PuppyBot Rescue - A Balancing Game

Matthew Champer, Samantha Collier, Ricardo Merchan, Michael Christel, Bryan Maher, Scott Stevens
Entertainment Technology Center, Carnegie Mellon University

Example URL: http://www.workingexamples.org/example/show/624

This project is taking all the experience from our past games in the ENGAGE initiative and collaborating with Sesame Workshop to build a game under their Electric Company IP.

Seed

Tell us about your idea or project. What's your vision?

We want to get this game into the hands of thousands, if not millions of kids and see if science concepts can be taught at a younger age (ages 6-9), in an appealing way. Will children play our game and learn from it? Will they play it not because it is assigned homework, but because it's fun? One of the ways we are hoping to do this is by working along with Sesame Workshop to develop a game that could be launched onto their newly re-designed Prankster Planet website. In doing this, we would be able to reach larger numbers of users than past projects.

The science content we are teaching comes naturally through age and experience, and so we're trying to structure our game so that younger children that have yet to master it and obtain a grasp of it might understand the more complex concepts sooner. Sometimes kids can understand the action, but not the logic behind it - and bridging the two is our goal.

What problem are you trying to solve and why does it matter?

Teaching kids through games. But not just typical teaching - we want to inspire kids through games that have embedded educational objectives to become interested in science and engineering careers. Simply put, we want kids to find the fun in tomorrow's careers.

What history or context should we know about?

This game is being built on a foundation of knowledge and experience from our past games:
RumbleBlocks (http://www.workingexamples.org/example/show/592),
Beanstalk (http://www.workingexamples.org/example/show/593),
Teeter-Totter-Go!, and Helios (http://www.workingexamples.org/example/show/595).

All of the above games excluding RumbleBlocks were designed and built around research papers written by a Carnegie Mellon University professor, Robert Siegler, based on balancing a beam. For example, see Siegler and Chen, Development of Rules and Strategies: Balancing the Old and the New, J. Experimental Child Psychology 81 (2002), (http://www.psy.cmu.edu/~siegler/siegchen02.pdf). With this foundational knowledge, we are able to design the challenge of the game relatively easily, while focusing our core effort on the mechanics and the look and feel.

In general, each game is developed in the course of one semester through the combined skills of a multidisciplinary project team. These efforts are guided by iterative playtesting with children. That is, at many points during the semester children play the game and through direct observation, questionnaires, and interviews the development team learns what works and what does not for the target audience. The game is tweaked and made ready for the next round of playtesting. The teams have produced a few papers documenting this process. Beanstalk is discussed further in (http://doi.org/10.1109/IGIC.2013.6659126) and Helios discussed further in (http://doi.org/10.1109/CGames.2013.6632614) -- see these papers for additional resources on the game design, on Siegler rules, and on the use of playtesting during development.

Controls and Mechanics

With early RumbleBlocks user testing, we found that the younger the child, the more difficult it was to control a mouse, and even more so, a keyboard (even if told to use just a select set of keys). RumbleBlocks utilizes left and right mouse click, spacebar (alternate for right click) and the arrow keys (also alternates for the right click). We found that using the left click and spacebar to rotate a block was the best combo. More on iterative playtesting to develop RumbleBlocks can be found in the conference paper at (http://doi.org/10.1109/CGames.2012.6314570).

With Beanstalk, we decided to remove all the complexity of the controls from RumbleBlocks (click and drag, combination of inputs such as hold a block and press spacebar to rotate, and multiple inputs). Beanstalk was simplified so that now a 5 year old could play comfortably - all controls were just a click - no dragging, no combination inputs.

Over time, we found that the over-simplification of Beanstalk's controls were a bit monotonous. So with Helios, dragging was reintroduced, with touch input in mind - and then with PuppyBot Rescue, flinging the blocks as a new mechanic was incorporated. This allows players that can't completely control dragging some freedom to controlling their object.

Inventory

Since RumbleBlocks, we've been advised to put the entire inventory out on display for the players - something we always tried to incorporate, but never found a justifiable way in doing so.
With PuppyBot Rescue, this was highly considered in the design process, and ultimately - incorporated.

