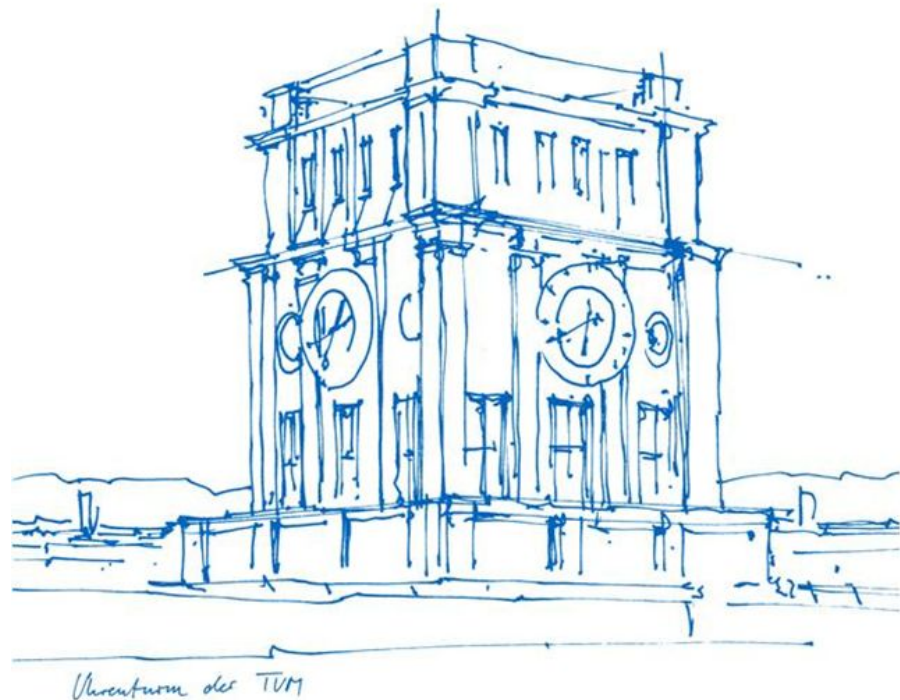


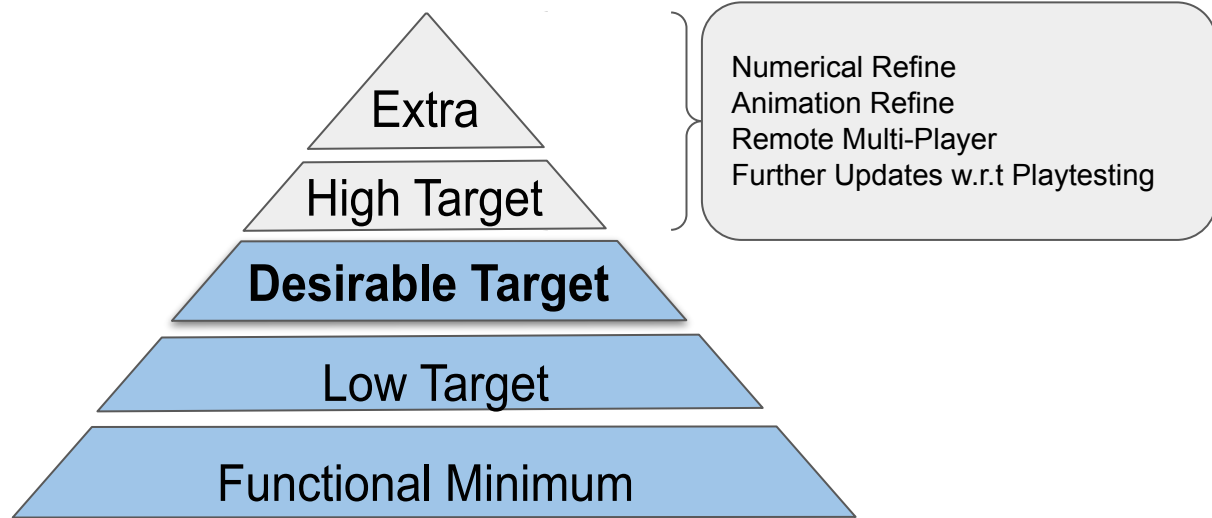
Alpha Release Rebomb

Yaxuan Dai
Mahdis Sabzevarzadeh
Miguel Trasobares
Jialin Yang

Technical University of Munich
TUM School of Computation, Information and Technology
Chair of Computer Graphics and Visualization
Garching 08.01.2025



