Doctor Bongo
Design Document

Mads Kjærulf Caspersen
Hans Petter Ellingsen
Kristian Helle Jespersen
Kenni Skafte Mortensen
Christoffer Holmgård Pedersen
Jasper Theisen

December 20, 2006

Contents

1 Doctor Bongo - The Groove Prescription.

Doctor Bongo is a combined rhythm/puzzle game. It combines the challenges of fast thinking, timing and strategic overview into a hectic, enjoyable single or multiplayer experience.

Doctor Bongo takes the very intuitive input and feedback schemes of rhythm games and merges it with the puzzling task of manipulating pieces with mutual and cascading influence upon one another into a certain desired state. In this design document we will describe the concept and prototype for Doctor Bongo and describe our design choices validated by playtests.

In the following we describe the story behind Doctor Bongo, in detail outline the unique gameplay, the marketing expectations of the game and the technical implementation of our prototype. We also document the creative process that led to the development of this specific game idea and its unique identity and how the original idea was refined through the tools of testing and evaluating.

2 Story

This section contains a description of the story of Doctor Bongo as well as a description of the different characters present in the game. We will refer to different gameplay elements during this section. These will be explained in depth in section 3.

2.1 Complete Story

Dr. Bongo is a brilliant high energy physicist - a slightly mad, but kind fellow. Many of Dr. Bongos mind boggling creations and startling scientific results, can be attributed to his unique inspirational method, of combining ganja and bongo drumming, to achieve a higher state of consciousness; a position from where he looks down and sees the everlasting truths and connections of the loving Universe. Dr. Bongo knows that all living things are connected. He knows that love is what binds the Universe together mon. He knows that you must always stay true to big mother earth and da ganja mon. Oh, and you should always fight the power! When we meet Dr. Bongo he has, in collaboration with his super-intelligent monkeys, created a time machine to reach key points in history, were he hopes to spread the groove, and fight the powers of hatred and war, and help da people stay true to da mother earth. With the help of his monkeys dancing. His adventures take him all the way from the towering yuppie filled skyscrapers of the 80's to Viking war boats. And at one point he even has to up the groovyness of the Bush administration.

In the prototype level, Dr. Bongo has warped back to the eighties and is disheartened by seeing that all the yuppies have abandoned the ways of love,

ganja and groove for individualism, cocaine and synthpop. Deciding that he must help the poor yuppies get back the groove, he enters a fourstory skyscraper and starts drumming ...

At the end of the level the combined dancing of the yuppies puts the building into a state of cascading resonance amplification and the building succumbs to its own vibration and comes crashing down. The yuppies stand on top of the rubble, cheering, chanting and dancing, kept safe by the power of love and groove.

2.2 Game Characters

Dr. Bongo: The main protagonist of the game. Dr. Bongo is half highenergy physicist, half enthusiastic rastaman.

The Superintelligent Monkeys: (Not featured in the prototype) The superintelligent monkeys are Dr. Bongo's creation and his main coworkers. They wear lab-smocks, glasses and occasionally smoke pipes. They power Dr. Bongos time machine by pedalling when they travel through time.

The Yuppies: Once young and ideological teenagers, the NYC zeitgeist of the 80's led these poor men and women astray, and turned them into grown up greedy business people preoccupied with hostile takeovers and pointless self praise through a warped hedonistic lifestyle with no love for the connectedness of the mojo of all living things. Dr. Bongo and the monkeys have come to save them and set them straight mon'.

2.3 Character Design

The characters of Doctor Bongo are designed to signal humour and fun drawing on archetypes and a cartoony style.

Dr. Bongo Gives the impression of being a laid back type - but with a mission. Using his special powers for the greater good of man.

- Is black with large lips and rastafari hair to imply a connection with Jamaican culture and the credo of fighting the power.
- Speaks with a humorous, cheesy, laid-back Jamaican accent.
- Wears a lab smock to show his scientific background.

The Superintelligent Monkeys Rely on the comic juxtaposition of monkeys and intelligence, drawing heavily on the popculture archetype of the intelligent monkey as exemplified in e.g. Planet of the Apes, Spaceballs, Futurama, etc.

- Wear smocks and glasses to show their extreme intelligence and scientific proficiency.

- Speak with a British oxbridge accent to further accentuate their aura of intelligence and refinement.

The Yuppies Represent the inverse ideals of Dr. Bongo's, preferring greed over love and mindless pop over groove.

- Are bland, boringlooking and dressed in formal black suits.
- Until influenced by Dr. Bongo, the yuppies stand in a closed circle, talking to one another, perhaps discussing stock value, fancy restaurants and comparing business cards.
- Once influenced by Dr. Bongo the yuppies show, through their dancing, that they all have heaped up latent groovyness inside, trembling to burst through their facades of jadedness and arrogance.

2.4 Narrative Devices

In order to merge the narrative with the gameplay there are several narrative devices in Doctor Bongo. The basic devices are:

- Character representation - avatar and non player characters
- Background and setting
- Sound
- Game mechanics

The setting is the underlining narrative device for Doctor Bongo. In the yuppie level the setting is a skyscraper in a cityscape in the eighties. The exaggeration and caricature of the city and buildings enhance the comic and cartoonish style of the story.

The characters represented by the avatar, Dr. Bongo, and the NPC yuppies work as narrative representatives of the players actions. The character acts not only according to the player's actions but also according to the narrative - static yuppies become dancing happy/groovy people because of Dr. Bongo's (the player's) influence.

This is also a merging of narrative and game mechanics. The game mechanic indicators like the meters also convey narrative information because it indicates the level of groovyness among the yuppies.

Even though the sound is directed towards the player it is also the voice of Dr. Bongo. The audio feedback gives the player an indication on how well she is doing but also comments on the narrative progression.

3 Gameplay

Doctor Bongo features a combination of rhytm and puzzle gameplay. Through time pressure the game puts the player in the position of frenetically keeping up with bongo rhythms while at the same maintaining an overview over different states of the level. This makes especially the inexperienced player create implicit heuristic models for solving puzzles, while the experienced player on any given level eventually will attain a transparent understanding of the level. Making the game fun to play for the casual gamer but offering greater depth to the enthusiastic gamer.

3.1 Gameplay Description

The basic idea of Doctor Bongo is for the player to hit or tap buttons at the right time, as indicated by a rhythm. Playing well makes her win the game; playing poorly makes her loose the game.

The player starts on a level composed of nodes and hidden vertices between these. The vertices show which nodes affect other nodes. Each node has a target interval, and the player is capable of raising the level of a node by selecting that node and then playing at that node. The goal of the game is to have all nodes at their target interval at the same time. The player selects a node to start on and a short beat is played to the player to inform that this is what she is supposed to play.

The beat continues to play, but now the player is supposed to make a key input at each sound, a timeline will indicate to the player which key to press and when. If the player succeeds a beat (composed of 5 to 9 sounds and lasting 4 secs), the nodes value is raised. All nodes connected to her current node are also raised in value. Nodes not connected to her current node will fall a bit in value. If she fails the beat, all nodes will go down in value.

Because nodes affect each other, it's not completely obvious in which order or how many times a player should play at each node. And she has to consecutively play a bit at each node to get all nodes to their target interval.

We want the player to experience slowly building up a song by combining the different beats of the level, until he has made a groovy tune that completes the level.