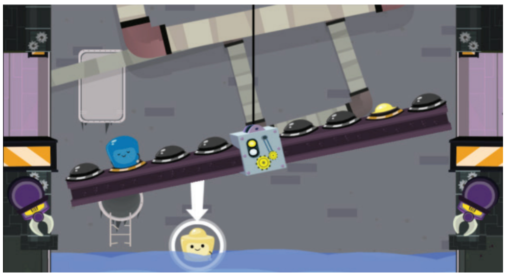


Figure 1: http://www.workingexamples.org/uploads/Image/711

Audio & Instructions

With our first few games, we found that audio, when used as instruction, was ignored if there wasn't an equivalent amount of on-screen visuals. With a lot of collaboration and advice from Sesame Workshop, we reduced and simplified the voiceovers throughout our current game, PuppyBot Rescue. Kids have short attention spans, and we must consider that in our designs.

What are your goals and how will you know if you've achieved them?

Our goal is to build an educational game that embeds the educational objectives behind fun mechanics. We then collect data about how the beam is balanced and the contrasting case answers through a logging server to determine if children are learning from the game.

Types of data that we collect are:

- # Sessions
- # Problems encountered
- # Moves made within a problem
- Amount of time spent on each problem
- Problem start states (e.g., inventory, beam set-up)
- Problem end states (success or failure)
- # Idle states (time of no player activity; not counted as part of time within a problem)
- # Sessions reaching any particular level, including overall win-state at end-of-game

With a massive amount of data, we can determine if, and how, children are learning. This information can help to both improve how this game teaches Siegler rules, as well as how educational science games are designed and deployed. If successful, maybe in 20 years we'll see a rise in engineering and science careers/degrees. In particular, we are incorporating adaptive learning so that for players who are quite successful, the game will remain challenging and interesting by proceeding quicker to more advanced problem levels. For children who are experiencing a lot of failure, the game will move more slowly through problem levels. We are striving to remain in a good state of flow, not too easy to be dull, not too challenging to be frustrating. Since we deal with a big enough age range that child learners have different capabilities, we need to be flexible in our problem advancement approach. Our design is guided by the work of Csikszentmihalyi and by the various game lenses espoused by Jesse Schell in his Art of Game Design book. For struggling players, we offer gentle scaffolding help and on repeated failures more detailed precise help to work through problem levels as needed. The adaptive progression though problem levels and the use of scaffolding are also recorded in the game logs, enabling us to study different types of adaptive and scaffolding strategies.

Who will your work impact? What do you know about them?

Kids! Kids from all over the US. And as we all know, kids are the future. But we're also keeping parents and teachers in mind; from things like audio usage in classrooms to designing the game so that when at home, there are no reading roadblocks.

Our demographic is Kindergarten through 3rd Grade, with a broad age range of 5-11, but with a more core range of 6-9 (our game is adaptive, being able to adjust the content to individual users, so there's no reason to not include someone who may still find the game enjoyable and challenging).

Kids love touch input devices (tablets, phones, etc.) and we've designed our game in HTML5, by-passing the middle man of going through app stores and having to download content, needing just your modern web browser. Hence, the game will also run on computers and other web enabled devices.

*Note: Not tested on all mobile browsers. If you are having problems, please let us know. Tested and runs on Chrome, Firefox and Safari with current generation devices (ca. 2013 forward).

How does this work relate to what others are doing in the field?

There are a few groups out there that are choosing and building HTML5 games as an option over an app installation. There are pros and cons to each, but with HTML5, we can launch one build of the game universally.

In terms of the educational aspect, we are attempting to teach advanced educational concepts, based on research written by a Carnegie Mellon University professor, Robert Siegler, based on balancing a beam (i.e., rules that children don't often acquire before becoming teens) at a much younger age, targeting ages 6-9. Some educational developers are working on tutors, aimed at being complimentary to the educational material being taught in the classroom. Here, we want the game to stand on its own, capable of being used by a child at home without parent or teacher intervention.

How can our community support you? (Expertise, resources, etc.)

Anyone can help! We need lots of playtesters, critiques, teachers, etc. Know someone that fits into our audience? Even better, have them try the game and send us some feedback! Found a bug? Something doesn›t look right? Maybe we missed something. The more feedback, the better our product. Please promote our game link to teachers, parents, and children. The more players, the more play data we can collect and then use to revise the adaptive problem level progression and scaffolding techniques to deliver a better experience to players.

With these games, we are trying to inspire more children at early ages to pursue science and engineering careers through the use of fun and entertaining educational games. We like being pointed to other games, seen anything neat out there? Send it our way!

Tell us about the team you have assembled or hope to assemble.

