Introduction to Mekorama

An introduction to the game Mekorama
(developed by Martin Magni) written by his fans.

Table of Contents

DOCUMENT OBJECTIVES

This document is meant to serve as an introduction to Mekorama - an iOS/Android game developed by Martin Magni. It has been written by the members of (and published to the forum of) www.mekoramaforum.com. It is understood that most people visiting and joining this forum will already be well versed with the Mekorama game. Still, for the sake of completeness, and to support the other documents, the information herein may prove useful to some.

MEKORAMA APP

Objectives

The hero of Mekorama is B, a beautiful and amazing yellow bot. He's waiting for you to tell him where he has to go. He has only one mission: getting to the win spot. Actually, he is even less concerned about having to get there himself… as long as some bot or some object triggers that darn red button! Therein lies the challenge. Are you up to it?!

Presentation

Page 1: PLAY LEVELS

When you launch Mekorama (the game), you will be presented with a blue page with 50 cards representing the original **50 levels** of the game. You may also notice a flag in the top right corner suggesting that this is page 1 of 5. If you pull down on the page, it will in fact reveal the words: "**Swipe to flip page**." It's one of the greatest things you can do in the app!
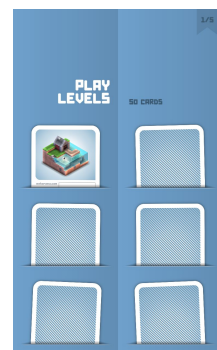
Page 2: SCAN LEVELS

Swiping to the left reveals a purple page. This page showcases levels scanned into the app using one of the two buttons at the bottom of the page. As they are described, the **camera button** is used to capture a QR code displayed on another device screen while the **image button** is used to capture a QR code previously saved to your device's photo library. Capturing a QR code will load the level uniquely identified by the code into your Mekorama app and it will be represented by its own card like those on Page 1 of the app.
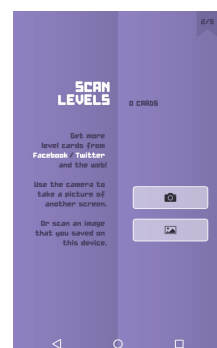
Page 3: MAKE LEVELS

This light blue page displays a card with a plus sign ('+'). Tapping this plus sign brings up a fresh card that, if "played," brings up a **level editor** with a blank slate. After entering the level editor, the card becomes a new level that will be stored on this page. The level/card can be eternally further edited from your app - as long as you don't delete it! You can share this card (as all others), but since there is no upload feature (yet?) for the level editor, it cannot be edited in other devices. There is no backup feature (yet?) either.

Page 4: PAY WHAT YOU WANT

This green page is a simple request for a contribution to the developer in return for playing Mekorama. There are **six buttons** representing different suggested monetary amounts that lead to an in-app payment pop-up. Any payment will unlock hints for the 50 levels in Page 1 of the app, remove the nagging donate request pop-up, and help Martin Magni be able to feed himself and work on the next development project.
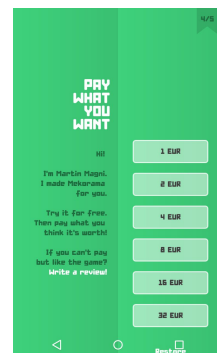

Page 5: OPTIONS and CREDITS

This pink page has some neat information about the game, but also four buttons:
- **Music**: can be turned on/off;
- **Sound**: can be turned on/off;
- **Hints**: can be turned on/off; and
- **Free camera**: can be turned on/off.

The fourth one, free camera, is particularly interesting as it (if turned on) allows smooth panning around the y-axis of a level in play. This can be useful to peer into parts of the level to get a better vantage point.
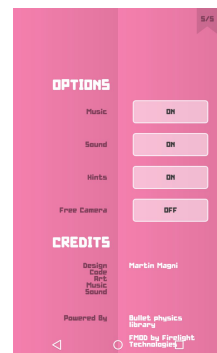
Note: There is also a title page before Page 1 and a page after Page 5 listing a website: "www.mekorama.com". This website features additional resources including a great quick **FAQ** list. One of the most popular questions is where can I find more levels. The best answer to this is to visit the fan-based www.mekoramaforum.com.