1 Task Progression



Timeline Update

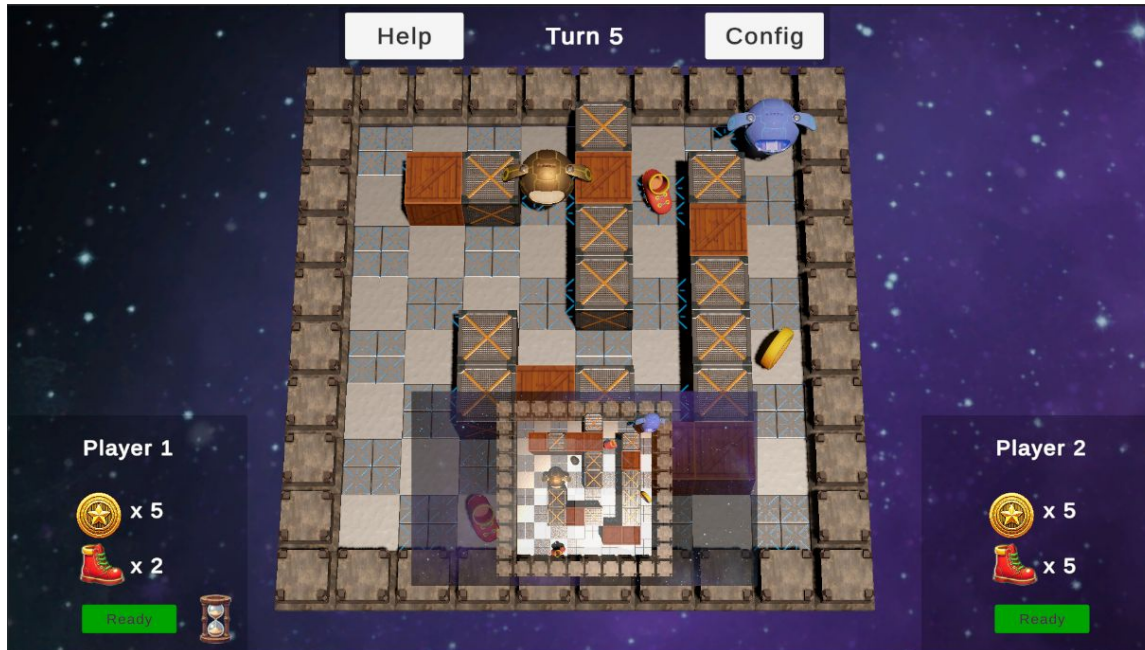
Date	Milestone	Week	Layer	Task	Time		Owner	State
					Expect	Actual		
Nov 27-Dec 03	Interim demo	4		time travel mechanism 2/2	8	10	Jialin	DONE
				cascaded explosion refine	8	5	Yaxuan	DONE
				specifical assets	8		Miguel	Later
			Desirable	map generation 1/3	8	8	Mahdis	DONE
				map manager refactoring (new)	3	1	Mahdis	DONE
				local multiplayer 1/3	8	10	Yaxuan	DONE
				version integration	4 * 2	4 * 2	All	DONE
Dec 04-10	5	Desirable	map generation 2/3	8	8	Mahdis	DONE	
			audio effects (new)	8	8	Mahdis	DONE	
			explosion visual effects 1/2	8	4	Yaxuan	DONE	
			explosion refactoring (new)	3	4	Yaxuan	DONE	
			preview time travel (new)	8	12	Jialin	DONE	
			more weapon and objects	8	6	Jialin	DONE	
			version integration	4 * 2	4 * 2	All	DONE	
			map generation 3/3	6	6	Mahdis	DONE	
Dec 11-17	6	Desirable	explosion visual effects 2/2	6	6	Yaxuan	DONE	
			numerical refine	6	0	All	Later	
			version integration	4 * 4	4 * 4	All	DONE	
			specifical visual/audio assets	8	12	Mahdis	DONE	
Dec 18-24	7	Desirable	turn&level logic (refine)	6	6	Yaxuan	DONE	
			bugfix before integration			All	DONE	
			version release	3 * 4	3 * 4	All	DONE	
Dec 25-31	8				0			
Jan 01-07	Alpha release	9		holiday(Dec 24-Jan 06)		0		
Jan 08-14		10	High	remote multiplayer	2 * 8		Yaxuan	
				assesment & bugfix	2 * 8			
				version integration	4 * 4		All	
Jan 15-21	Playtesting	11	High	remote multiplayer	2 * 8		Yaxuan	
				improve mechanics	8			
				refine animation/characters	8			
				version integration	4 * 4		All	
Jan 22-28		12		remaining tasks & test	4 * 10		All	
Jan 29-Feb 04	Final release	13	Extra	extra tasks & test	4 * 10		All	