Our team consists of:

Scott Stevens - PI (Principal Investigator)

Mike Christel - Co-PI, Programmer

Matt Champer - Artist & Designer

Samantha Collier - Artist & Producer

Bryan Maher - Programmer

Ricardo Merchan - Sound Designer & Writer

As noted earlier, it builds from a number of Entertainment Technology Center (ETC) (http://www.etc.cmu.edu) projects that are listed as separate working examples: see ETC DARPA ENGAGE (http://www.etc.cmu.edu/engage/) for further details. PuppyBot Rescue development is supported by DARPA's ENGAGE program.

Sprout

Tell us about your process and how your idea is evolving throughout the project.

The early years; or the evolution of our project:

This project has been an ongoing process over the past 3 years. In that time, we have developed 4 different games and are currently on our 5th. Because of that, our project has a lot of experience and history to draw from. With our current game, we looked back at everything we learned from our past experiences and then thought about what goals we wanted to reach with this one.

We migrated our development platform from Unity3D to HTML5; mainly due in part that schools need to install a plugin to use anything made in Unity3D. With HTML5, we can develop and push out to multiple platforms and skip over app store gates and web browser plugins.

Over the course of our earlier prototypes for what eventually became PuppyBot Rescue, we trimmed down the UI (User Interface) in order to keep the focus on the core game mechanic. After we got a good idea where we wanted to go, we began thinking about a few basic features.

We asked ourselves this:

1. We wanted the interaction to be fun and break it down into its most basic form. How can we make the action of interacting with our game fun?

2. What are some fun mechanics that could reach this goal?

3. How can we keep the game play simple, so the educational objectives are the focus?

We began trying to solve these questions by brainstorming and coming up with a series of concepts. In the beginning, we focused on 4 basic ideas encompassing a few different mechanics and ways of interacting with the game.

With those 4 ideas we made storyboards and game documents to better understand where they could be taken; this process was very helpful. A lot of ideas sound really achievable after you begin to walk yourself through how they would pan out.

In order to make the interaction more fun, we went with a 'flinging' mechanic - where you can pull down a little bit and fling the object up by letting go, almost like a slingshot. But we also kept the drag mechanic, because we found that some players attempted to use that mechanic instead - and we shouldn't punish a player for not using a specific mechanic if they feel something works better. We have tried to incorporate this idea through the game, trying to accommodate different possible interactions that we have observed during playtests.

The 'flinging' mechanic was born from collaboration between Sesame Workshop and past playtest experiences and results. Since we find just simply dragging and tapping are somewhat mundane mechanics, we knew we had to spice it up a bit, while still keeping it relatively simple.

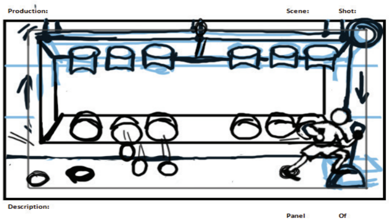
What are some of your initial concepts or designs? We'd love to see them.

There has been a great deal of development and changes made to our concept and design along the way.

As mentioned, we started with 4 different ideas. We came up with these by: thinking about what sort of mechanics we wanted to have that would be fun, what would be a cool way to interact with our "beam" that we are using to support our educational content, and keep the focus centered on these things.

What we came up with:

Some of our first ideas were games where you would move an avatar back and forth to throw things on the beam. While you did this, a trouble-maker would be constantly unbalancing the beam causing you to rebalance it until time ran out.



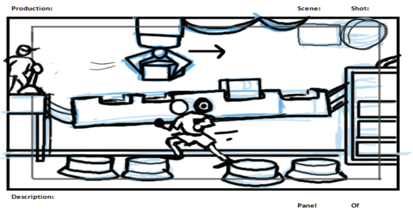


Figure 2: http://www.workingexamples.org/uploads/Image/519

Figure 3: http://www.workingexamples.org/uploads/Image/523

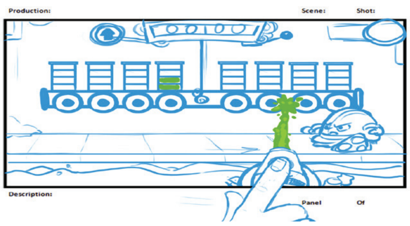
Some of our other concepts centered on games similar to ones you would play at fairs. These were more simple concepts that were similar to mini-games.