Cards

Cards are your gateway to a level. They are a visual representation of the level within the pages of the app and elsewhere online. When a partially visible card in the app is tapped (whether it is on page 1, 2, or 3), that card will zoom to fill the screen. After doing so, there are several pieces of information now visible and four actions that you can take.
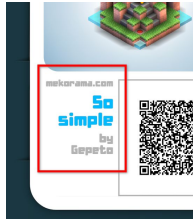

Card information

The information presented includes a thumbnail picture of the level, a QR code uniquely identifying the level, the name of the creator of the level, and the title given to the level. Note that in the original 50 levels, the level number will appear instead of the creator's name. Also note that if the creator didn't add the title and author, the defaults "New Level" and "by Unknown Author" will appear instead.
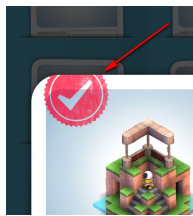
Card completion

When a level has been completed (i.e. the win block has been triggered), a checkmark appears at the top left of the card. Note that this checkmark resides strictly with the device on which the level was completed. In other words, it does not get exported with the card when shared. Therefore, if you need to prove that a level is solvable, you will need to take a screenshot showing the checkmark.

Card actions

Three buttons appear at the bottom of the card. The first one is an option to delete the level (you can't do this in the original 50 levels), the middle one is used to play the level, and the third enables you to share/export the level. Note that you can play the level simply by tapping anywhere on the card as well. Finally, you can exit the card by either swiping it away or clicking the X in the top right corner of the screen.

GAMEPLAY

Game Mechanics

This section will describe some simple and general operation and workings of the game.

Tap to move

As seen in the original level #1, "Crash Course," the first thing we learn is that if there is a spot in the level where B can walk to, tapping that spot will send B to it. Tapping in an inaccessible spot will result in a temporary red X appearing. If B is already in motion, tapping in an inaccessible spot will also stop B from continuing. The object of the game is to tap B all the way to the win block. Later we will see that triggering the win block in otherways is also acceptable.

Swipe to rotate screen

As seen in the original level #1, "Crash Course," the second thing we learn is that we can rotate the perspective of the level about the y-axis in order to get a better view. Note that there is a menu item on Page 5 of the app that allows toggling of the free camera. This switches between being able to see the level from 90-degree incremented angles or from all angles. The latter is not needed for playing the original 50 levels, but may be needed to successfully complete subsequent (user-generated) levels.

Using draggables

As seen in the original level #1, "Crash Course," the third thing we learn is that we can drag certain blocks. We can drag blocks that are brown in colour and have circles inscribed on all of their sides - aptly called "draggable" blocks. If the draggable blocks are connected to other brown blocks (or objects that can move), they too will be dragged along as a unit.

Draggables attached to sliders: In level #1, the draggable is limited in its travel by an attached "slider" which is sliding along a "rail." Sliders dictate the direction that the draggables can travel and a rail dictates the range of travel. In the original level #2 - "Run Of The Mill," we see the same action in the vertical direction.

Draggables attached to anchored motors: In the original level #6, "Spin Out," we learn about a secondary use case for a motor (explained as "Using a Motor as a Pivot" in the *Gameplay Features* section). With this trick it is possible to rotate draggables.

In all cases, draggables can be used to **complete paths** for bots and balls, **transport** the bots/balls to new locations, or **bump** into them for purposes like redirection, or subsequent bumping into other draggables.

Gravity

The Mekorama app is programmed using the Bullet physics library. One of the main wonderful effects of this is the concept of gravity in the game. There are a few objects that can fall or be bumped in such a fashion as to make them fly through the air for a bit as gravity takes over. For example, in the original level #11, "Lean Pivot," gravity is required to allow B to travel from one arm of the larger draggable to another. In original level #23, "Balance Ball," the player must work against gravity's effect on the ball and the balancing platform. And in original level #28, "Slam Dunk," gravity is both your friend and your enemy. In a couple later levels (without spoiling which ones), gravity is to be used more expressly as a mechanic in order to complete the level.

