Foo Rescue V1.0

Goal

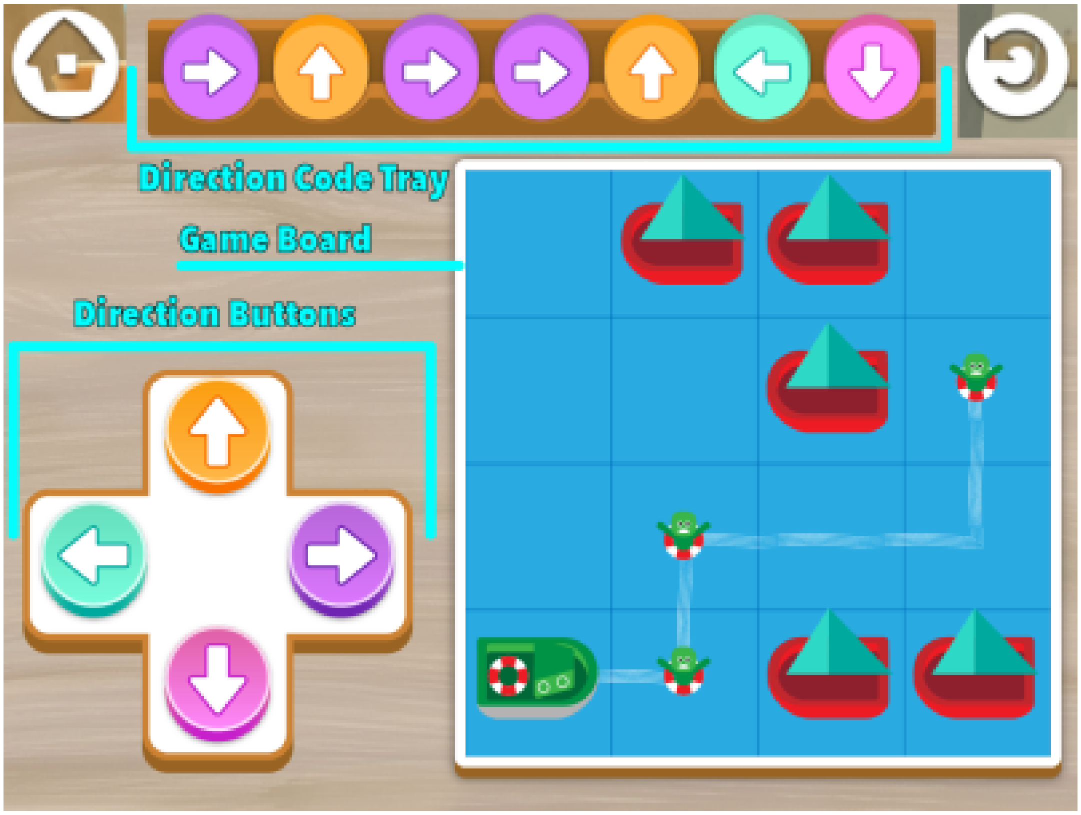
Our goal for Pre-K puzzles is to adapt learning objectives from our codeSpark puzzles to better fit a younger audience.

Layout

The layout of the game will consist of 3 main sections.

- Direction buttons which players interact with
- A code tray which stores the directions the player selected
- A game board

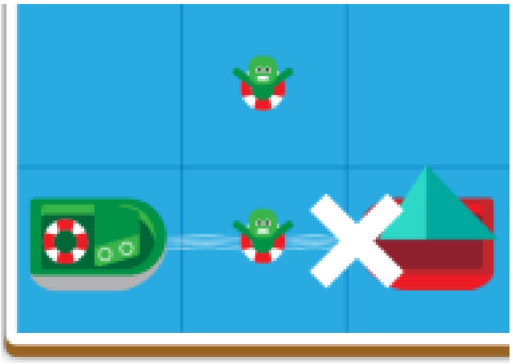
The gameboard, direction buttons, and direction code tray will all look like pieces of a real board game. They will be modeled in 3D and will respond to camera movements.