3.2 Controls

The controls consist of four different areas that correspond with four different drumbeats in the game. The main control device is intended to be the Nintendo Gamecube Bongo-Controller for Donkey Konga. The secondary device will be a normal keyboard. In the case of the keyboard we will section it into different areas, each representing a different single drumbeat, this can be seen on figure

1. In the case of the main controller, the bongo drum controller, we will map

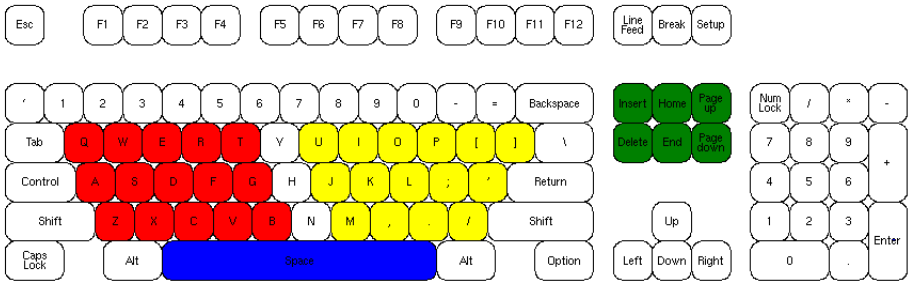


Figure 1: Keyboard controls

a drumbeat at the upper/large part of drum and a level/node button in the lower/smaller part of the drum. The game must thereby be played with two of these controllers, which then includes four upper beat areas and four lower level change areas, this is shown on figure 2.

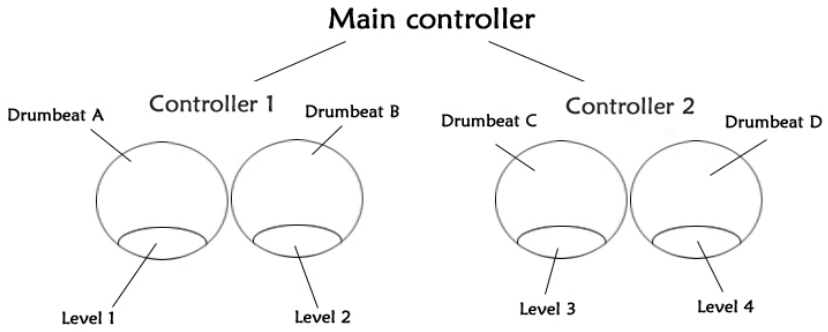


Figure 2: Donkey Konga Controls

3.3 Interface

The interface consists of several types and areas of feedback, as shown on figure 3 in section 5.2. The most important feedback is the timeline located at the bottom screen, which show the different kinds of beat (represented by coloured spheres) and when to hit each of them. The hit indicator on the timeline, represented by two small arrows indicates the "exact" time to hit each beat. The spheres give a visual feedback either:

Positive - you hit at the correct moment - green tick

Negative - you missed beat - red X

The movement between floors/nodes is indicated by the picture/avatar of Dr. Bongo. Each floor has a score indicator represented by a meter on the left side of the floor. The meter has a danger area beneath the line and a highlighted area where the floor score should be at. The different areas on the meter are further indicated with use of colours:

Red danger area, this is beneath the line and above the target area.

Green safe but not in the target area.

Yellow target area.

Correct playing and advancement is also indicated through the use of character animation. The avatar of Dr. Bongo located on the right side of the screen moves his hands every time the player plays two sequences of beats correct. In addition the yuppies on each floor start dancing each time two yellow fields on the meter is filled. This is done by playing correctly, giving a more character and narrative driven feedback. In addition there is a time indicator in the upper right corner.

3.4 Rules

Doctor Bongo is a crossover of two game genres - Puzzle and rhytm. So the rules applying to each of these elements have been placed in each their section. There are however some interaction of these two types of mechanics. Which will be explained here. Followed by a in depth analys of the two mechanics. Before you continue, you might want to read section 3.1.

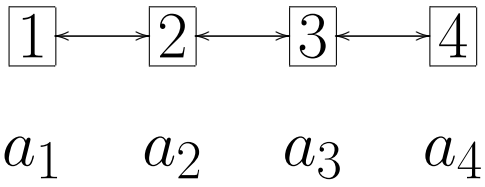
The game is equivelant to solving a set of equations as shown in theorem 3.1, where the gameplay is to input your solution by playing at each node the correct number of times. The hard part of the game is uncover what this hidden equation really is, and to adjust to the changes in the equation as time goes. As the reader might notice, most of the rules of the game is concerning the puzzle part, and emphasis will be put on that part in this text.

Rhythm Rules

When the player arrives at a node, a short beat will be played and shown on a timeline to inform the player what he is about to play. After this the player is supposed to repeat the beat. Everytime he succesfully does this. the current and adjacent nodes will rise in value. If the player presses the wrong button as input or at the wrong time, the beat will fail and all nodes will fall in value.

Puzzle Rules

What follows here is a general mathematical description of how our puzzle work, except for the fact that we're working with four nodes, but that also be generalized. If the reader is uncomfortable with advanced math, she is advised to read section 3.7, which serves as an example with real values instead of variables.



The nodes 1 to 4 are the four points at which the player can play his bongo in order to complete the level. The intervals $a_1, \ldots, a_4$ stated underneath is the target she wants the current node to reach, when all 4 nodes are in the right interval, the level is completed. The arrows indicate how the different nodes affect eachother. The player can move between the nodes, everytime he goes to a node, a beat will start playing and player repeats the rhythm. Everytime he repeats the rhythm correctly, his current node will rise in level. The adjacent nodes will also rise as indicated by the arrows. For example: If the player is at node 3 and repeats the rhythm perfectly, Node 3 will rise with k, and node 2 and 4 will rise with l. If he repeats the rhythm one more time, Node 3 will rise an additionally k and node 2 and 4 an additionally l. Everytime the player repeats a rhythm, the nodes not affected will fall in level. In the above example this means that node 1 will drop m in level. This sort of mathematical behaviour can be expressed with algebra and is done so below.

Theorem 3.1 (The Great Formula) *Let $k, l, m \in \mathbb{N}$, where k is the value the active node will rise when a correct beat is played, l is the value adjacent nodes will rise, and m is the value the remaining nodes will fall and the amount all nodes will fall on a wrongfully played beat. Let i =1,2,3,4. Furthermore let a_i indicate the goal value of node i, and x_i indicate the amount of correctly played beats in node i. Also let $y \in \mathbb{N}$ indicate the amount of wrongfully played beats. Clearing the skyscraper level is equavalent to solving the following set of equations.*

$$\begin{aligned} a_1 &= kx_1 + lx_2 + mx_3 + mx_4 - my \\ a_2 &= lx_1 + kx_2 + lx_3 + mx_4 - my \\ a_3 &= mx_1 + lx_2 + kx_3 + lx_4 - my \\ a_4 &= mx_1 + mx_2 + lx_3 + kx_4 - my \end{aligned}$$

This immediatily gives a bit of knowledge. Applying a bit of linear algrebra to the matrix, gives the following results. There are more variables than equations, so there is an infinite number of solutions. But since we're looking for positive solutions only, this will limit the the amount and also make some combinations impossible to solve. However, the mistake variable y is sort of a dummy variable, and the player will only have one possible solution based on the amount of errors he's made so far. Once we replace $a_1, \ldots, a_4$ with intervals we can widen the solution space. Looking at y as a fixed value makes the number of variables equals the amounts of equations and we willl get exactly one solution if the determinant is different from 0. The above formula also makes it possible to design node values by choosing values for x_1 to x_4.