Directing R bots

In the original level #7, "Pagoda Push," we learn about R bots - or red bots that seem to turn right anytime they encounter an obstacle. This is intentional. And in this level we need to use B as an obstacle to get R to turn around toward the direction of climbing the tower. Examples of obstacles include things like draggables, other bots, balls, and obviously gaps or ends in the floor - anything that would prevent the body part of an R bot from continuing to the next linear space.

Triggering the win

Another lesson we could learn in the original level #7, "Pagoda Push," is that it does not have to be B who triggers the win. Although it is possible for B to trigger the win in this level, so too is it possible to direct R to the win. Both are acceptable. In fact, any means of triggering the win is acceptable. So long as the win block is triggered, the level will be considered completed. Another example is in original level #23, "Balance Ball," where the ball is used to trigger the win and yet another example is seen in original level 46, "Sarlacc Pit," (for which we won't spoil the surprise).

Avoiding zappers

In the original level #15, "Home Security" we are introduced to zapper blocks. What we learn is that if B walks within a block-space of a zapper, then B will be zapped and die. Only two levels later (in original level #17, "Bot Bypass") we learn that these zapper blocks can be attached to the top of an R bot (replacing its eye)! Be careful as this makes the zapper block mobile and B will still have to remain at least two blocks away at all times.

Riding atop R

In some cases you may find that instead of an eye or a zapper, R may have a metal pillar attached to him. Not only does this keep his body shape acceptable looking, it allows R to move around without having his head catch on corners like a cube block might. More importantly, though, the flat top of the metal pillar can be walked on top of by B (if at the same height). Example of this is seen in the original level #22, "Head Above Water."

Game Object Dynamics

This section talks about the objects/blocks that can be found in the Mekorama game. There are five general object types:

1. **Stone objects** are all solid grey and fixed in the gamespace (they include stone blocks, stairs, wedges, and the stone pillar);
2. **Metal objects** are all mostly brown (or perhaps dull brass-looking) and **may** move within the gamespace (they include metal blocks, metal pillars, metal half pillars, draggables, sliders, rails, motors, the ball, and the win block);
3. **Bot parts** are the bottoms of B and R bots and the eye [note that other combinations can be made, but for categorization they have been listed here as such];
4. **Other objects** include a grass block, red brick block, a zapper block, and a fence block; and
5. **Water** is simply a fill of translucent blue colour between certain blocks.

Here is the full list including a description of how each block is generally used:

Stone: grey solid cube. Since this block is a perfect cube, it has only one orientation and as all stone-coloured objects, they are immovable. Some may perceive these blocks as concrete.
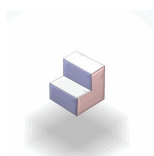
**Stair**: grey/stone-coloured L-shaped block that is fixed in the gamespace. Given their shape, there are twelve possible orientations for the block. Depending on their orientation, they can have interesting effects on bot movement. Furthermore, if used in different orientations, this block could make for good art (window edge, seat, recessed door, etc.). Some may call this block a step, but stair block is more common and accurate.

Wedge: exactly one half of a solid grey/stone-coloured cube split diagonally. This block is fixed in the gamespace with twelve possible orientations. It can allow balls to roll down and free-draggables to be dragged up them. They can also be used for art and for allowing other moving blocks to pass by more easily. Some may call these blocks a ramp, (particularly if orientated in a way similar to the picture on the left).

Stone Pillar: grey/stone-coloured cylinder that can be orientated in three directions (one in each axis). Sometimes it is called a cylinder. Like a wedge, it can be used for art and for allowing other moving blocks to pass by more easily. It is similarly fixed once set in the gamespace.
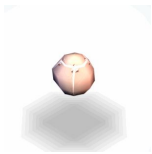
Ball: brown/metal-coloured near sphere-like shaped object that can move in accordance with the physics library as expected a ball would, i.e. roll, fall, etc. Although it is a metal-coloured block that can move, it does not attach itself to any other block (as there is no flat surface by which to do so).