Direction Buttons

In the Pre-K version of puzzles players will use a 4 button interface similar to videogame D-Pads to input the sequence of movements their avatar will follow.
When players press on a direction button the corresponding direction arrow token slides out of the D-Pad and moves into the left most spot within the direction code tray at the top and a guide line appears which shows where the combination of arrows will lead the avatar to. If the arrows lead the avatar into a wall or other obstacle, the guide line will stop there and a visual indicator will signal that the avatar can not proceed.

Ghost guide blocked by obstacle:



Early levels will omit the code tray and allow players to move the avatar one space at a time using the arrow buttons. This would help scaffold players into the experience, allowing them to get used to the controls before having to solve puzzles. A ghost guide would not be necessary during this mode.

Direction Code Tray

The direction code tray stores the combination of directions entered by the player which governs how the avatar will move around the game board. Players add to the direction code tray by pressing the direction buttons in the D-Pad area. Players may remove arrows from the tray either by dragging the arrow outside the tray or by pressing on them then pressing again to confirm deleting them. Removing any arrow other than the last in the tray (the furthest right) causes every subsequent arrow to shift left and fill the empty position. Players can drag code tiles over another tile to swap places.

Deleting Code:

Code will shake after first press and delete after second. If player taps elsewhere on the screen the code deletion will be canceled.

Game Board

The game board provides a visual reference for the player's goal for each puzzle. The game board consists of:

- An avatar
  - The piece the player is controlling with direction arrow code
- A grid of movable spaces
  - Each square = 1 unit of movement
  - Grid is always 4x4
- Obstacles
  - Art which indicates the player will not be able to pass through a specific space
- Collectibles
  - The player must collect all to proceed to the next level
  - Pieces are removed from the board when the avatar makes contact with them
    - For the Tugboat design we can add the foos that are picked up to the boat
    - For the Kaiju design the buildings can be moved to the left of the board, above the direction buttons
- A guideline
  - Art provides a preview of where the avatar will travel based on the arrows in the code tray
  - If the arrows lead the avatar into a wall or obstacle a visual indicator will signal the player can not proceed that way
  - We may want to have the ability to turn hints off on later levels