We have the following variables to adjust the difficulty of a level

- The rhythm played at each node and how precise the player has to repeat it.
- The target interval at each node.
- The way each note affect other nodes.
- The fade time, i.e. the speed at which nodes not affected by drumming decrease in value.

3.5 Scoring, Winning- and Losing Conditions

The scoring system in our prototype is rather simple. Your score is just the time it took the player to complete the level. A possible bonus feature that could be added to the level, is where certain nodes at a certain short time interval would yield a bonus time, hence giving the player more time to complete the level, and increasing their score.

You win the game when all nodes are at the correct level. In the skyscraper level, this means each node is between a score of 60 and 80.

You loose the game if any of your nodes reaches a preset minimum or maximum, in the scyscraper level this is set to -10 and 100 with $m = -1$. This forces the player to play at different nodes in the beginning of the level in order to get their score above 0, so the player can focus more on specific nodes. You also lose the game after 5 minutes of play. These two loosing conditions have been balanced based on calculations in section 3.7 and with ing.

3.6 Modes

Since the gameplay in single player mode quickly can become repetative in our game, we have a few visions to increase replaybility of Doctor Bongo. None of the following ideas have been implemented in our prototype or playtested. This will of course need to be done in any later iteration.

Level Editor

We want to give the player the possibility to create their own levels. By being able to choose from a large array of backgrounds and feedbacks. Being able to set up the nodes and arrays by themselves. And last but not least, being able to record their own bongo beats to put on the different nodes.

Multiplayer Mode

Since we have 2×2 bongodrums as controllers it is possible to play the prototype in a cooperate mode. But we haven't playested this mode much. A cooperate mode would probably require that the beats were designed to be played by two players on each their controller in order to be really fun.

Another option is a competative mode, where the players go up against each other on the bongo-controllers. Levels for this mode could involve players moving around on the same graph and affecting the same nodemeters in different directions. The player who first gets a meter maxed wins. Another option is that the players have each their graph to move around on, and the player who first completes his level, wins the game.

A problem with the multiplayer mode is that the two player's beats will overlap each other, and thus the sounds scheme will become distorted, and it will be harder to distinguish ones own beat from ones opponents. This can perhaps be used to our advantage by using it as a form of catch up logic, where the winning players sounds gets lower and the losing players sounds louder. In a graded fashion.

3.7 Levels

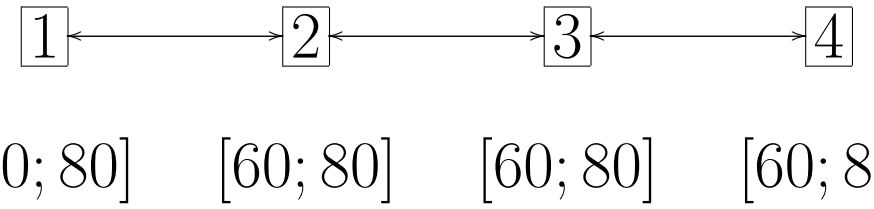
We have only implemented one level in our prototype, and that is the skyscraper level. But as theorem 3.1 shows it is easy to generate new levels. The only additional thing you need to make is the narrative of the level, the graphics and the bongo beats. In the following we will try to explain how the skyscraper level is designed, and how its possible to do a lot of balancing of the level, before we even playtest it, using theorem 3.1.

Skyscraper Level

Dr. Bongo has warped back to the eighties and is disheartened by seeing all the yuppies have abandoned the ways of love, ganja and groove for individualism, cocaine and synthpop. Deciding that he must help the poor yuppies get back the groove, he enters a fivestory skyscraper and starts drumming ...

At the end of the level the combined dancing of the yuppies puts the building into a state of cascading resonance amplification and the building succumbs to its own vibration and comes crashing down.

The level layout is as follows with the desired state of each level being between 60 and 80. And where you loose the game if any meter goes below -10 or above 100. You also loose the game if you do not complete the level within 300 seconds.



In this level the current nodemeter rises by 8, connected nodes by 2 and non-connected nodes drops by 1 after a succesfully played beat. All nodes drops by one if you fail a beat. This means that $k = 8$, $l = 2$ and $m = -1$. The initial state is that all meters are set to 0. So solving the skyscraper level is equal to solving the following equation.

$$\begin{aligned} 60 + y &= 8x_1 + 2x_2 - x_3 - x_4 \\ 60 + y &= 2x_1 + 8x_2 + 2x_3 - x_4 \\ 60 + y &= -x_1 + 2x_2 + 8x_3 + 2x_4 \\ 60 + y &= -x_1 - x_2 + 2x_3 + 8x_4 \end{aligned}$$

where $y \in \mathbb{N}_0$ is the amount of wrongfully played beats.

By solving this eqution we can extract the amount of times a player has to play at each node. These solutions will often not be integers, which is the reason we made the target value of a node an interval, instead of an exact value. The following table shows how the solution[1] to the puzzle changes as the amount of wrongfully played beats increases.

Mistakes	0	1	2	3	5	10	15	20	30	50
Node 1	7.8	8.0	8.1	8.2	8.5	9.1	9.8	10.4	11.7	14.3
Node 2	5.2	5.3	5.4	5.5	5.7	6.1	6.5	7.0	7.8	9.6
Node 3	5.2	5.3	5.4	5.5	5.7	6.1	6.5	7.0	7.8	9.6
Node 4	7.8	8.0	8.1	8.2	5.7	9.1	9.8	10.4	11.7	14.3
Total	26.0	27.6	29.0	30.4	33.4	40.4	47.6	54.8	69.0	97.8

[1]These where calculated using the matrix solver at http://mac6.ma.psu.edu/lin_equations/index.html

The Total row shows the total amount of beats (including mistakes) needed to be played in order to complete the level with that amount of mistakes. Thus we see that after 30 mistakes, it takes $69 \times 4sec \approx 5min$, to complete the level.

The above numbers, gives a very good indication on how the level was created, but we also used playtesting, to conclude that our values where right, and gave a fun playing experience.

4 Marketing information

In this section we will try to give an overview of who plays and ays for Doctor Bongo. By analysing some of the key features of our game.

4.1 Target Audience

Doctor Bongo is intended for a relatively wide audience. The game should be enjoyable to the casual gamer as well as the enthusiastic rhythm game fan. And the games multiplayer capabilities should make it a good choice for party games. The humoristic style and tone of the game, implies that it should be marketed to the 18+ segment, as the references to rastafari culture, the yuppies of the 80's and not least, the ganja, put certain requirements of maturity on the player of the game. However, we should expect players below 18 to get a hold of the game and playing, possibly even with their parents accept.

4.2 Platform

The game should be designed to run on all current generation consoles, ideally equipped with suitable custom controllers. The prototype is developed for PC.

4.3 Top Performers

There are, to our knowledge, no games available that combine the aspects of both rhythm gaming and puzzle gaming. As the most noticeable aspect of Doctor Bongo is the rhythmic playing and the puzzle part has a more subtle role, the top performers should be drawn from the rhythm/party game genre. Here the main alternative choices in games to consider are the Donkey Konga titles.

Other top performers (and alternative buying choices) include the titles: Guitar Hero, UmJammerLammie, Vib Ribbon, ReZ, Guitaroo Man, various dancing games. And other party games as Singstar or EyeToy.