Metal: brown solid cube. Since this block is a perfect cube, it only has one orientation. It is able to move if attached to something else that can move (such as a draggable, motor, slider, or bot body), otherwise, it stays fixed in the gamespace. The same holds true for metal pillar, half-pillar, rail, win, fence, and zapper blocks (but not draggables, motors, sliders, and balls, which can never be fixed unless blocked from moving).

Metal Pillar: brown/metal-coloured cylinder that can be orientated in three directions (one in each axis). Since nothing can attach to the rounded sides of this block, it is most often used when two sets of draggables need to push up against one another (e.g. original level #16, "Crank It Up") or act as a spindle for a free-draggable (e.g. original level #23, "Balance Ball").

Metal Half Pillar: brown/metal-coloured block with a flat top, bottom, and one edge side; the other three sides rounded. This is also a block that can potentially be moved, and is most useful whenever both a flat side and an opposite rounded side are needed (or look good).

Draggable: brown/metal-coloured solid cube with circles imprinted around the centre of each side. Any set of blocks required to be dragged by the player requires at least one of these blocks in the set by which to drag the entire set with.

Slider: brown/metal-coloured cylinder with a dark hollow core that moves only in the direction of its hollow core and thus dictating the movement of any other attached blocks. If attached to a rail, it will ride back and forth along the length of the rail. If no rail is attached, then the slider will be bounded only by the edges of the gamespace or some other block that could obstruct it. If the slider is attached to a draggable block, then rather than being self-propelled, it will move only along with the draggable.

Slider Rail: brown/metal-coloured straight (three orientations) or 90-degree curved (twelve orientations) stanchion piece. This object's primary role is to act as a guide for sliders - both as a visual cue and as a limit to the slider's range. Note that a slider can travel only over a straight rail and not over a curved rail. Since rails are metal-coloured objects, they can also be used to join other metal blocks so as to keep them together as one draggable unit.

Motor: brown/metal-coloured, octagonal, button-like shape with a protruding black cross on one side (much like the start of an axle in a LEGO® set). It can be used to power the rotational movement of other metal-coloured blocks or act as a pivot when attached to a draggable. [Note (that particularly for this block) there are some other behaviours and peculiarities with the motor that are beyond the scope of this document.]

Win: brown/metal-coloured block with a red circle on the top and bottom and stars on the other sides. Sometimes it is called an "end button," "goal," "finish," "target," or "trigger." Note that triggering either top or bottom red circle will complete the level played. And it can be a moveable block too.

Fence: red panelling with brown/metal-coloured edges on two sides. Although this block can be attached to and move with other moving objects (like often seen as sails in a windmill), it is special in that it cannot be walked on by a bot (either by B or R no matter its orientation). This block can be orientated in twelve different ways.

Zapper: a half-height octagonal shaped device with four electrical prods extending up from it. Although this block can be attached to and move with other moving objects, it is special in that it will kill B if B comes within one block-length of it (or it - the zapper - comes within one block-length of B).

Grass: beige/brown block with a green top that looks like vegetation on top of soil. And since it is a perfect cube, it rests only in one orientation. It is one of the three block types that are always fixed in the gamespace and thus affect the boundaries of water. And it is the only block that changes if covered with water - the green vegetation part is removed if underwater.

Brick Red: often just called a brick or brick block, this red cube has lines on it that make it look as if made from bricks. As it is a perfect cube, there is only one orientation for the block. It, like the grass and stone-coloured blocks, are always fixed and thus set the horizontal boundaries for water.

Water: the translucent blue fill colour that appears as water between the furthest stone-coloured, grass, and brick blocks (which it uses as its horizontal boundaries as these blocks are always fixed in the gamespace). The water does not affect gameplay (it only affects the sound), and it makes for a pleasing visual feature of a level.

