

PRCO203 - Individual Report

Jack Griffiths
10547816

Rhythm Fun
Rhythmic Learning™ - www.rhythmic-learning.co.uk
Jacks-in-the-Box - www.bit.ly/jacksinthebox

Introduction	2
Background to the project	2
Client Introduction	2
Project Introduction	2
Requirements	3
Project Requirements	3
Legal Requirements	4
Design and Implementation	4
Design Process	4
Implementation	5
Testing	5
Weekly Functionality Testing	5
Usability Testing	6
Project Management	7
GitHub Repository	7
Weekly Meetings	7
Presentations	8
Conclusion	8
Appendices	8

Introduction

The intent of this report is to bring closure to our group project for PRCO203 - Live Integrated Project. I will talk about the entire development process, from requirement specification to completion, my position in the team and the activities I carried out as Technical Lead.

Our Team, Jacks-in-the-Box, consisted of :

Jack Griffiths : Technical Lead

Aden Webb : Project Lead

Jack Brewer : Client Liaison

As a team we worked out a very rigid and detailed process early on, and our roles and tasks were catered to our personality types. This meant development was a very smooth process.

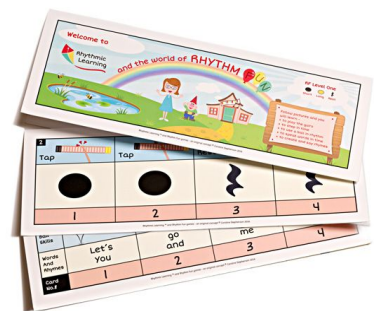
Background to the project

Client Introduction

Our client is Caroline Stephenson, owner of a company called Rhythmic Learning™. The company's main objective is to help children learn using rhythm. She uses various songs and instruments in sessions at schools and clubs that explore different areas of education including times tables and analogue clock reading.

She pitches that there are 8 benefits of rhythm learning for young children aged between 3-10 years old. [*Provided in Appendix A*]

Our Client's existing Intellectual Property (IP) consists of cards that she shows to students, which she then plays along to a rhythm. The children are given wooden instruments which they tap or slide along to a metronome to this displayed beat.



Project Introduction

This project is to create a Rhythm Game App for iOS and Android. This app needs to appeal to young children of 5 to 10 years old, for use by them at home.

The app needed to have a set amount of 'beats' that the user can play though, with increasing difficulty. The interaction will come when the player taps in time with a pre-determined or randomly generated beat. This can be presented with a variety of front end graphics including footsteps and ball bounces. The player will be scored based on their accuracy with a traffic light system score.

The aim of the app is to both aid in education, but also the marketing of the products that Rhythmic Learning™ has available, so the app will need to be consistent with her branding and teaching practices to achieve this effectively.

Requirements

Project Requirements

In order to aid in our development, we wrote a formal specification on our minisite which was approved by our client after our first meeting. This specification was broken down into 5 'Phases' :

Prototype Phase : The Prototype Phase is designed to create a small build of a project to show our initial ideas in a workable prototype. This Prototype had the bulk of functionality planned and created, but not fully polished. *[Breakdown Provided in Appendix B.1]*

Initial Project Phase : The Initial Project begins to give a good example on how the final project will look, as it will start to implement some of the final artwork given to us by Caroline. This was also where we tested and implemented final functionality. *[Breakdown Provided in Appendix B.2]*

Minimum Viable Product Phase (MVP) : The MVP (Minimum Viable Product) is a commonly used term that is used to represent the Minimum Product that can be handed in to our client to satisfy the specification. Our MVP consisted of polishing the existing project, and only a few new features. At this point our client was able to get a much better idea on how the final project will look and function. *[Breakdown Provided in Appendix B.3]*

Minimum Awesome Product Phase (MAP) : MAP (Minimum Awesome Product) is the stage of the project where we will have all the base functionality done, and need to start to think about adding stretch goals and refining the product ready for release. This is also where we would do our usability testing*[Breakdown Provided in Appendix B.4]*

Polishing Phase : This phase was created to cover the 2 weeks we had left in our schedule, to make any final changes and Hand-in documents. *[Breakdown Provided in Appendix B.5]*

These phases were really helpful in breaking down the entire project and keeping ourselves on track with completion. The 5 phases were written up into user stories in our Trello, and hence were easy to refer to when creating new tasks. We kept perfectly on track during most of this process, except from our MAP phase which we ended up extending due to changes in specification regarding the implemented tutorial.