4.4 Feature Comparison

As noted above, the rhythmic part of Doctor Bongo's gameplay is present in many other titles, and some indeed the degree of refinement of the mechanic found in

Doctor Bongo. However, the puzzle part of Doctor Bongo sets it apart from its contenders and provides a number of unique features including:

- The requirement for overview and planning.
- Experiencing the development of greater insight into the mechanics and rules of each level.
- The possibility of creating new beats, levels and scenarios ...
- ... and sharing them with others around the world.

4.5 Sales Expectations

Given the target audience we should expect our main buyers to belong to the segments that Bateman & Boon (2006) classify as the lifestyle and family segments, which amount to circa 10 million and 30 million individuals respectively per market region. Though these markets are fairly large, penetration of any given title in these segments is likely to be low, 10% and 5% for lifestyle and family segments respectively.

Given that the market is already saturated with top performers and strong brands in rhythm games, the best strategy will be to market the game with an emphasis on humour and puzzle elements. Should such a strategy succeed, 10% and 5% penetration rates should probably be considered best case scenarios. And only with a full crossplatform launch.

5 Technical Spec

This section is a reference on how gameplay elements should be implemented in the product. Section 5.1 will deal primarily with development within Gamemaker, but other development platforms could be considered. In section 5.3 the technical specs of the current controller will be described, and the section will consider the possible creation of a game-specific controller as well.

5.1 Development Platform and Tools

Audio Tools

In the creation of auditive media for the production, a tool capable of isolating and indexing the musical elements of the beat should be used to ease the hardcoding of success/failure criteria of each player-stroke with regards to each individual beat. The BPM of each beat is predefined, and the preferred tool should be able to index the timing of each drum-stroke. Furthermore the tool must contain a feature to isolate the individual instrument within a given percussion-set. A

possible tool, which was used in the prototyping, is Reason by Propellerhead Software [2]. Beware of "low-end" tools e.g. Fruity Loops by FL Studio [3] since triplets, e.g. to play three beats of identical time-spacing within a 3/4 time signature in a 4/4 time signature, are not possible to create with exact precision.

Development Platform

Since no gameplay elements of the game demands out-of-the-box technical solutions, most developement platforms/scripting languages can be used. A good tool for prototyping is Game Maker by Mark Overmars [4] , since it contains a timeline functionality which is quite useful in the creation of a rhythm game. Issues with synchronisation should nevertheless be looked out for, since the sequential loading of elements is not completely transparent within the Game Maker platform. See appendix B for more information about the sync issue.

5.2 Game Engine

This sections deals with the implementation of the issues described in section 3 from a more technical point of view. To give a better understanding of the following text, a short description of the visual elements is listed below

We have the following elements in our gui in figure 3.

1. The status bar / volume meter where the player is presented with an indication of how well she is doing on each floor.
2. The floors, where the company employees dances, which is directly related to the volume meter.
3. Time countdown to show how long the player has left to complete the level.
4. Doctor Bongo playing the drums whenever a player-stroke was successful.
5. The timeline which shows which drum to stroke at a given time.

In the following section the different elements of the game engine will be elaborated upon:

Score: The score is as stated in theorem 3.1, divided into subscores pr floor. If a beat is played correctly, points should be added to the current floor and the adjacent ones. To figure out if a beat was played one needs to take a look at each individual drum stroke and investigate its state. Subsequently the list of beats should be run trough at the final time step to validate the overall success-state.

[2] http://www.propellerheads.se/
[3] http://www.flstudio.com/
[4] http://www.gamemaker.nl/

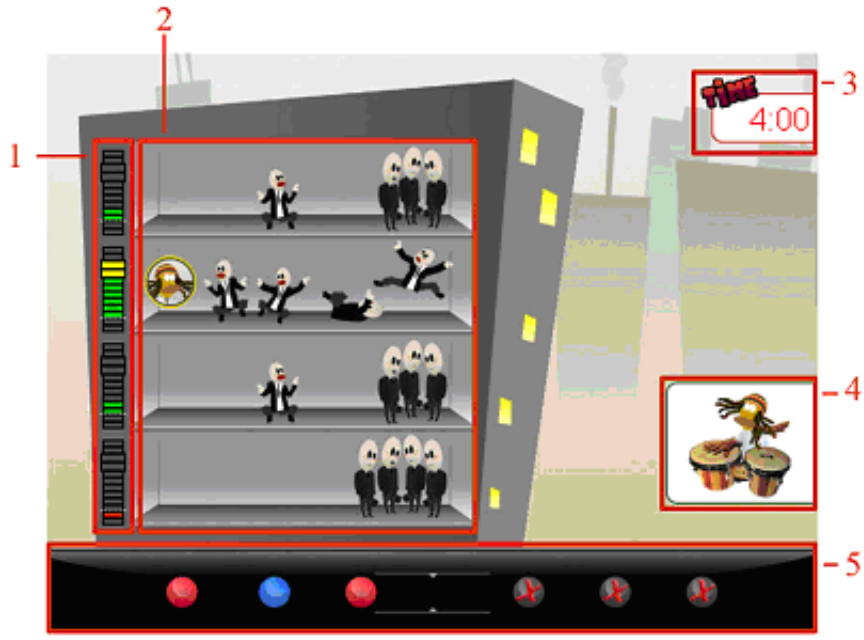


Figure 3: Screenshot from the game indicating various visual elements

```
beatComplete = true
for (i = 0; i < ds_list_size(global.bongoBeatDSList); i+=3)
{
 if (ds_list_find_value(global.bongoBeatDSList, i+2) == false)
 {
  beatComplete = false;
  break;
 }
```

Function should be called to test on the validity of timing/key-identity versus the timeline position and the information stored regarding the current beat. In the case of a successful stroke, the Boolean value contained in the beat-state array should be set to True.

Timing: A timeline on 120 units should keep track of when to create a visual representation of the beats, i.e. the coloured balls in the timeline, by comparing its current position with the value stored in an array of beats for any given floor. Since the player should have an indication of which button to press, it is necessary to create the ball instance at a given time before the actual beat. In the current iteration of the prototype, the offset is set to two seconds, but testing should be done to verify the timing with regards to player perception.

```
bubbleCenter = 16; //Half the width of the ball
bubbleAbsScroll = 300; // Distance on each side of marker
bubbleStart = (room_width/2)-( bubbleAbsScroll+bubbleCenter);
bubbleEnd = (room_width/2)+( bubbleAbsScroll-bubbleCenter);
```

```
bubbleSpeed = 6; //Pixels pr frame
bubbleOffset = round(bubbleAbsScroll/bubbleSpeed); //Timing offset
```

The code above is an example of proper positioning of the ball, and how to calculate the offset to ensure that the ball arrives at the centre of the game-screen at the correct time. It should be noted that problems may arise with non-integer results since we are working in absolute pixels, so rounding should be applied which will, in the worst case, only result in an inaccuracy of 1/30 sec. The ball-instances should be kept track of in an array and tested whether or not their positions are outside the region of interest. In that case they should be deleted from the array, and the instance destroyed.

The balls must have the ability to change appearance based on the success/failure of the player. This is done simply by exchanging the sprite of the instance to reflect its current state. Even though an array keeps track of the states of the beats, there is no direct connection to the list of ball-instances which means that a method must be applied to ensure that the correct beat state is presented for each individual ball. This is done by a cross reference between the ball currently at the centre of the screen, and the current time of the timeline in conjunction with the timing of the beat.