Eye: white sphere with a flat bottom and a black dot that somewhat randomly looks around. The bottom of the eye can be attached to a bot body (or the game floor or another block). The behaviour of the dot becomes more predictable when attached to B.

Robot Tap to Walk: this is the bottom half of the **B bot** (body, arms, and legs). Regardless of the head configuration, nearly everyone just calls this object the B bot or just B (even in the headless configuration). Probably because of the letter 'B' written on this bot's front. ;)
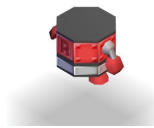
Robot Turns Right: this is the bottom half of the R bot (body, arms, and legs). Depending on the configuration of the head, the following are more common terms:
- With an eye or a metal pillar for a head: **R bot** (or just R)
- Headless: **headless R**;
- With a zapper for a head: **zapper bot**.

Note: the dynamics of the R and B bots are vast and complicated. As such, these dynamics have been left for the following *Gameplay Features* section of this document (and the appropriate sections of other planned documents).

Gameplay Features

In this section, some "features" of the game will be explained. By features, we mean behaviours that, although they may seem odd (maybe even as bugs), they are either intentional or (otherwise observed within the original 50 levels and thus) deemed to be intentional.

"Physics Library"

Mekorama was coded using the Bullet physics library. Thus, observations of physics like gravity and objects in motion are actually core to the game - so much so that they can even be used to the player's advantage (see "Gravity" above in the *Game Mechanics* section for some examples).

"Water Effects"

Starting with original level #1, "Crash Course," we immediately learn that water seems more of an artistic effect than a game element. All objects, including bots, can enter water without any change in behaviour. What does change is the sound and visual effects. When objects enter the water, they make a splunk sound and a ripple effect. The sound of B's feet tapping underwater also changes to a more muffled sound. You'll also notice that water can only be at one height in a level and that grass blocks will always lose their green tops underwater.

"Low Ceiling" (AKA "Claustrophobia")

If B is asked to go to a spot that has a block immediately above, B will go to that spot and then keep going until he reaches a spot without a low ceiling (with a few exceptions). It may seem that his route is calculated in that order, but in fact it is calculated when the spot is tapped. If there is no exit at that moment, B will simply go to the spot tapped. If there is an exit, his route will be planned for him all the way to that spot beyond the low ceiling. If an R bot happens into his path, B will stop when encountering him. Other exceptions to this feature are explained in the "Advanced Level Design" document. This effect can first be seen in original level #2, "Run Of The Mill."

"Using a Motor as a Pivot"

Often a motor is used to rotate blocks attached to its sides, but the opposite can also be true: A motor can itself be turned by a draggable attached to it. In this case the motor is just a pivot point for the draggable unit. This effect is first seen in original level #6, "Spin Out." [Note that there are some exceptions or restrictions to this feature that are beyond the scope of this document.]

"Stone Pillar Allergy"

There exists one block type that an R bot cannot traverse while a B bot can - this is the stone pillar. Since R will refuse to step on a stone pillar, he will instead turn as he normally would encountering an obstacle. The stone pillar can be in any orientation (vertical or horizontal). This effect is seen in original level #14, "Back Track" and level #24, "Upmill Battle."

"Machinery"

In the original level #18, "Crank It Up," we learn that one set of draggables can be used to move another set. Furthermore, since B is not standing on the draggable being dragged, he does not have the familiar sticky property keeping him on the draggable. In fact in this level you can make B jump. The trick to having one set of metal blocks move another set - yet not be connected as a whole - is to use round edges between the two sets of draggables. In this case a metal pillar connected to one set is used to push against rails connected to the other set..

"Side-by-Side Stairs"

The first time that multiple sets of stairs are placed side-by-side is in the original level #21, "Venice Volts." If you have time to experiment while avoiding the zapper bots, or you simply mis-tap, you may find that B does not like to cross from one stair block to another one adjacent to it. Instead, B will insist on going to the top or bottom of the stairwell that he is currently on, step one block over, and then return back up/down the adjacent stairwell to reach the spot tapped. This seems intentionally coded by the developer, @Martin Magni.