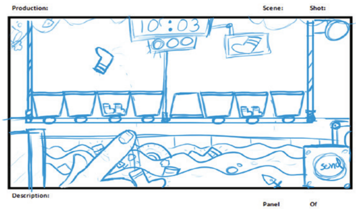
After looking at all the concepts we had, we decided to stay away from having to interact or control an avatar. We wanted the focus to be centered mostly on the beam. Moving away from controlling an avatar kept our controls simpler, and kept the game in more of a mini-game space; keeping a quick paced, fun and simple game. Also by avoiding an avatar, deploying the game for mobile devices meant not having to come up with a clunky control scheme to control on a touch interface; something we believe is too complex for our demographic.

From the original set of ideas, we narrowed it down to three concepts.

From there, we began to flesh out the interaction by creating detailed storyboards that walked through the whole experience.

Idea 1: Sewer Slime: You are tasked with taking samples of the slime that was building up in the sewers. While doing this, you need to make sure the beam stays balanced. Once the beam was properly balanced, you would pull a lever to raise the beam out of the sewer. It isn't as simple as you think, not with the baddy bots getting in your way. To keep them from blocking you, you have to slime them, knocking them out of the way.





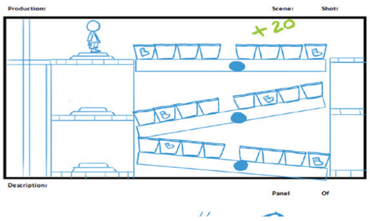


Figure 4: http://www.workingexamples.org/uploads/Image/524

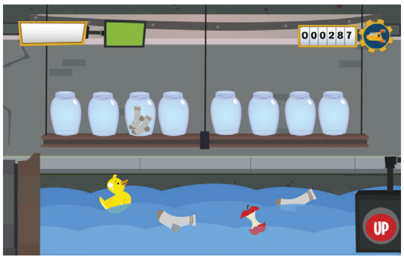
Figure 5: http://www.workingexamples.org/uploads/Image/525

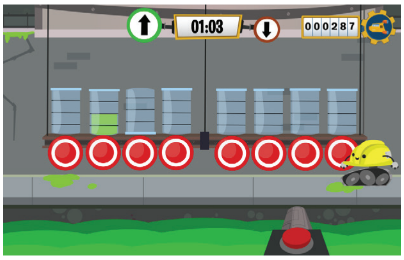
Figure 6: http://www.workingexamples.org/uploads/Image/526

Idea 2: Sewer Toss: Things on earth have been going missing: socks, toys, etc. After looking for an answer, we find that they have been falling down into the sewers. To help recover these items, it is up to you to load items onto a beam to take them back to the surface. While trying to recover items, one of the important stabilizing cords break. Now, to be able to get the beam to rise properly, you need to fling items out of the water into baskets while making sure to keep the weight balanced. The faster you collect needed items, the more points you get, but watch out for some of the creatures living in the sewer thatcould mess you up!

Idea 3: Balancing Act: (This was seen more as a testing mini-game). The tricky pranksters have scrambled 3 beams - only one being properly balanced. You're trying to get out of the sewer but which way is the way out? It's up to you to figure out which beam is balanced correctly.

From those storyboards, we created mock-ups to show the styling of the art.





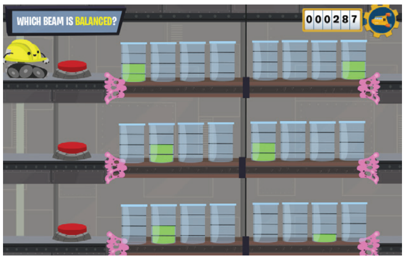


Figure 7: http://www.workingexamples.org/uploads/Image/527

Figure 8: http://www.workingexamples.org/uploads/Image/528

Figure 9: http://www.workingexamples.org/uploads/Image/529

After reviewing these concepts with Sesame Workshop, we merged pieces of each together until we landed on a solid game. They also suggested changing the gameplay so the player isn't stationary, but rather traveling through the sewer. We hoped that this would give the kids a sense of moving through the space and making progression, to provide accomplishment.

Where did we land?