2 Challenges

Map Generation | Audio Effects | Time Travel Preview | Visual Effects

2.1 Time Travel Preview

- Hourglass holders need to know the result of the time travel before using the item

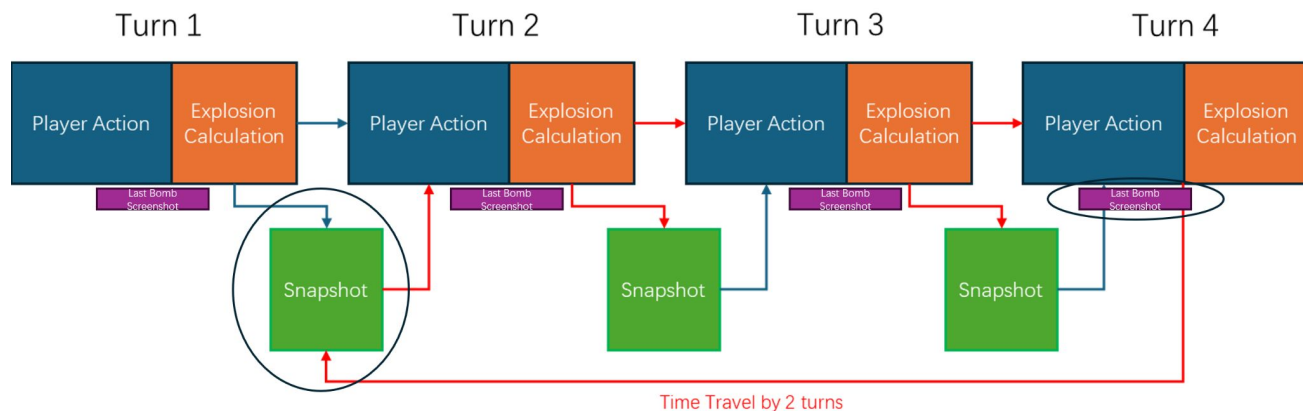


2.1 Time Travel Preview

Initial plan: Duplicating scene

- High memory and computation power cost
- Deep copies required
- “Hidden bombs” invisible

Solution: Combining snapshot images with real-time last bomb rendering results



2.2 Visual and Audio Elements

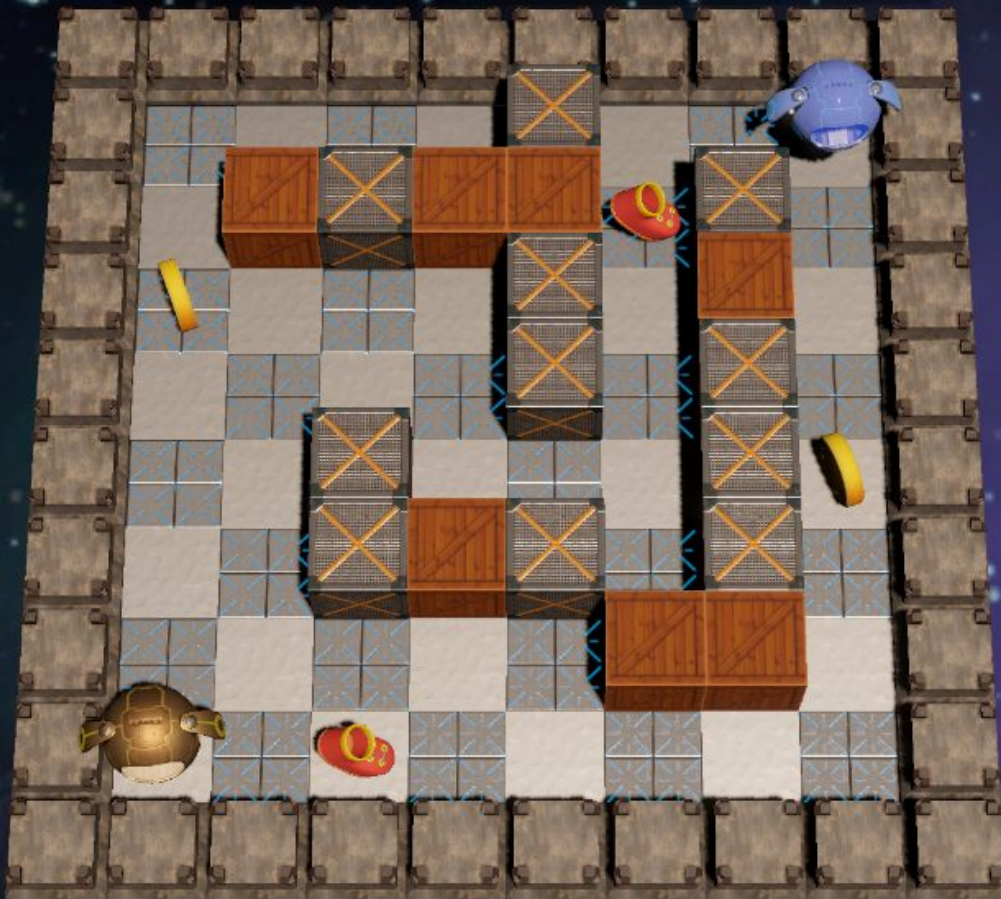
- **Game World Vision:**
 - Players are robots battling in a space-themed environment.
 - Assets were chosen to align with this futuristic, robotic theme.