The following code example is a guide to implement the method described in the above, although creating another method could be advantageous in a sequential iteration step.

```
if(ds_list_size(instanceList) >= 1)
{
 lastObject = ds_list_find_value(instanceList, 0);
 timeLinePos = Bongo.timeline_position - offset;
 dsListPos = ds_list_find_index(bongoBeatDSList, timeLinePos)
 boolValue = ds_list_find_value(bongoBeatDSList, dsListPos+2)
 instanceAtCenter = (roomCentre)-bubbleCenter)+(offset*bubbleSpeed)-32;
 panelLines.sprite_index = spr_panel_lines
 for(w = 0; w < ds_list_size(instanceList); w+=1)
 {
  if(ds_list_find_value(instanceList, w).x < 405 &&
  ds_list_find_value(instanceList, w).x > 395)
 {
  panelLines.sprite_index = spr_panel_lines_on
 }
 if(abs(ds_list_find_value(instanceList, w).x-instanceAtCenter)<=(offset+1))
 {
  currentBall = ds_list_find_value(instanceList, w)
  if(currentBall.sprite_index!=spr_ball_succes){
   if(boolValue == true) {currentBall.sprite_index = spr_ball_succes}
   else {currentBall.sprite_index = spr_ball_error}
  }
 }
}
```

Sound: The drumbeat sounds of all floors are instantiated at game-start. The beat belonging to the floor in which the protagonist currently resides will

be played at full volume, whereas all other beats will be played at a volume of 70 %. The volume settings should be tested further upon in a later iteration step. Protagonist speaks is controlled by testing the score of each floor against the score in the previous time step, and is tested once every 120 frames. If the increase/decrease results in a change in the predefined intervals a randomly selected speak, which fits the current change of interval, is played. The interval changes that are interesting are in prioritized order;

I. If a floor intensity has fallen below 0.

II. If a floor intensity is to high.

III. If a floor intensity is at the correct level.

IV. If a floor intensity has increased.

The intensity levels are: $]-\infty;0]$, $]0;10]$, $]10;20]$, $]20;30]$, $]30;40]$, $]40;50]$, $]50;60]$, $]60;70]$, $]70;80]$, $]80;\infty[$

Graphical feedback This section will dive into describing the graphical feedback elements and their triggers. The intensity levels are mapped in various ways e.g. *1.* and *2.*, but depict the same value, with *1.* having the highest resolution mapping. The mapping of *1.* is similar to the intensity levels listed in the section above, with a slightly higher resolution at the extremities i.e. that the set of $]-\infty;0]$ is divided into $]-\infty;-7]$, $]-7;-4]$ and $]-4;0]$, and similar for the set of $]80;\infty[$ which is divided into $]80;90]$ and $]90;\infty[$. With regards to *2.*, the resolution of the mapping much more limited. The subsets are as follows; $]-\infty;0]$, $]0;60]$, $]60;80]$, $]80;\infty[$.

In *4.* we have a close up of the protagonist which should indicate strokes being played successful. Whenever a stroke have been validated as successful, as described in the section 3.5, a predefined animation will be played to fit the drum-stroke. The timer (*3*) is simply an indication of the countdown that will end the game.

5.3 Controller Technical Specs

Since the controllers are non-specific to the game, some compromises must be taken. To play the game it is necessary to have two bongos, connected to the pc through a Gamecube interface simultaneousy. Each bongo consists of two drums with two buttons on each drum, making a total of 8 buttons in the entire setup. Four are used to changed floors, while the other four are the stroke sensitive buttons with which to interact with the game.

The possibility to create a game specific controller should be considered. The drums should be individually coloured and marked with symbols to encounter

issues with colour-blind players. The controller should consist of four drums and buttons placed on the side to change floors. These ideas concerning controller creation is preliminary, and in-depth testing should be performed.

A Design Method and Creative Process

We started this project with a general brainstorming session where we talked about different ideas that we wanted to work with. This was a "pitch your idea" kind of session. One of the initial ideas was actually a bongo drum rhythm game. We decided on the other hand to go back home and think about this idea and some other ideas and also to come up with one additional idea. During the second brainstorming session we began from scratch and started with around fifthy verbs which we sorted and rolled a dice two or three times to get different combinations of games and game mechanics. Some of the ideas that came from this process were for example a cow pushing game (push as many as possible of sleeping cows over on the side avoiding angry bulls and farmers in the process) and a monkey game similar to Lost Vikings or Lemmings, were you have different genetically manipulated monkeys that can perform different tasks and you have to choose between them to get through different obstacles. We also stumbeled upon the notion of a bongo drumming game again while playing with the verbs and it became obvious for us that this was the most interesting and fun idea to explore. We then proceeded to flesh out how this kind of game would be most fun and original. We thought of the possibility of creating a puzzle element alongside the classic rhythm game and did quite a extensive design of the system (both mathematically and conceptually) before we proceeded to begin testing in gamemaker. On the other hand we tested the possibility of the rhythm part in gamemaker to see if it was at all possible to do this kinda game.

After completing these first creative steps our main source of inspiration became informal and formal playtests.

Throughout the production of our prototype we of course underwent countless sessions of informal playtesting. This section, however, focuses on our formal playtests using subjects totally or partially naïve towards the game. The information we gained from these sessions we considered practically invaluable for making informed design choices. This appendix describes the most important results gathered from playtests and the design decisions they produced.

Our playtest results had the most importance for design decisions of the following categories:

A.1 Visual Feedback

Players want and need a lot of visual feedback when playing any rhythm game - and especially one with complex rhythm elements. Our players were very dependent on the coloured spheres on the time line and seemed to use these more than the actual rhythms as depicted in figure 4.

Players, and particularly novice players, have a highly limited visual bandwidth when playing a game which requires a high degree of timing and coordination of off-screen actions (i.e. hitting the right areas of the keyboard/the right

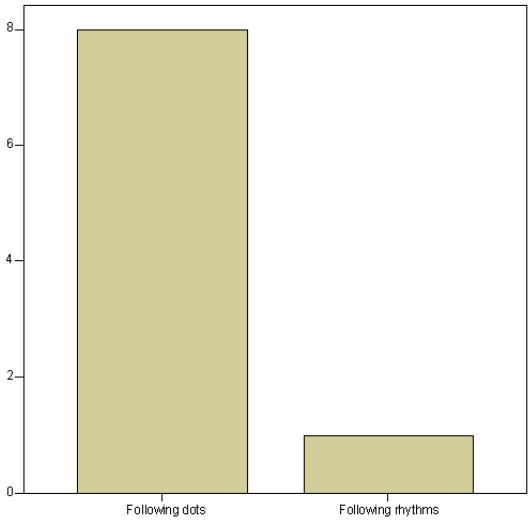


Figure 4: Initial Strategy

bongo drum). There simply isn't time to scan all areas of the screen, when you also have to keep an eye on the timeline and the controls you're using. This points to the necessity of having the most important visual feedback in the player's primary field of vision - the timeline. And it also points to the fact that it is a good idea to make visual information in other parts of the screen highly visible and preferably redundant (see Appendix C).

This information had several implications for our design: We estimated that the bottom part of the screen taken up by our timeline was already saturated with information. Since novice players were so highly dependent on the information from the line we would not risk adding any confusion by adding more information. Therefore we decided to put the meters describing the states of the individual node alongside the nodes and use the animations of the dancing yuppies for redundancy letting them mirror the state of the node. Also we decided to look into possibilities of giving the player feedback on the state of the nodes through audio in addition to video.

A.2 Auditive Feedback

A feature added primarily on the basis of playtest results was the audio feedback on node states. As noted above novice players had little surplus of visual attention to keep an eye on the meters in the left part of the screen. Also, during interviews, players in a general way requested more auditive feedback Therefore we chose to make the feedback on node states further redundant by adding short speaks by Dr. Bongo, commenting on changes in the states of the individual nodes.