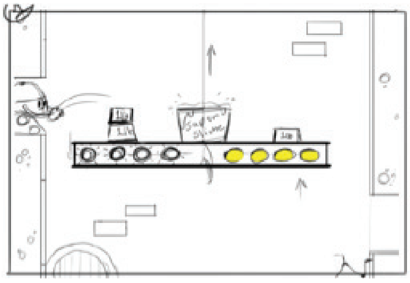
The earlier form of our current concept was that you would be pulling a beam up out of the sewer carrying some container full of slime. And on its way to the top, you had to make sure it didn't spill. To do this, the player has to keep the beam balanced all the way up, trying to stop the efforts of the bots that are constantly adding and subtracting weight to the beam.

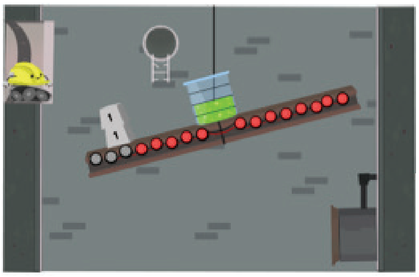
Have your initial concepts/designs changed? Why have they changed? Show us how they're being refined and iterated.

The finalized concept/design we went with has changed dramatically over time. Initially, the game was very systemic, rigid and felt like you were solving a bunch of problems. Over time with feedback from Sesame Workshop and playtests, we've evolved the game into a more dynamic, faster-paced experience, adding fun animations and movement wherever we can and making the interaction as fun as possible. It has come a long way from its simple beginnings, and has turned into a fun game it's currently growing into now.

Where we started:

From concept to mock-up, we started off pretty simple:





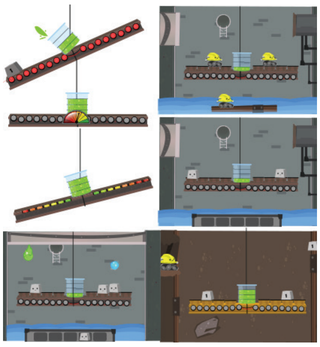


Figure 10: http://www.workingexamples.org/uploads/Image/538

Figure 11: http://www.workingexamples.org/uploads/Image/531

From there, we began diving into our game. Focusing a lot on how the beam would look, and how to interact with it. Soon, we added water elements and off-balanced pipes that we would need to balance and unbalance the beam to connect.

We quickly realized that the slime meter was a poor choice to show balance. After paper testing, we found children didn't make the connection with the slime and the balancing of the beam. Another element we removed was the complexity of the lights on the beam, creating an overall simpler interface for children to interact with.

Another very important piece to our game was designing the weight. Throughout the game, you would be interacting with this object. We found that children really enjoyed the bots off to the side, and we wanted to carry that over to the weight. We gave our weight-object a personality. Starting off very simple; the first change was to make it friendlier by rounding it out and giving it more animation. We found that the game greatly increased in fun after. From there, the weight changed a lot over the progression until we found what would work best with what we had.

Number one lessons learned:

- Make the object interesting and appealing
- Personality makes a world of a difference
- Make a clear connection between the object and the beam
- Mixing the idea of electrical objects with water was a very bad idea

These lessons lead us to our final design.

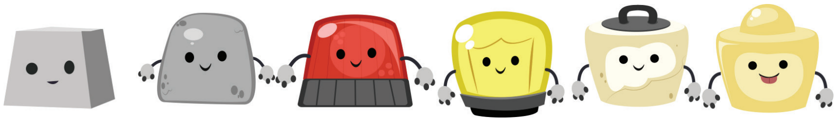
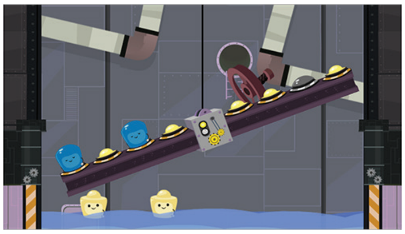


Figure 12: http://www.workingexamples.org/uploads/Image/540

After a good bit of progress, our design changed to its current state. Clearly, a long way from its simple beginning...



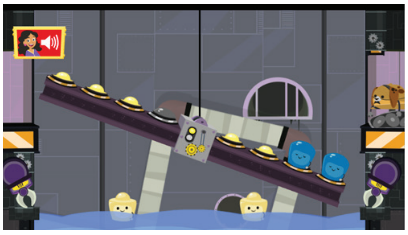


Figure 13: http://www.workingexamples.org/uploads/Image/546

Figure 14: http://www.workingexamples.org/uploads/Image/548

...Helping a bot out of the sewer by balancing and unbalancing beams so it can travel up a path. While doing this, you need to make sure you solve the beam problems fast enough, or water will come up and short out the gears that are pulling the beam up.