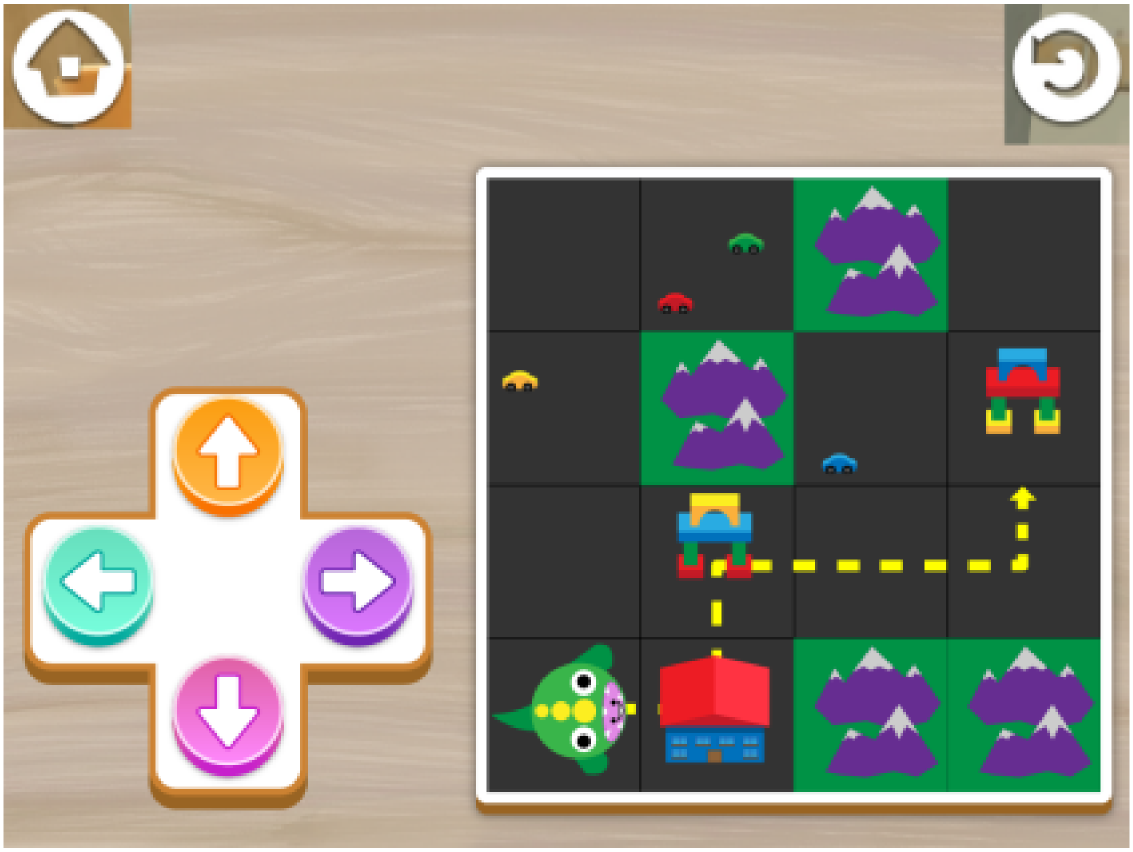
Game Intros

Pre-K puzzles will have 2 different intros, one at the start of a pack of levels, and a shorter version which plays at the start of each level.

The intro which plays at the start of a pack of levels will begin with a black screen then fades in while the camera zooms out. The camera starts at a low angle focused on a folded up board, then pulls up and repositions to reveal the entire board, the code tray, and the direction buttons as the board unfolds (similar to camera movement in this video). After zooming out the obstacles, collectibles, and the player avatar all drop into place.

Each level will start with a black screen then fades in as the camera zooms out. The camera starts zoomed in on the place where the avatar starts and zooms out as the avatar drops into place.

Direct Control Alternative



We are starting the player out with direct control over their avatar using the direction buttons, then introducing the direction code tray after a few levels.

This method allows players to experiment around with the controls to help them get used to them. This helps ease players in, allowing them to build an understanding of how the buttons affect the avatar.

Having 2 different functionalities for the same buttons is a potential risk. To address this we will utilize visual language and narration to signal the shift in functionality. These cues include:

- No code tray on direct control levels
- Direction code tiles shoot out of the D-Pad during indirect control level, but not during indirect control levels

If the player taps a direction button, then taps another before the avatar has finished moving to their previous destination; the avatar will immediately move to the first destination, then start their animation for the follow up destination.

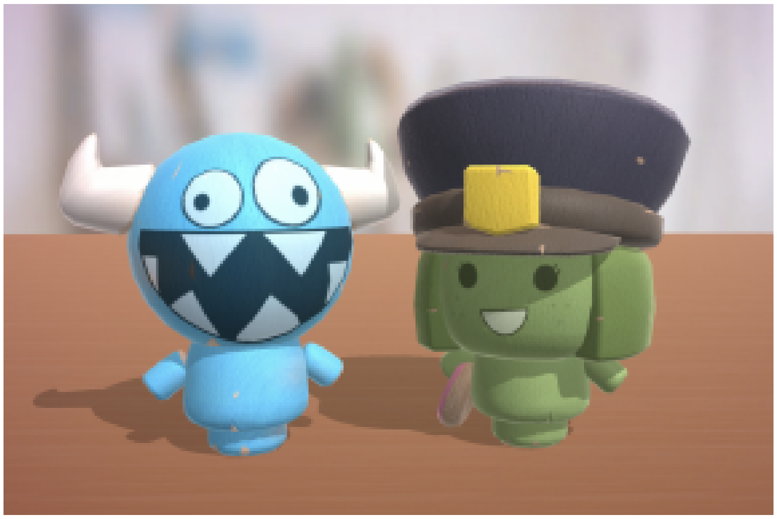
First time on Indirect (coding) controls

If our playtests show that our audience is struggling when switching to indirect control for the first time, we can make adjustments to the affordances the player is given to make the puzzle solution clearer.