However, the importance of the rhythms should of course not be downplayed.

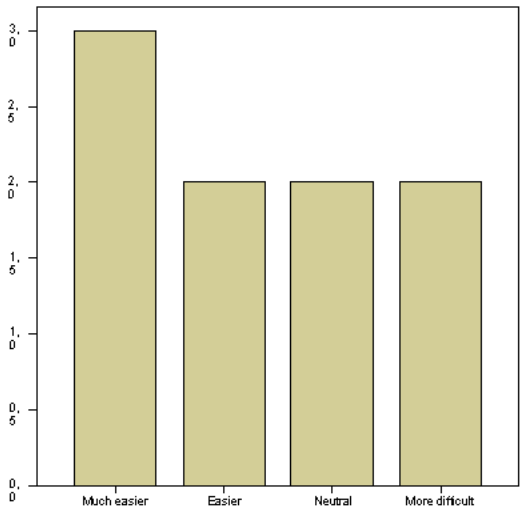


Figure 5: Effect of coloured dot.

Our playtest reports showed that all novice players but one started playing by the time line, but shifted to playing primarily by the rhythms once they had some experience, as shown in figure 4, where we asked our test subject whether they changed their strategy during the course of the game, as seen in figure 6. The one person who answered no started by playing primarily by the rhythms and did not switch to playing by the timeline during the course of the game.

A.3 Controller Choice

Also our choice of controllers was motivated in part by playtest feedback. Initially we did not know of the Nintendo Donkey Konga controllers, but already by our first playtest our players were asking for more intuitive hardware for interacting with the game. We did some experiments with zoned keyboard controls and a dance mat to try to accommodate out testers' very understandable wish, but once we discovered the bongo controllers we immediately chose to shift to using them which our subsequent playtesters responded positively to.

A.4 Level of Difficulty and Tweaking

A clear conclusion that could be drawn from our observational data from our playtests was that the topmost node in the skyscraper level is significantly harder than the other levels. Most playtesters, once they had a basic grasp on the game, showed a pattern of returning to this node, becoming frustrated, playing on the easier nodes for a while before returning and so forth. This made us consider lowering the level of difficulty of this beat, but at the same time the

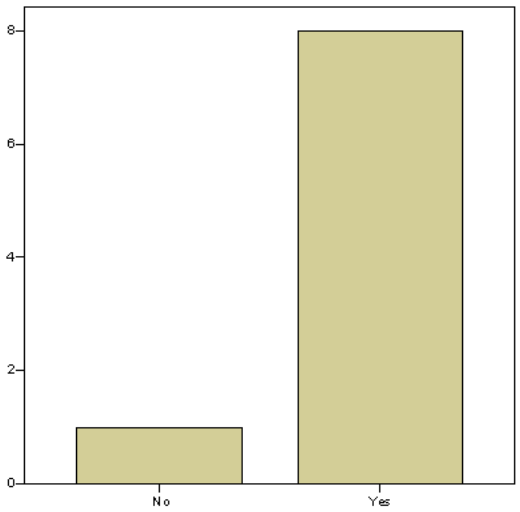


Figure 6: Change of strategy.

vigorous way in which our testers kept returning to this node suggested that our players considered the 4th node a challenging element. Eventually most testers mastered even the 4th node and responded with reports of great satisfaction on reaching this level of play. With this in consideration we chose to keep the large imbalance in the difficulty of the nodes, since the 4th node provided players with a 'major challenge' overshadowing the other nodes. Actually this seemed to provide a greater sense of completion once the level was finished - no players directly reported this, but we inferred it from our observations. Ultimately this became the deciding factor.

In addition many small features were tweaked through a combination of informal playtests and small comments and observation from the formal playtests, but the ones documented above show the value of systematized playtesting. See results and notes from individual playtests in Appendix C.

B Bugs

During the iterative design process, a few bugs have arisen, which have not been corrected in the current iteration step. It is a conscious choice since the correction of these demands a complete rewrite of the game engine. If the development of the game should continue, it would call for a complete roll-back in which the entire code would be rewritten. This section will account for three major bugs, which should be focused on in subsequent iteration steps.

Button mash

In the current state of the engine it is possible to "Button-mash" i.e. hitting the buttons repeatedly with no thought of timing. This is possible since each stroke calls a function which checks the validity of the timing, but in case of an error, the game-engine takes no steps to "punish" the player. In case of further game development, this fact should be taken under consideration.

Synch issue

As mentioned in a previous section, the game suffers from a synch issue, which the development not entirely can account for. The issue does not occur at every game-start, but appears randomly. Slower machines have a tendency to have a higher rate of occurrences than high-end computers. Even though the game is quite simple in essence, Game Maker, which is the platform of choice, seems to be quite inefficient and thereby cause the issue to arise. A problem in discovering the basis of the issue is that the Game Maker platform is quite non-transparent e.g. it is not obvious how the loading of game object e.g. sounds and graphics occurs. The game should probably be scripted from scratch in a later iteration step, which should dramatically increase transparency and performance.

Indication feedback

When the balls travel across the screen, it should be indicated if a stroke was successful or a failure. Unfortunately the array keeping track of the true/false state of each beat is not directly connected to the array of ball-instances, which creates problems when indicating its state to the player. A more elegant solution should be implemented in the final product.

C Playtests

We made two formal playtests during the development of the Doctor Bongo Prototype, below are the raw data assembled from these. We also did a lot of informal playtests of the game both by our own team and outsiders.

C.1 Playtest - November 14.

```
Topics of special interest for this playtest:
1. Do the players discover the button-mash exploit?
2. What are the players focusing on - timeline or fancy graphics/puzzle?
3. What are the players focusing on - screen or keyboard?
4. Test of intuition of gameplay.
5. To asses the frustration level of the player.
6. Test of story. Before players commence play, we tell them the story of Dr. Bongo vs.
the yuppies and observe their reaction.
7. Test of puzzle.

Location: GameLab.
2 subjects tested with 2 observers (Christoffer & Hans Petter).
Some noise and minor elements of distraction.
One subject had prior knowledge of the game, including goals, mechanics and progress of
development.

Method:
Subjects played Doctor Bongo v10 on the colour customized keyboard.
Before playing, subjects were given only the background story and a brief instruction on
keyboard mapping and changing floors.
Subjects were observed and notes were taken during play. After play comments were entered
into the notes, ad hoc questions were posed, and the players answered the Playtest
Questionnaire v3.

Summary:
1. Button-mash exploit
No players discovered the button-mash during play.

2. On-screen focus
Players were highly focused on the bottom bar with moving dots and reported having little
or no attention left for focusing on meters or graphics.

3. General focus
Players were intently focused on the screen, only from time to time diverting their gaze
to the keyboard.

4. Intuitiveness of gameplay
Players had no trouble understanding the functions of the rhythms and dots. However, the
player with no prior knowledge of the game did not grasp the puzzle element of the game.
His understanding of the game goal was that all meters had to be maxed out in the red
level and he seemed to think, that he had to go from level 1 to 4 in a numbered order.
He did not notice that the different levels influenced one another. As the second player
had prior knowledge of the puzzle mechanics, he is disregarded here.

5. Frustration level
Low on levels 1-3, high on level 4.

6. Test of story
Players enjoyed the story and responded with amusement and laughs.

7. Test of puzzle
See intuitiveness above. No players finished the puzzle due to level of difficulty of level 4.
Had the players been able to play level 4 it seems plausible, though unproven, that they would
```

have maxed out all meters and waited for the puzzle to complete by reduction of points. This strongly suggests that maxing out one or more meters should be made a losing condition. Possibly by adding the mechanic of lives to the game.