As each phase was 2 sprints(weeks) long (Except from MAP which we had to extend), it was always fluent on the order of which to complete things, as long as it was done by the final deadline of the phase. Each week resulted in a weekly blog post on our minisite on the current progress and how it met the phase requirements, which kept us and the client up to date with what had been completed.

At the end of each phase we also had a meeting with our client, which allowed us to keep our client engaged with current progress and future plans. I created an Android build of the app at all of these meetings, allowing her to give us the best feedback on the final result.

The formal specification was written up by both Jack Griffiths and Jack Brewer, and implemented into trello by Aden Webb (Project Lead).

Legal Requirements

Other than Technical Requirements, we also had a few Legal Requirements put in place before development. Due to the nature of group work, we wrote up a Memorandum of Understanding. *[Details provided in Appendix C.1]*. This has been agreed to by all of the team members, and outlines work hours rules, contact times and what happens if one of the team members are ill or away. Secondly, our client insisted that we put a Non-Disclosure Agreement (NDA) in place before development. *[Details provided in Appendix C.2]* Aden wrote up a great NDA that covered the IP of our clients product, but also allowed us to use this app as a portfolio piece in the future.

Finally, we had a few concerns with the ethic and social requirements of developing an app that would be released for children. This mainly consist of our User Testing Ethics *[Details provided in Appendix C.3]*, and the User Rating Guidelines we would have to meet in order to say our product is suitable for children. *[Details provided in Appendix C.4]*. We came to the conclusion that our client has enough experience in working with children of this age bracket and is an experienced source in previewing and vetting our content.

Design and Implementation

Design Process

Our design process started immediately after receiving our client. From her initial pitch we were able to look into possible designs, features and implementation of her ideas that we could take along to the first meeting as to guide our client as to what we could create. We started with a few moodboards in our group Realtime board so that we could begin to research. *[Provided in Appendix D]*

I made a moodboard on existing similar games, our client liaison researched GUI Design and color pallets, and our project manager created a menu map and some basic UI sketches. When researching similar games, I only had the client's pitch to go from, so I started looking into existing children's rhythm games. I took trailers of these games to our first meeting and Caroline was very helpful in telling us what aspects she did and didn't like of these titles. As User Interface design was my responsibility in development, I found myself going back to these moodboards a lot when thinking of layouts for menus and game screens.

Luckily, our client already had a large amount of digital artwork created for her branding. We were given all of these files to use and implement into the project. Our client also had a colour scheme that she followed, so we had guidance on colour choice from the very start of the design stage. The fact that she already had an existing IP meant that there was reference artwork/UI layouts to develop from (regarding the beat cards), meaning prototyping was quick.

I took a few aspects of all of the moodboards and combined them to come up with the final designs for menus and the gameplay screens. *[Provided in Appendix E]* These were ran by our client and her artist to approve colouring/Art asset usage to ensure that it matched her Intellectual Property.

These designs match her colour scheme, are friendly and inviting to children, and have the large, clear buttons that are obvious to children and help adults quickly navigate the application.

Implementation

As Technical lead, I chose that this product should be created in Unity. Mainly because all of our team had a large amount of experience working in Unity2D from the first stage of this course, but also the fact that it can easily export to iOS, Android and PC with the relevant SDKs.

Feature Implementation was majoratively completed by Jack Brewer, who was the most proficient programmer of the group. This included the main Beat Detection script, as well as Saving/Loading user created beats.

Jack started by creating a metronome that would play a tempo reliably, which the client insisted on hearing before we continued, due to it's vital part in the overall functionality.

Secondly was Beat Detection. First of all, pre-determined beat prefab objects are created by us and then loaded into the play screen and displayed. The Beat Detection works by taking Timestamps of user input, and then comparing that to timestamps of the beat that has been loaded into the play screen. This script can be paused for tutorial purposes, and also plays an audio clip of our client saying "Get Ready Play Now" before gameplay, something that she does in real life so we believed essential to copy into the app. This loaded beat can also be displayed as side-on animations of a foot stomping or ball bouncing, which is done also using prefabs of animation blocks that are loaded into a location.