Major Changes and Why:

- No longer working against a bot, but instead you are helping a lost puppy bot. This pulls from kids' desire to help and have a positive motivation.
- Making the objective of why the child needs to balance the beam very clear with showing that the puppy bot is trapped and needs your help.
- Simple beam and block mechanic; stripping away an over-cumbersome UI (User Interface) and interactions. With clear objectives there is less room for misinterpretation.
- Blocks with fun interactions and personality to add life to the action of balancing the beam.

Once we had that design mostly hammered out, we had to make the transition, or the contrasting case part of the game.

The original design was set up more in a way that you had to help Puppy-Bot open a door to continue on to the next level. You did this by answering a question via pushing a button.

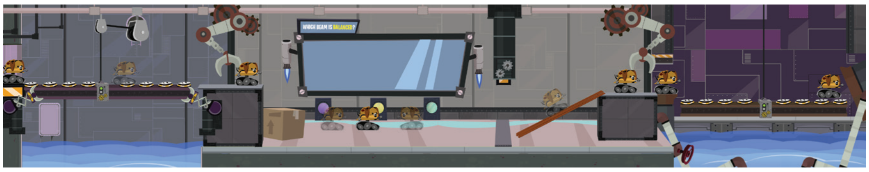
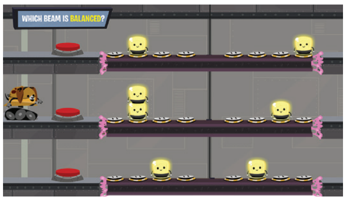


Figure 15: http://www.workingexamples.org/uploads/Image/542

The only problem with this was that it somewhat distracted the player from the whole game and wasn't interacting much with the environment. That led to our second design. For this, we went back to an earlier design for the game and repurposed it.



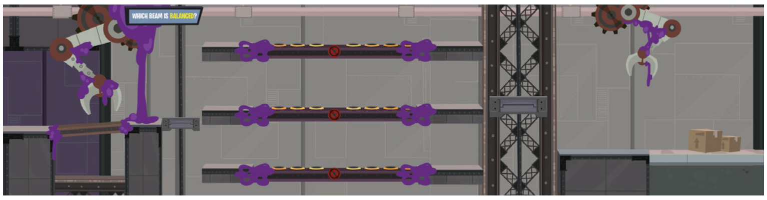


Figure 16: http://www.workingexamples.org/uploads/Image/545

Figure 17: http://www.workingexamples.org/uploads/Image/544

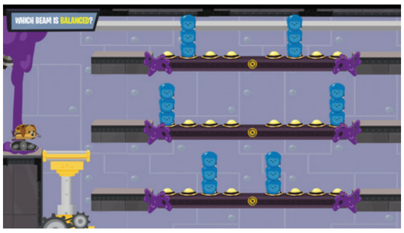
From there, we made some adjustments to combine the two concepts into one; a large transition area that Puppy-Bot needs to travel through to get to the next level.

But we weren't done yet. After building the prototype we realized that this level wasn't very interactive, but was going in the right direction. To add some more fun elements, we incorporated a moving platform you had to move to direct Puppy-Bot to the correct beam.

After some testing rounds, we found that about half of our users were having difficulty with the elevator mechanic (dragging up and down). While it might be easy on a tablet, a mouse as an input device for kids is a whole other story!

So what we did was went back to the drawing board - and took away any thought-process involving mechanics and design - and went with a simple, fun and rewarding mini-game without any penalties. Kids love simple fun stuff.

Our new Mini-Game:



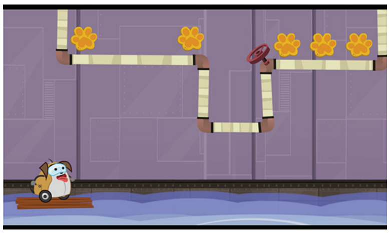


Figure 18: http://www.workingexamples.org/uploads/Image/551

Figure 19: http://www.workingexamples.org/uploads/Image/664

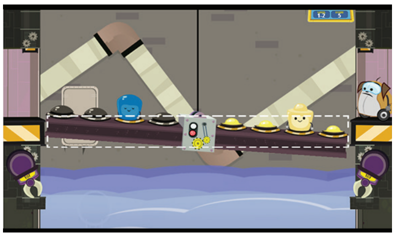
PuppyBot Snacks!