Primary play strategy - dots or rhythm
Players started by focusing on the dots. One player reported shifting to listening to the rhytm early in the game, though still giving most or all of his visual attention to the dots. The other player reported staying focused on the dots. This suggests that attempts should be made to have most or all of the games feedback arise from the bottom bar of the screen, since most or all of the player's visual attention is focused here.

8. Other notes
Both players reported that the keyboard mapping of level 4 was problematic. They would have like to have the green zone in the middle. This seems to an issue specific to each rhythm which implies that controller/colour mapping should be tested for each rhythm individually.

The player with previous knowledge of the puzzle mechanics noted, that even though he knew that the winning condition was to have the meters in the yellow zone, he consciously tried to keep all meters in the maximum red level. This he explained with the fact that he always tries to achieve the best possible score in any game, and since he saw maxing out the meters as a representation of doing your best, he consistently pursued this goal and tried to maintain all levels maxed out, while struggling with level 4. This suggests that the meter solution may cause some trouble for players with a particularly 'achieving' style of play.

Players were baffled when meters took on negative values and didn't understand why values didn't rise when they played correctly.

Finally, players noted that they would like more visual and auditive feedback, one suggesting a "winning sound" when reaching the desired level on a floor, another suggesting intensity of sound or volume level going up along with the meters.

Raw notes from playtest 14-11-2006
PQ results (qualitative answers omitted, integrated in text above)
Question: Player1,Player2
Musical experience: Y,Y
Musical skill: 2,2
Easy rhythms: 0,1
Quickly grasped idea: 1,2
Fun while playing: 0,0
Potential of fun: 2,1
Rhythms slow of fast: 1,0
Following rhythms easy or difficult: -1/2,-1
Coloured dots => easy or diff.: -2,-2
Beginning played by dots or rhythm: D,D
Changed strategy during play: Y2,N

Player 1
Premise: only story as instruction
High focus on screen
Rhythm 1 seems easy
Stays on level 1 until intervention
Seems to think he has to build up from bottom of screen to top, adding from one meter to the next
Adds up on meters all the way up to red spontaneously
Generally good
Rhythm 4: Laughs on change to last level due to high difficulty Beginning frustration
Stops due to frustration
Q: What is the relation between the different meters?
A: Didn't notice relations between meters would like feedback from bottom line
Feedback: Level 4 is to difficult Mapping difficulties from rhytm to keyboard placement of zones
Red cross zap to "OK" is confusing Wants feedback on completing floor - e.g. floor Meter placement is problematic
Feedback & sound: More sound in - more sound through feedback More intensity as levels gain
Spontaneously thinks that it is a one-way up from floor to floor

Player 2 (Mike)

Quickly grasps idea due to previous information about game
Easily plays rythms 1 and 2
Frustrated by level 4
After ca. 1 minutes practice almost plays 4 perfectly, but gives up due to frustration
Goes to level 3 instead
Seems to become more intently focused on keyboard over time
Intuitively goes for putting all meters into the red max
Returns to level 4 - plays almost perfectly, but consistently misses one (maybe due to hardware problems)
Turns to maintenance of levels 1-3
Only leaves levels when they are maxed out to red
Seems to be frustrated by level 4, pressing on but not reaching any correct streaks (and indeed, if he did would not see any points due to negative value). Reacts with frustration to level 4 when ending game session.
Doesn't like placing of zones combined with rhythm on level 4.
Finds other levels very easy - and with good mapping/key placement.
Doesn't understand why level 4 doesn't come up through relational influence.
Plays up to red strategically, and ads that he likes to reach max score, so he plays up to red no matter what..
Preferred the dance mat, when he tried that.

C.2 Playtest - November 28.

Topics of special interest:

1. Button-mash exploit.

2. What are the players focusing on. Timeline or fancy graphics/puzzle.

3. What are the players focusing on. Screen or keyboard.

4. Test of intuition of gameplay.

5. Frustation level.

6. Test of story. Before players commence play, we tell them the story of Dr. Bongo vs the yuppies.

7. Test of puzzle.

Playtest results

28-11-2006

GameLab?. 7 subjects tested with 5 observers. Some noise and minor elements of distraction.

All subjects had some prior knowledge of the game, including goals, mechanics and progress of development.

Method:

Subjects played Doctor Bongo v13 on the 2 colour-coded DK Bongo Drum Controllers giving 4 drums totally.

Before playing, subjects were given the background story and a brief instruction on controller mapping changing floors.

All subjects went through multiple play sessions/trials.

Subjects were observed and notes were taken during play. After play comments were entered into the notes, ad hoc questions were posed, and the players answered Playtest Questionnaire v0.3 (PQ_0.3)

Summary:

1. Button-mash exploit
2. On-screen focus
3. General focus
4. Intuitiveness of gameplay

5. Frustration level
6. Test of story
7. Test of puzzle
8. Primary play strategy - dots or rhythm
9. Other notes

PQ results (qualitative answers omitted, integrated in text above)

Question: Player1,Player2,Player3,Player4,Player5,Player6,Player7

Musical experience: Y,Y,N,Y,N,Y,Y

Musical skill: 2,2,1,3,1,2,2

Easy rhythms: 1,2,-1,2,2,1,2

Quickly grasped idea: 0,1,1,2,2,2,1

Fun while playing: 2,2,1,2,2,2,2

Potential of fun: 2,2,2,2,0,2,2

Rhythms slow or fast: 0,0,0,1,0,0,0

Following rhythms easy or difficult: 0,0,0,1,1,1,1

Coloured dots => easy or diff.: -1,0,-2,1,-1,0,1

Beginning played by dots or rhythm: D,R,D,D,D,D,D

Changed strategy during play: Y3,Y2,Y2,Y3,Y3,Y2,Y2

<b>RAW NOTES<b>

Playtest 1.

Mads INexperienced tester. Basic instruction in use of controllers 4 bongo setup Windowed mode
Distracting comments from team

Playsession 1

High focus on bottom bar

Laughs when changing to level 4. Comments level 4 very hard.

Fails game, but wants to try again.

Is informed about winning condition by team.

Didn’t intuitively grasp the concept of the dangerous low levels in first playthrough

Playsession 2

goes almost directly to level 4

Becomes frustrated over difficulty level and leaves level 4 for level 3.

Likes the rhythms of the game.

Comments than when the easy rhythms are reahced there is attention in surplus to focus on other things.

Is beginning to grasp the loosing condition of too low level values.

Playsessions 3

Starts at level 1 - 1 up

MOves to level 2 - 1 up

Moves to level 3 - 1up

MOves to level 4. Gives up due to fall of level 1.

Moves to level 4 to replenish value

Almost got rhythm now. completes several beats. Yellow dot bug confusing,

At 2.45 left starts to go for achieving winning condition

Goes to level 4 to achieve winning condition, but other levels fall

Goes into maintenance of levels

time runs out before winning condition attained.

Interview

Is informed of yellow dot bug on level 4 which gives some enlightenment

Comments on lack of auditive feedback on beats played wrong.

Comennts that level 4 is a rhythm different from the other ones, which increases difficulty, but which is good.

Player doesn't notice the relation between the different levels of the building.

Poses questions about and is informed about optimal strategies of game.