The third major system is the ability to Create, Randomly generate and save beats. This works by reading/writing into a .txt file that is saved onto the users device, using a letter to refer to each type of beat and reading that to import the beats into the play screen.

The overall implementation of these features was relatively smooth and non-problematic. Due to the nature of this program there was very few expensive functions or animations to run on mobile, so the fact that I had thought about optimisation from an early stage meant it was a quick and simple process.

Testing

As the Technical Lead, I took on the majority of the testing of this project. This included both Weekly and Final Usability Testing that ensured our development was staying on track and we didn't get caught out by unexpected bugs.

Weekly Functionality Testing

The weekly testing took place the day before our team meetings each week. This meant that I was able to bring along any issues I had found and we could fix them together before signing off that sprint. During the sprint, I would fill in my [testing spreadsheet](#) [Example provided in Appendix F] with any functionality, design changes or bug fixes that were made during that week. Anything and everything that could be tested was given an entry so that I could ensure it all worked together at the end of the week. Each sprint had its own table, which showed the following information:

- Task Name
- Assigned Developer to that Task
- Task Type (Feature Implementation, Design Changes, Prototyping or Bug Fixes)
- Testing Method (How testing was carried out for each task)
- Expected Result
- Actual Result

- Resolved (Whether the task was fixed there and then or was added as a task for the next sprint)
- Notes
- Various Statistics (Task Distribution, Success Rate [Tasks that have Actual result of “As expected”], Testing left to do and Developer Distribution [Not any indication of actual hours put into the project, just for general reference when assigning tasks])

This weekly testing was done on an Android build on my mobile, so that I could find any errors with file saving, tap detection or other problems which arise on mobile but not PC builds. Every 2 weeks (approx.) I would build the app also onto Aden’s phone so that we could find any errors that came as a result of using a different device.

I had done a lot of formal testing during my A level Computer Science, so was confident in executing this process with a larger project. Doing this testing with a team also helped me realise how this same system would be used on larger projects, with bigger teams.

Usability Testing

Regarding our Usability Testing, we would be required to fill out extra forms that allowed us to run usability testing on people under 18 years old. This would apply to us, as we would ideally get a range of children to test the app while developing it so that it meets the target audience’s expectations when released.

Filling in these forms/getting them approved would add to our development time, and hence we went down an alternate route of User Testing. Caroline Stephenson contacted Hyde Park Junior School, where we had a meeting with senior management to pitch the session and setup the school’s tablets with our app.

The easiest way of getting feedback from children was to ask very simple closed ended questions. This is why we created a small questionnaire for kids to fill in after the session which would guide us as to whether or not they liked how the app looked and whether they understood it’s functionality.*[Provided in Appendix G]*

This was accompanied by Unity Analytics set up by Jack Brewer, which would live stream the results of each level played by any device connected to the internet. This information was then used to draw up results on playtime and score achieved so we could draw conclusions on a child’s gameplay. All of this was done completely anonymously.

The results from the Usability Testing support the fact that we completed our R&D (Research and Development) stage effectively. As shown in the graph *[Provided in Appendix H]*, the amount of players that achieved an amber score decreased over time, and the amount that scored green increased. This graph alone proves that over time the children that played the app gradually increased in skill.

We also received a lot of feedback from the teachers on the app’s overall look and functionality, and how it could be improved to meet the needs of the pupils. This has all been fed back to our client, and includes things such as increasing button size and using icons rather than images, to improve accessibility to younger children.

From the organisation of this usability testing I have learnt that a lot there are a lot more ethical and legal considerations to take into account when organising people testing your product, especially with younger kids. I also discovered Unity Analytics, which I didn’t even know existed, which I will definitely use during usability testing sessions in future projects.

Project Management

Overall project management was done by Aden Webb (Project Lead) on Trello. We very quickly adopted and understood the Scrumban workflow that adopts a 'swim lanes' style approach to project management but also allows developers to take tasks as and when they wish. Our Trello was massively helpful for me both while developing, and managing the repo, as it became home to our user stories, and all information on the current and planned sprint. The Backlog section also allowed us to dump ideas that we have which will later be evolved into larger tasks.