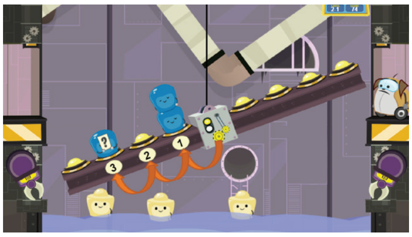
PuppyBot rolls onto screen, down an elevator and onto a raft and floats across screen. While PuppyBot is en-route to his next sewer shaft of unbalanced beams, the player gets to help him out by feeding him lots of PuppyBot Treats! Because, well, solving puzzles makes you hungry! To feed PuppyBot, you simply tap a treat and time it so PuppyBot snatches them up in time.

While this seems like a pretty straight-forward task, we did run into some unexpected problems. When creating this, we thought the object of the game would be very clear, but we quickly learned that above anything else, you need to test with kids because as designers sometimes we get too familiar with our work and have a hard time looking at things objectively. When we finally got this in the hands of kids, we were shocked to find that most kids didn't understand what they should be doing during this part of the game. The few that did catch-on, weren't sure of how to interact and stumbled over the controls. Thanks to testing, we were able to address this problem by providing proper visuals to help instruct the player in a simple, quick way that doesn't subtract from the mission of the mini-game.

Hinting System:

We also integrated a hinting system in order for a user who may not be fully getting the difficulty of the problem, or may just not understand the mechanic. After a determined amount of time, and if the beam is close to being solved - but not quite yet - a dotted line appears to show the user that the beam isn't quite balanced yet. This iteration comes from playtests and observing how players think the beam is balanced when it's actually just nearly there.





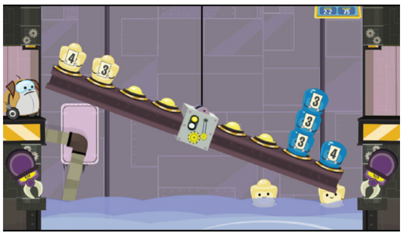


Figure 20: http://www.workingexamples.org/uploads/Image/665

Figure 21: http://www.workingexamples.org/uploads/Image/703

Figure 22: http://www.workingexamples.org/uploads/Image/704

Advanced Hinting

Upon any successive fails, the game loads a more step-by-step instructional hinting system:

Where arrows count one-by-one, each block on the beam and how far it is away from the center. The sleeping blue blocks hold up “?” a question mark to indicate that amount of force isn’t known - yet.

Next, the blocks hold up a card indicating their amount of force on the beam at that location. Kids can experiment by moving them around and seeing how each block affects the beam. In this way, they can count and determine how much each side needs to be in order to get the correct sum on both sides.

What we learned about Audio:

- Certain sound effects were very repetitive and annoying, and so we either replaced them or tweaked their levels to minimize the annoyance factor.
- Running music, sound effects, and voice overs all at the same time muddies the overall effect, and distracts the brain while listening. What we ended up doing is normalizing the decibel levels down to -6 for voices, -10 for sound effects, and -20 for music.
- Voice of the antagonist motivated kids a lot.

How will you make sure that this thing you’re creating will be adopted by your audience?

By getting it on PBSkids.org with the awesome help from the folks at Sesame Workshop. Once there, we will be at the fingertips of millions of kids.

But what if we don’t make it!?

While we are working towards the objective of creating a game that can be skinned in the Prankster Planet IP, we are also creating a version that can go to websites such as Learning.com and BrainPop. Comments welcome: please suggest where a game teaching 6-9 year olds Siegler principles of balance should be placed so that parents and teachers can find it and assess its value, and so children can find it and play and enjoy the fun (and learning) it offers.

How might your project scale to provide greater impact?

Adaptability:

We’ve just begun implementing an adaptive feature into our game. In brief, that means an excelled user who can manage to continuously get perfect or very good scores will advance further up our problem stack in difficulty. But, when they begin to fail, we start to slow them down to the normal pace, or even bring them back to some skipped over problems.

For users that don’t excel, we keep them at the normal pace, which is built to be very gradual and scaffold actions so that they can learn how to balance and solve the beam. At the moment, we don’t ever drop them back to earlier levels (only skipped over levels), but after a failed problem, we repeat the problem until they advance.