- **Limit buttons on D-Pad:** One option we have is to limit which buttons are available to interact with. If a puzzle only requires up and right to complete, we would remove the interaction for down and left and update the visuals to help communicate this.
- **Limit space in the code tray:** Another constraint we could implement is adjusting the length on the code tray to only allow the shortest possible path. Knowing the exact number of spaces required to solve a puzzle helps make the solution clearer and allows for a smaller cognitive load on the player.

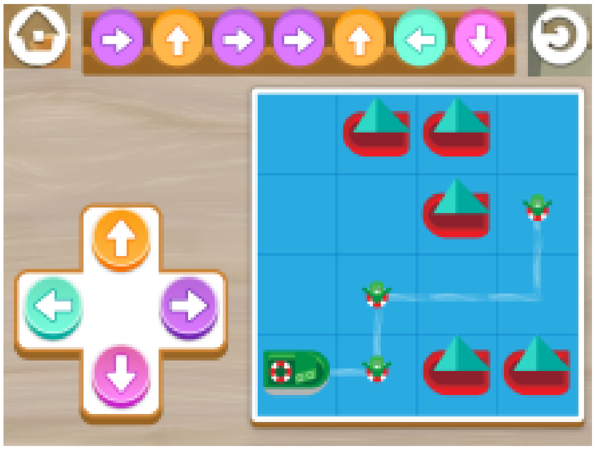
Aesthetics

Characters

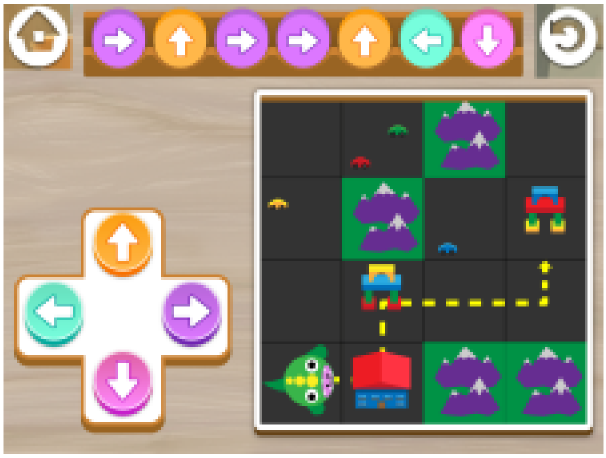


The visuals of the puzzles will combine Fooville with the idealized realism that we strive for in the rest of our Pre-K app, represented as wooden peg versions of the Foos. This will allow us to reuse aspects of the existing puzzles, like the stories we show, while feeling both aesthetically unique from the prime game and cohesive with the other Pre-K sections.