- **Sources of Assets:**
 - Audio & Visuals: Materials used from Unity Asset Store, Sketchfab and Pixabay
 - Custom Icons: AI-generated icons were used for UI elements like the inventory system

Help

Turn 1

Config



Player 1



Ready

Player 2



Ready

2.3 Map Generation

- Using ***Random Walk Algorithm*** for procedurally generating the map
- **Map Structure:**
 - Divided into accessible (tunnels) and inaccessible (walls) areas
- **Initial Map Generation:**
 - A map with unbreakable walls is created based on given dimensions.
 - A 'Walker' is randomly placed, replacing walls with floors to create tunnels
- **Tunnel Digging Algorithm:**
 - Random direction (up, down, left, right) and length are chosen.
 - The 'Walker' continues digging tunnels and updating its position until the desired number of tunnels is created.

2.3 Map Generation

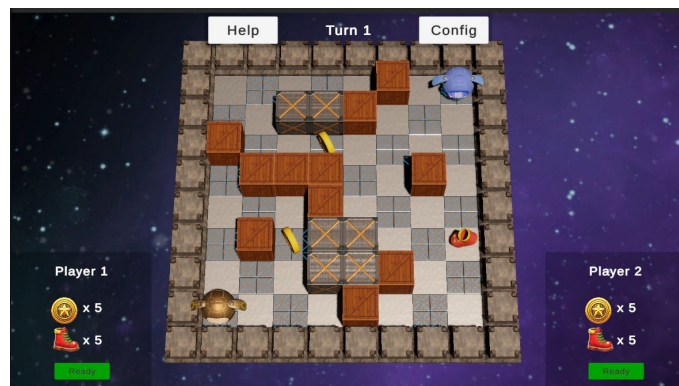
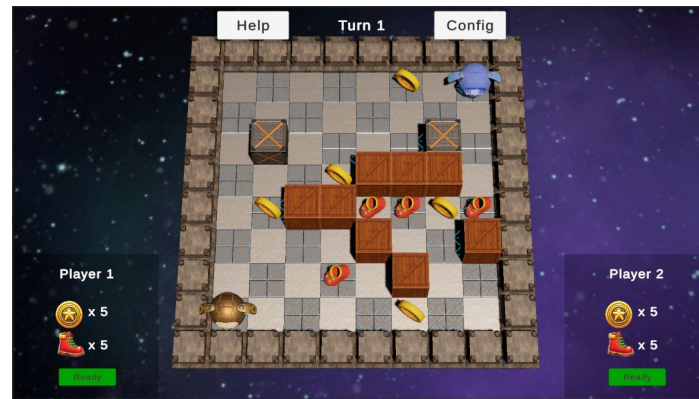
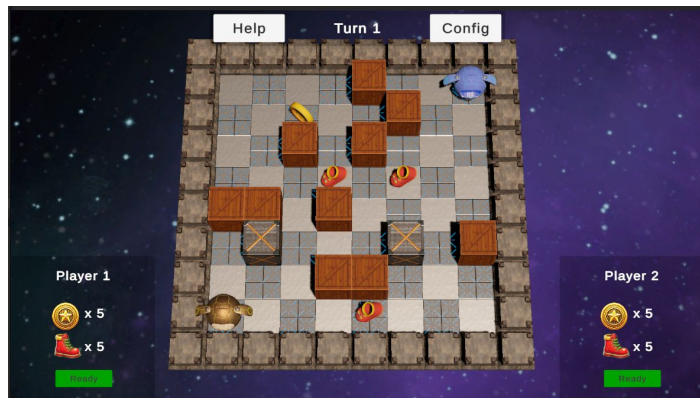
- **Player Placement:**
 - Players are placed in corners.
 - Maps with unbreakable walls in corners are rejected.