Playtest 2 Alex Inexperienced tester. Full-screen mode Informed of goals, winning condition and losing conditions. Also informed of relations between different levels. Yellow dot bug info of level 4.

Before start - comments on much information

PLaysession 1

Small controller confusion, cleared up by instruction

Frenetic play style.

Intervention by team to inform of controller use.

Restart.

Playsession 2

Goes to level 4 head on - by coincidence?

Grasps lvel 4 reletively quickly but becomes fixated on level 4 and looses due to other levels.

Playsession 3

Goes directly to level 4 again

tarts to act strategically toward level of individual levels.

Game goes out of sync...

Playsession 4

Goes directly to level 4

Plays level 4 experly....

Goes to level one to counter negative values

Very high immersion into game, interspersed with short periods of confusion

Adds Extra style to play on level 2 - comments fairly easy.

Changes to level 4, initial confusion

Quckly tries to get into rhytm again, but doesn't complete.

Wants to play again.

Playsession 5

Goes directly to level 4 (perfectionistic play style, takes challenge head on)

Fights with levl 4 until last moment - goes to maintain, raise other levels.

At level 1 inverses internal representaiton of blue and red :)

Gets a little stressed by several levels in the red

Pulls through gets level 4 to max green

At 2.50 starts playing strategically, defensively, maintaining all levels at green values

Raises lvl 1 to max yellow Ignores lvl 3 red until later

Goes level 3 and starts raising values

very immersed

Raises lvl 3 to yellow level

Windows messdae interrupts

Gets all levels 1-3 to yellow

Goes to lvl 4

Is surprised and confused by level 3 going into top red

Quickly sees drop principle.

Almost wins, but loses on time-out just before attaining yellow value.

Interview

Comments that once 4 rhythms are learned, the game becmomes more easy.

Playtest 3 Thanos Inexperienced tester. Full-screen mode Informed of goals, winning condition and losing conditions. Also informed of relations between different levels. Yellow dot bug info of level 4.

Playsession 1

Initial confusion over controllers.

Graps basic principles

Doens't have atention surplus to see levels going into red at time enough.

Didn't initially grasps that lvel change is possible at any time on preference.

Playsession 2

Still some controller use problems.

Goes to level 2 and uses quick tapping exploit.

Some confusion over the way that values of levels rise and fall.

Playsessions 3

Goes to level 1 and graps concept.

Uses quick tap exploit.

Goes to level 4.

Playsession 4

Lvl 3 - successful beats, stays on level while other levels drop

Accidentally goes to levl 4 by hitting the 4 floor button.

Didn't feel attention surplus to keep an eye on levels.

Interview

Feels that losing condition is strange, when you at the same time get OK sounds from Dr. Bongo.

Applauds synch tech and graphics (probably out of pleasing strategy as seemed frustrated over gameplay).

Playsession 5

Fails fairly quickly due to confusion over levels, involuntary chaning of levels.

Some controller problems - recieves instruction from team.

Playsession 6

Much more success with adapte controller handling/use

Some trouble with taking yellow color for green under stress

Interview

Suggests playing the game as a cooperative game.

Reports some confusion over level changing and suggests using different colours to indicate bongos.

Playtest 4

Michael Inexperienced tester. Full-screen mode Informed of goals, winning condition and losing conditions. Also informed of relations between different levels. Yellow dot bug info of level 4. Informed of optimal use of controllers.

Allowed to fool around before actual start of playsessions - to grasp concept of game.

Playsession 1

Cancelled due to techincal problems.

Controller intro repaeted

Playsession 2

Starts on lvl 3

Some intro diff but graps concetpt

Overlooks level 1 falling into red area.

Loses die to drop of level 1.

Surprised of loosing due to loss of points on lvl 1.

Playsession 3

Surprised by difficulty of lvl 4

Frustrated voer diff level on lvl 4

Goes into random tapping, but then gets the rhythm. Looses just as he is getting the rhythm.

Wants to play again.

Playsession 4

Starts on level 3 after accidentally changing floor.

Intently focused on bottom bar, but has resources to keep an eye o meters

Starts playing up all levels.

Starts maintenance strategy.

Quickly gets rhythm of level 4 and starts playing.

Starts playing upl vl 4 towards yellow

Once yellow goes to level 2 to maintain.

Goes to level four and getscontroller mapping wrong - "oh fuck"

Is stressed when lvl 2 suddenly goes into red self killing

Small confusion on lvl1 after lvl4

Attains winning condition intentionally with ca. 40 secs left.

Playtest 5

Lau Experienced user instructed about new controllers.

Playsession 1

quickly graps new controllers. stays away from 4th floor. raises 1st and 3rd floor first. goes to 4th floor after going to red. is smiling ad laughing during play. laccidently changes floor instead of playing the beat, and realises it after 10 secs. Losses the game due to 5min timeout. very close to winning-

Playsession 2

starts on 4th floor. complains about the ball bug. Seems eimmersed into the game. Nods his head to the rhytm. overplays 1st floor. overplays 3rd floor. Losees the game due to impatience about letting the 1st and 3rd floor loose in value.

Playtest 6

Mike experienced user. knows about new controls from watching Lau.

Playsesseion 1

Started on 4th floor. Lost due to 1st floor inactivity.

Playsession 2

accidently plays on the floor button. Starts out by raising 1st floor. then raising 3rd floor. Uses the short break in the beat on 1st floor to get overview. losses due to inactivity on 4th floor. complains its too hard.

Playesession 3

Starts on 4th floor. Prob due to frustration. Losses the game because he can't complete beat on 4th floor.

Playsession 4

Starts on 4th floor. Doesn't seem to grasp the rhytm on 4th floor.

Playsession 5

starts on 2nd floor. Moves to 3rd floor to raise the level on 4th floor. Is very focused on the timeline. Losses because 4th floor is to hard.

playsession 6

Starts on 2nd floor. Moves to 3rd floor to raise level on 4th floor. Moves to 4th floor. seems frustrated about 4th floor. Doesn't listen to the rhytm, only looks on timeline. And can't get in to the beat- Move back to 3rd floor to raise 4th floor.

Playsession 7

We convinced the player to close his eyes and play on 4th floor, he nows managedes to succed 2 beats. restarts game.

playsession 8

starts on 4th floor. Seems to have better grasp of the rhytm. Seems to get better at 4th floor. Losses due to 5 min timeout.

playesession 9

starts on 4th floor. 2nd floor to regulate lover floors. Masters all rhythms, but misses one every now and then on 4th floor. Has a good eye for the decreasing levels. Masters 4th floor fairly well (better and better). Completed!
Playtest 7 Playsession 1

Starts on 1st floor. First time player, didnt grasp puzzle/level decrease whitout instruction. Restart.

Playsession 2

Starts on 4th floor. Struggles with 4th floor rhytm. changes to 3rd which is fairly eaysy. Restarts.

Playsession 3

Starts on 3rd floor. Easier start. Manages a better start. Increas in skill, even on 4th floor. Eyes focus on dots. Overview increased in regards to puzzle/meters. Good controll on all levels.

Completed!

References

[1] Tracy Fullerton, Christopher Swain, Steven Hoffman, *Game Design Workshop*, CMPBooks, 2004

[2] *www.bongohero.gratiswiki.dk*

[3] Robert Messer,*Linear Algebra: Gateway to Mathematics*, Addison Wesley , 1997

[4] Chris Bateman, Richard Boon *21st Century Game Design*, Charles River Media, 2005