Themes/Stories



Tugboat to Toyland

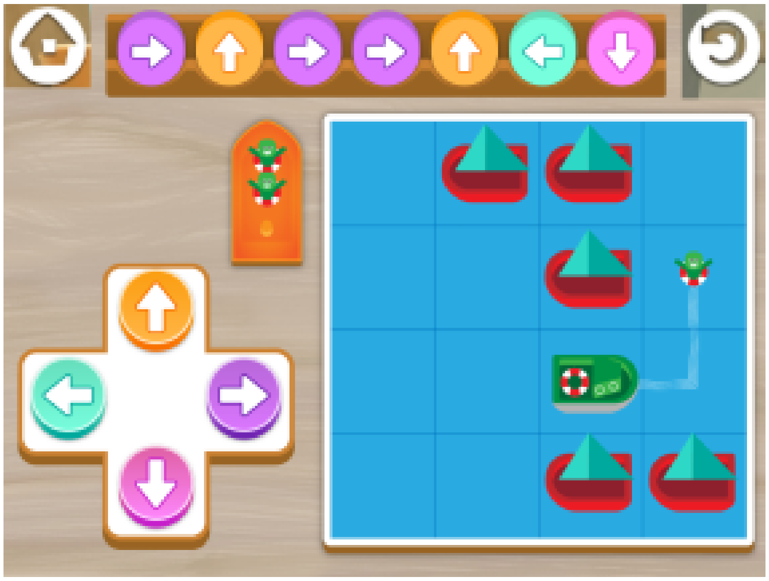


Kaiju Demolition Crew

Tugboat to Toyland

You are a tugboat captain and your job is to ferry foos from Fooville to Toyland. Avoid boats and other obstacles while picking up stranded Foos along the way. When Foos get rescued, they move off the board and are placed near the top left corner of the board (where they won't come into contact with tiles that slide out of the D-Pad).

Foos get collected in a lifeboat on the side of the board.



Obstacles

For the obstacles the system will randomly select between islands with and without lighthouses on them. To keep the randomization in check, we will put constraints on how the lighthouses will appear. A more ideal constraint is to have the lighthouses only appear once per land mass (collection of touching obstacle tiles), if that implementation is too costly we can instead restrict each level to only contain 1 lighthouse.

- Easier implementation: Only 1 lighthouse can appear per level
- More difficult implementation: Only 1 lighthouse can appear per land mass

Story (comic panels at start)

Panel 1: Tug boat heading into port to pick up Foos (extreme wide shot with tug boat in foreground and Fooville in background.)

Panel 2: Foos waiting in line with luggage

Panel 3: Tidal wave casting a shadow over Foos (extreme wide shot with Foos on one side and wave on the other)

Panel 4: Foos struggling in the water

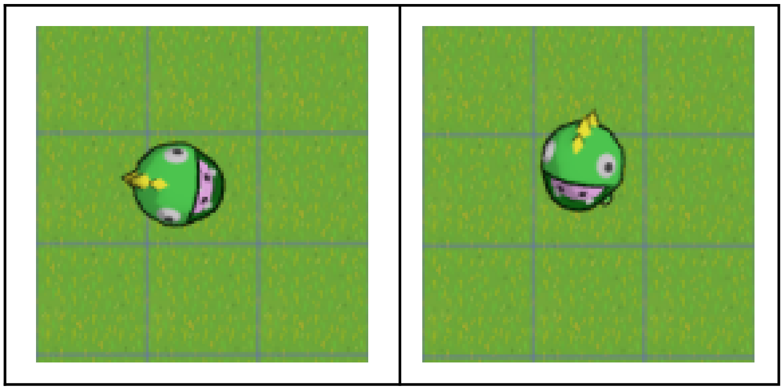
Panel 5: Close up of the boats speed set to "Super Speed"

Panel 6: Birds eye view of boat headed toward Foos in the water

You are Tom Tom, the Toyland tugboat captain, and you were on your way to pick up Foos when a giant wave washed them away (maybe we can have the Glitch surfing on the wave). Now you need to ride around rescuing the Foos on your way back to Toyland.