- **Optimization:**
 - Maximum attempts and tunnel length are adjusted to prevent endless loops.
 - Trials identified optimal values for fast, reliable map generation without crashes

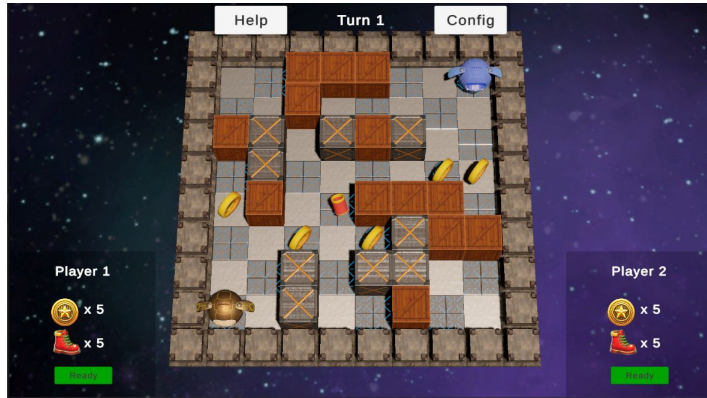
2.4 Breakable Wall and Items Placement

- **Adding Breakable Walls:**
 - Floors are replaced with breakable walls, allowing players to walk through them after breaking.
 - A random probability determines if a floor is replaced with a breakable wall.
 - Starting probability resets or increases based on the random number.
- **Placing Items (Coins, Boots, Hourglass):**
 - Items (coin and boots) are placed randomly using the same probability logic.
 - Item type (coin or boot) is chosen randomly.
 - Uneven distribution of items encourages strategic gameplay.
 - **Hourglass** is placed under a randomly selected breakable wall.
 - The position is determined by creating a list of all breakable walls and selecting one randomly.

