indoor scenes in robotics. *Pattern Recognition Letters*, 38:731–743, 2013.
- Zoe Falomir, Albert Pich, and Vicent Costa. Spatial reasoning about qualitative shape compositions. *Ann. Math. Artif. Intell.*, 88(5-6):589–621, 2020.
- Zoe Falomir, Ruben Tarin, Aurelio Puerta, and Pablo Garcia-Segarra. An interactive game for training reasoning about paper folding. *Multim. Tools Appl.*, 80(5):6535–6566, 2021.
- Zoe Falomir, Ramon Garcia-Pou, and Vicent Costa. A qualitative model to reason about object rotations – applied to solve the Cube Comparison Test. 2026. Under Review.
- Zoe Falomir. A qualitative model for reasoning about 3d objects using depth and different perspectives. In T. Lechowski, P. Wallega, and M. Zawidzki, editors, *LQMR 2015 Workshop*, volume 7 of *Annals of Computer Science and Information Systems*, pages 3–11. PTI, 2015.
- Z. Falomir. Towards a qualitative descriptor for paper folding reasoning. In *Proc. of the 29th International Workshop on Qualitative Reasoning*, 2016. Co-located with IJCAI'2016 in New York, USA.
- Zoe Falomir. A qualitative model to reason about object rotations (QOR) applied to solve the cube comparison test (CCT). *ArXiv preprint arXiv:2601.08382*, 2025.
- Florian Grigoleit, Sebastian Holei, Andreas Pleuss, Robert Reiser, Julian Rhein, Peter Struss, and Jana von Wedel. The qsafe project - developing a tool for current practice in functional safety analysis. In Marina Zanella, Ingo Pill, and Alessandro Cimatti, editors, *28th International Workshop on Principles of Diagnosis (DX'17)*, Brescia, Italy, September 26-29, 2017, Kalpa Publications in Computing, pages 297–312. EasyChair, 2017.
- H. Guesgen and M. Bhatt, editors. *Proc. of the Workshop on Spatial and Temporal Reasoning*, Pasadena, California, USA, 2009. International Joint Conference on Artificial Intelligence (IJCAI).
- Felix Haase and Diedrich Wolter. Behind the corner: Using qualitative reasoning for solving angry birds. 2021.

- K. Highfield and J. Mulligan. The role of dynamic interactive technological tools in preschoolers' mathematical patterning. In J. Watson and K. Beswick, editors, *Proc. of the 30th annual conference of the Mathematics Education Research Group of Australasia*, volume 1, pages 372–381. MERGA, 2007.
- Marco Kragten, Tessa Hoogma, and Bert Bredeweg. Integration of a teacher dashboard in a hybrid support approach for constructing qualitative representations. In Rafael Ferreira Mello, Nikol Rummel, Ioana Jivet, Gerti Pishtari, and José A. Ruipérez-Valiente, editors, *Technology Enhanced Learning for Inclusive and Equitable Quality Education - 19th European Conference on Technology Enhanced Learning, EC-TEL 2024, Krems, Austria, September 16-20, 2024, Proceedings, Part I*, Lecture Notes in Computer Science, pages 208–221. Springer, 2024.
- Beidi Li and Carl Schultz. Clingo2dsr - a clingo-based software system for declarative spatial reasoning. *Spatial Cognition & Computation*, 0(0):1–51, 2024.
- G. Ligozat. *Qualitative Spatial and Temporal Reasoning*. MIT Press, Wiley-ISTE, London, 2011.
- A. Lovett and K. Forbus. Cultural commonalities and differences in spatial problem-solving: A computational analysis. *Cognition*, 121(2):281 – 287, 2011.
- A. Lovett and K. Forbus. Modeling visual problem solving as analogical reasoning. *Psychological Review*, 124(1):60 – 90, 2017.
- Andrew M. Lovett and Holger Schultheis. Modeling spatial abstraction during mental rotation. In Paul Bello, Marcello Guarini, Marjorie McShane, and Brian Scassellati, editors, *Proceedings of the 36th annual conference of the Cognitive Science Society*, pages 886 – 891. Cognitive Science Society, 2014.
- A. Lovett, M. Dehghani, and K. Forbus. Learning of qualitative descriptions for sketch recognition. In *Proc. 20th Int. Workshop on Qualitative Reasoning (QR), Hanover, USA, July, 2006*.
- Vivien Mast, Zoe Falomir, and Diedrich Wolter. Probabilistic reference and grounding with PRAGR for dialogues with robots. *J. Exp. Theor. Artif. Intell.*, 28(5):889–911, 2016.
- Julius Monsen, Jakob Suchan, and Mehul Bhatt. Probabilistic answer set programming driven ranking of dynamic space-time belief models. In *Rules and Reasoning: 9th International Joint Conference, RuleML+RR 2025, Istanbul, Turkey, September 22–24, 2025, Proceedings*, page 156–175, Berlin, Heidelberg, 2025. Springer-Verlag.
- Reinhard Moratz, Niklas Daute, James Ondieki, Markus Kattenbeck, Mario Krajina, and Ioannis Giannopoulos. Bilateral spatial reasoning about street networks: Graph-based RAG with qualitative spatial representations. *CoRR*, abs/2512.15388, 2025.
- N. Newcombe. Picture this: Increasing math and science learning by improving spatial thinking. *American Educator*, 34(2):29–35, 2010.
- Albert Pich and Zoe Falomir. Logical composition of qualitative shapes applied to solve spatial reasoning tests. *Cogn. Syst. Res.*, 52:82–102, 2018.
- Vicent Santamarta, Pablo Garcia-Segarra, Noelia Ventura-Campos, Aida Moreno-Rus, and Zoe Falomir. Reversal error in mathematics: Training against it using €8404 coach. In Ismael Sanz, Raquel Ros, and Jordi Nin, editors, *Artificial Intelligence Research and Development - Proceedings of the 25th International Conference of the Catalan Association for Artificial Intelligence, CCIA 2023, Món Sant Benet, Spain, 25-27 October 2023*, Frontiers in Artificial Intelligence and Applications, pages 241–249. IOS Press, 2023.
- N. Sinclair and C. D. Bruce. Spatial reasoning for young learners, Research Forum. In *Proc. of the 38th Conference of the International Group for the Psychology of Mathematics Education and the 36th Conference of the North American Chapter of the Psychology of Mathematics Education (PME 38 / PME-NA 36)*, pages 173–205, Vancouver, Canada, July 2014.
- N. Sinclair, E. de Freitas, and F. Ferrara. Virtual encounters: the murky and furtive world of mathematical inventiveness. *ZDM. The International Journal on Mathematics Education*, 45(2):239–252, 2013.
- Michael Sioutis and Diedrich Wolter. Qualitative spatial and temporal reasoning: Current status and future challenges. In *IJCAI*, pages 4594–4601, 2021.
- Jakob Suchan. *Declarative reasoning about space and motion in visual imagery - theoretical foundations and applications*. PhD thesis, Bremen University, Germany, 2022.
- J. Wai, D. Lubinski, and C. P. Benbow. Spatial ability for STEM domains: Aligning over 50 years of cumulative psychological knowledge solidifies its importance. *Journal of Educational Psychology*, 101(4):817–835, 2009.
- Lijun Wei, Valérie Gouet-Brunet, and Anthony G. Cohn. Location retrieval using qualitative place signatures of visible landmarks. *Int. J. Geogr. Inf. Sci.*, 38(8):1633–1677, 2024.