Kaiju Demolition Crew

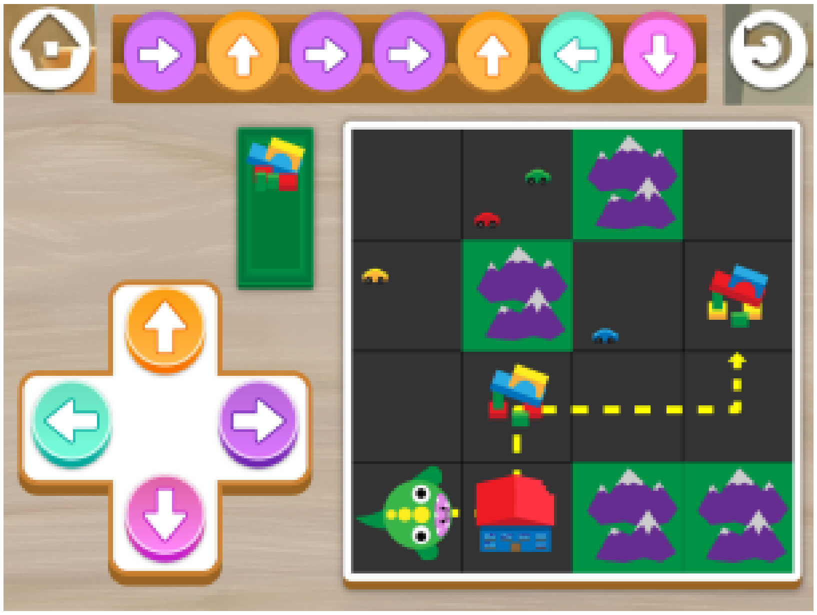
To be implemented in V2



Overhead view of dragon costume Foo

You are a Kaiju dinosaur (our dragon costume Foo) and your job is to demolish some buildings to make way for new ones. Run around the town, avoiding mountains and other obstacles to smash dilapidated buildings.

Collecting demolished buildings



Story (comic panels at start)
Panel 1: Woz in front of dilapidated building
Panel 2: Silhouette of monster behind building

Panel 3: Monster hand with business card being given to a scared Woz

Panel 4: Woz shaking Kimiko's (monster) finger

Panel 5: Kimiko smashing a building in the background with Woz giving the thumbs up in foreground

You are Kimiko, a kaiju that's trying to start up a demolition business, and you just got your first gig in Toyland. There's some major reconstruction going on and they need some of their buildings brought down fast. Make your way through Toyland while smashing as many buildings as you can to grow your demolition business.

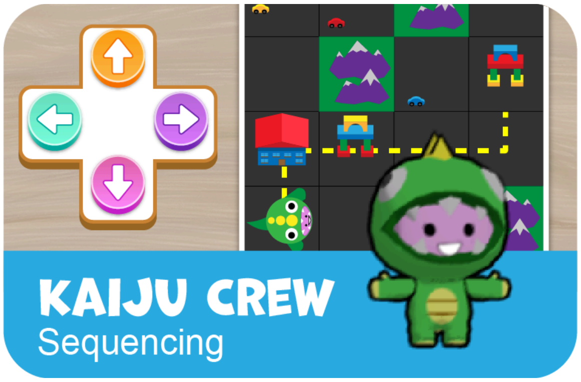
Board Game Look

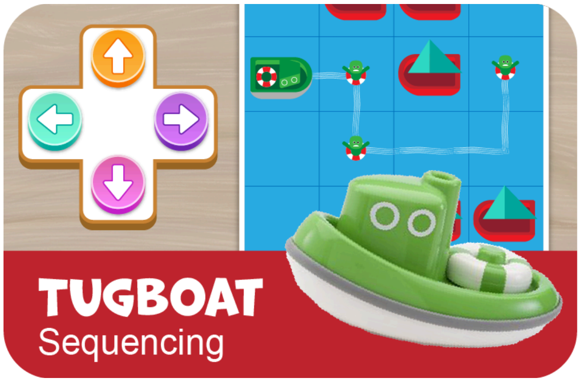
We chose a board game look for our environments for its strong connection to the type of unit based movement the player will work with. Board games also tie in well with our character direction, as they look like game pieces. The board game should appear high end, with a high quality wooden board and painted wooden pieces preferable over plastic ones.

Menu Tiles

The menu tiles for Pre-K puzzle games will follow the same general style as our other puzzle games. In the content section (top 2/3) the left side will show the direction input buttons and the right side will show the game board. Having the direction input buttons consistently displayed on the tile will help our Pre-K puzzles standout from our prime puzzles.