Generated Map Examples

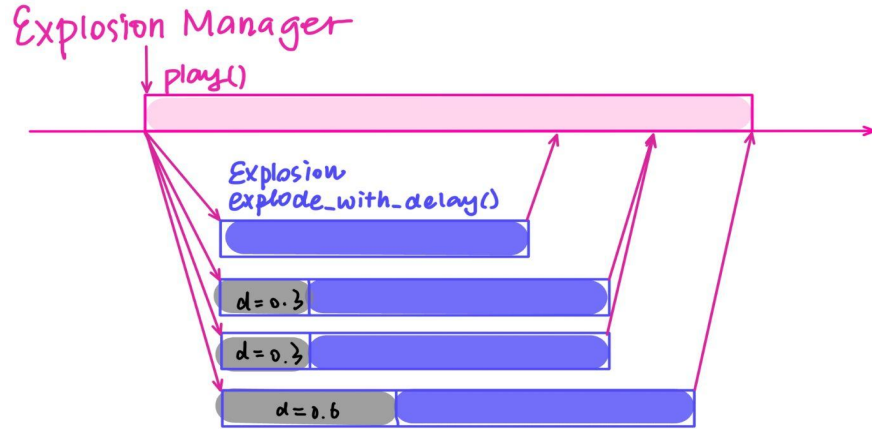


Generated Map Examples



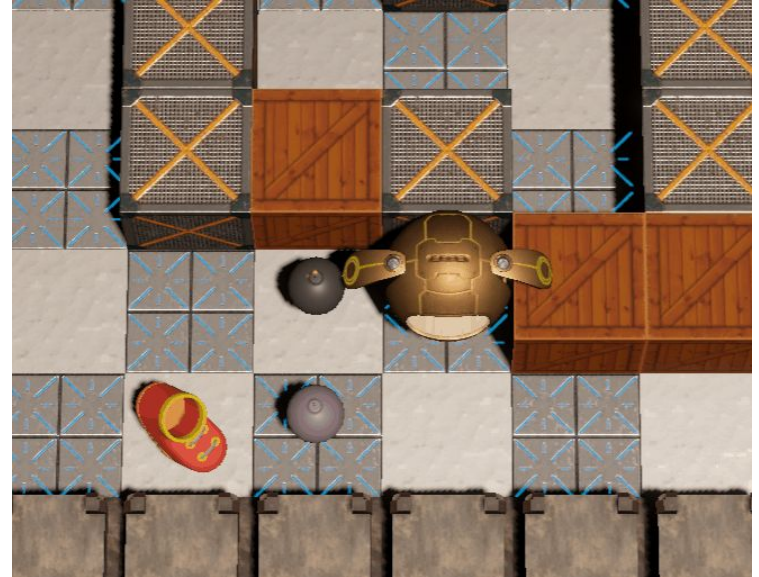
2.5 Visual Effect

Delay Control with Coroutine



Effect Control with Programmable VFX

- Flame Decay
- Distinct Visuals

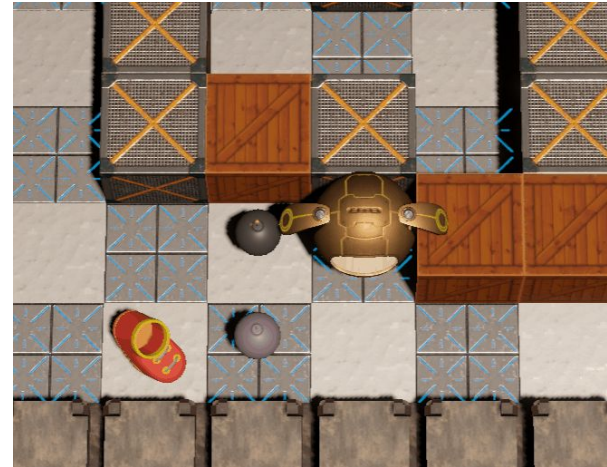
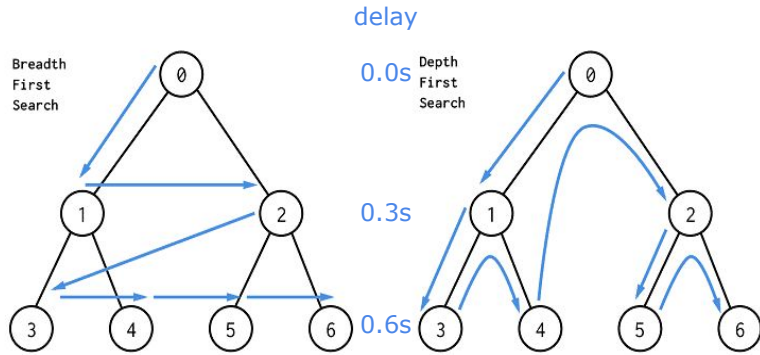


3 Design Revision

Cascaded Explosion Refinement | Other Refactorings

3.1 Cascaded Explosion Refinement

- Logic decouple:
 - Bomb: trigger & cascaded influence
 - Explosion: delay, range, visual and audio effects
- Cascaded trigger logic: DFS→BFS (for ChainBomb)
- Trigger raycasting with layer config (for SafeBomb)



3.2 Other Refactoring

- Consistent Resource Management
 - reference of objects → total resource number
- Centralized Bomb Configuration
 - Appearance: `bomb_prefab`, `explosion_prefab`
 - Basic attribute: `explosion_range`, `explosion_turn`, `price_in_coins`
 - Extra feature: `bomb_type`
- Map Generation
 - Initial Approach: Numerical Values (e.g 0 for floors, 1 for unbreakable walls etc) were used to represent map elements in function
 - Problem: the straightforward method became soon difficult to manage as constraints increased
 - Solution: Introducing constants `FLOOR`, `UNBREAKABLE_WALL`
- Item Placement
 - Initial Approach: Separate functions were created for placing primary items like coins and boots. Differentiation between coins and boots
 - Refactored the structure of adding items, for an easy addition of new item types. Eliminate the need to differentiate between item types. From a pool of items we randomly generate one, on a designated item placement position.

Thank you for your time!

Looking forward for your feedback!