The hero art in the bottom right corner is temporary, for the final we will want something that captures the story more. The look of the art should follow our idealized realism style.





Levels

Levels will be collected into 3 different difficulties. When the player taps the button they will be given 5 random levels from a curated list organized by difficulty based on their previous performance. This mirrors the way we're designing our micro-games, grouping content and delivering a unique experience each play through.

For the initial release, we will create between 20-30 puzzles.

If the players quits a level pack before finishing it, the next time they return they will start a new level pack.

Player Feedback

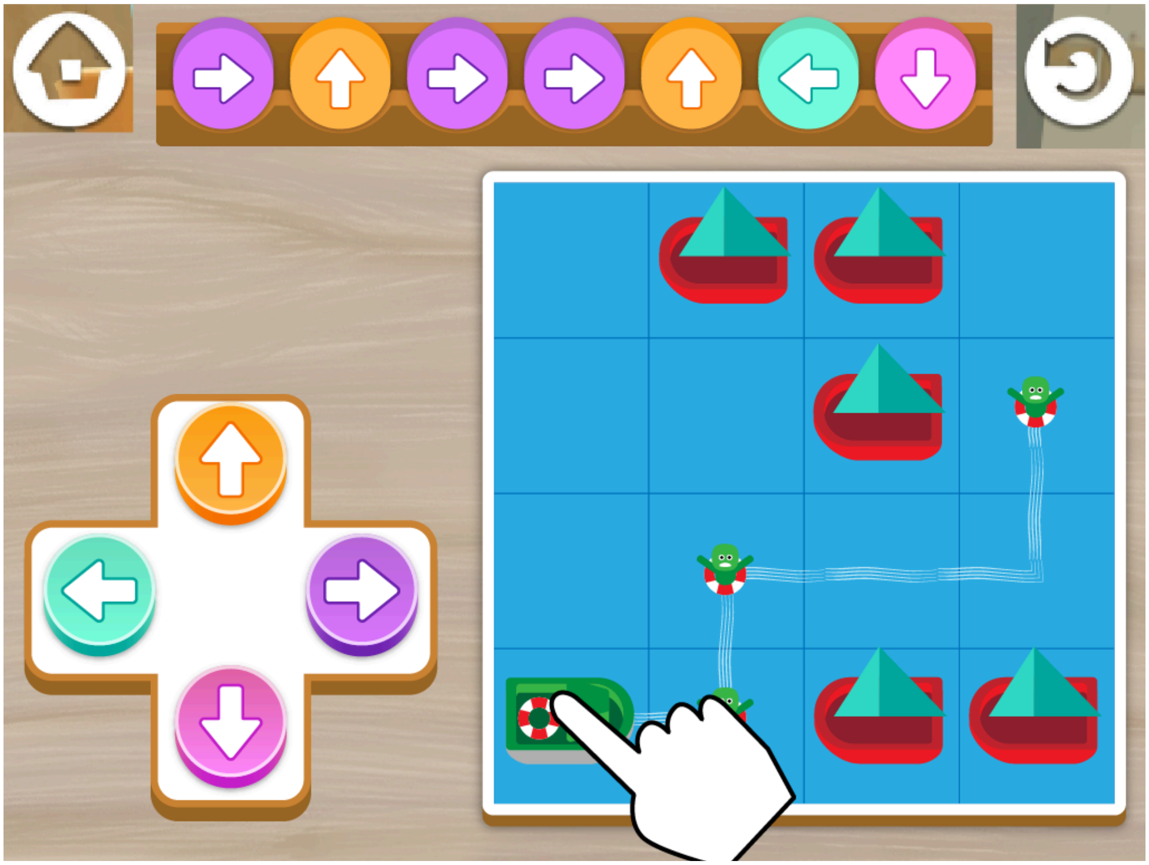
Unsure Where to Tap

If the player taps anywhere on the screen that isn't a button, the game draw attention to the buttons they should be interacting with by having them shake.

Ghost Guide

To help ease players into the concepts we will provide a ghost guide (similar to what we do in Splash Clash) which shows them where the avatar will end up depending on the movement they've assigned the blocks. The player's movement will be represented by a linear graphic stretching from the avatar to their ending point. If the player attempts to walk through a solid surface, the guideline will stop there and a visual indicator will signal that the avatar can not proceed.

Confirming the Sequence



Players will need to tap on the avatar to confirm their answer. Along with teaching this interaction in our early levels, we will also provide feedback when the player taps on any non-interactive elements on the screen. This mirrors what we currently do in codeSpark prime's puzzles, which allows us to reuse existing code and art.

If the player is idle for 3 or more seconds the following will occur:

- If there are no code tiles in the top tray only the D-Pad will shake
- If the code tray is partially full the D-Pad and the avatar will shake
- If the code tray is completely full only the avatar will shake

Scoring/Evaluating

Win State

The win state for Pre-K puzzles will closely mirror what we currently have designed in codeSpark prime. The steps are as follows:

1. The direction arrows and code tray are removed from the screen (they get cropped out when the camera zooms in)
2. The camera zooms in to frame the avatar
3. The avatar performs a short celebration animation
4. Celebratory music plays (we can reuse what we have in prime puzzles)
5. UI slides in with 2 buttons replay and next

The win state will trigger after the player achieves all their goals, even if the player has direction arrows remaining in the code tray.

Fail States

Players will have unlimited attempts to get the puzzle correct. If the player inputs incorrect direction arrows the avatar will execute the sequence until they hit an obstacle or wall, or run out of directions. From there the game will provide the following feedback based on how many times they've failed the level:

- The first time a player fails the level they will receive an auditory feedback letting them know it is a mistake. The reset button will scale pulse (like it does in codeSpark prime puzzles) and when the player taps on it the avatar will return to the starting position. The direction arrows will remain in the code tray, and they will shake to indicate the player can press on them, which will cause them to be removed from the tray. If the player presses on either a button or a direction arrow in the code tray the shaking will stop.
- The second time a player fails the level all feedback from the first failure will occur and if the ghost guide is disabled we enable it.
- All subsequent failures will repeat the feedback from the first failure.

Restarting

When restarting a round, whether triggered by the player or automatic, the process should be extremely streamlined to get the player back to the puzzle as quickly as possible.

Things that shouldn't occur on replay:

- Intro animations should not play
- Scene should not reload

Things that should occur on replay:

- All board game objects (avatar and collectibles) will slide back to their starting positions
- All code will disappear from the code tray

Collectibles

Pre-K puzzles will have collectibles similar to their codeSpark prime counter parts, but their application will be different. In codeSpark prime gems are used as a way to offer new challenges beyond the level goal, but in the Pre-K version collectibles will be the goal. Players will need to collect all the collectible objects to proceed to the next level. This will allow us to create more interesting puzzles which require the player to plan out their path around the board carefully. The collectible objects for Tugboat to Toyland will be the Foos floating in the water, and for Kaiju Demolition Crew the collectibles are the buildings you destroy.

No Stars

Unlike codeSpark prime, our Pre-K puzzles will not implement stars or other grading for level performance. The player will either complete the level, or they will fail it.

Completed Levels

When the player completes the level we will store the progress on our end, but the player won't see any updates on their end. Completed level data can be utilized to ensure levels that are new to the player appear more frequently than levels they've played in the past.

Metrics

To help build insight into a large portion of players we will send data to Mixpanel at key moments. These events will tell us about player performance (overall and based on current difficulty level), inform us if rubberbanding is helping player performance, and provide insight whether our features are being utilized.

What should be captured:

- Puzzle started
  - Player's point amount at start of puzzle
  - Time puzzle was started
  - Date puzzle was started
- Puzzle completed successfully
- Puzzle completed unsuccessfully (failed)
- Player’s current point amount
- Direction tile deleted from code tray
- Ghost guide enabled after failing