GitHub Repository

My responsibility was managing the Repository for this project. This involved creating branches for every developer, for each task they wished to complete. This was then merged into the Staging consistently throughout the sprint and then into master at the end of a sprint. This method meant that we always had a copy of the last sprint, current sprint, and branches were named as to what task they were achieving.

I had never done GitHub with teams before, so the concept of branches and pull requests was very new to me at the start of this module. The technique originally was to create branches from master as and when developers need them, so that each branch would create a pull request that could be attached to the Trello cards. (Completing the entire tasks lifecycle) *[Provided in Appendix I]* From discussion with other teams, and a hiccup with a lost commit, I decided to implement a staging branch which worked as a middle ground between us and master while a sprint develops. *[Branch management Diagram provided in Appendix J]*

This method also meant that I was able to create a build from staging at the end of each sprint, fully test it on my spreadsheet, and then merge it with master once I was happy everything was complete. Each client meeting takes place at the end of the branch management timeline, where I will take a build of master (in the occasion that Sprint 2 hasn't been fully tested), meaning the client didn't see a broken product.

For every task pull request I wrote a small section on the overall task, some technical notes, and an overall summary of what was completed. *[Example provided in Appendix K]* This helped Aden with writing the end of Sprint Reports, which were uploaded onto our Minisite. *[Example provided in Appendix L]* This was also where code review was undertaken by Aden Webb following a code standardisations document *[Provided in Appendix M]* that I wrote at the start of development.

Weekly Meetings

Each Sprint meeting followed the same weekly meeting checklist *[Provided in Appendix N]*, as to ensure we all fulfilled our roles at every meeting. At each meeting I also created a 'Release' on github, meaning we could track back to the commit where we completed each sprint, and what was completed at that point.

It became obvious very early on that we would create a lot of documentation/progress notes as time went on, and I thought necessary to bring them together. I made a minisite in order to hold our specification, progress, sprint and client meeting reports, and just any documentation that was created. The minisite is protected under a login screen, and hence continues to enforce the NDA that is held between us and the client.

Presentations

Our Interim Presentations helped with guiding our creation of the final presentation, and also allowed us to learn from other teams. I believe we had a very rigid process from the very start, so the main piece of feedback that we received was that we occasionally rushed over client introduction and went straight to our current progress/project management; which was sometimes confusing for people watching. We also decided to implement a staging branch thanks to being introduced to the idea from one of our peers.

We wrote blog posts for each Interim presentation with both a video and feedback, which was helpful when developing the next presentation.

We are going to guide our Final presentation with more focus on the usability testing, app functionality, client response and future development.

Conclusion

I believe that it is a fair assumption to judge the product's success based on the Client Feedback we received during development, and the usability testing feedback. During all client meetings, Caroline was impressed with our progress and only had a few small changes to make before we started the next phase. She was always very complimentary on the amount of work we did, and she loved being so involved in certain decisions and was very pleased with our interpretation of her Intellectual Property.

Our feedback from the usability testing shows that we were very successful in making an app that is suitable for children, from speaking directly to the pupils we were constantly complimented on the game itself. The teachers were hugely complimentary of our professionalism while dealing with the school and our ability to talk to the children.

In terms of our team, I believe we worked brilliantly well together considering we hadn't worked together before. We took our personality types and assigned roles based on those results, meaning we were all perfectly matched for the job we were doing. Aden's meticulous project management always ensured that the project never swayed away from what the client wanted, Jacks coding skills meant we had features completed quickly and my keen eye for Design meant that we worked together fluently to finish this project.

I have learnt a lot during this module. Especially repository management for teams, Testing for large team projects and presenting about our progress; no doubt skills which will follow me into the final stage.

WORD COUNT : 3484

Appendices

WEBSITE LOGIN

username : d.livingstone@plymouth.ac.uk

password : CGDplymouth

Appendix A : Benefits of Rhythmic Learning

1. Finding silence : Teaching children from early ages that silence is golden, and can bring peace.
2. Testing Hearing : There is a period in young children's development where hearing isn't tested for about 2-3 years, and these sessions help with finding problems early on.
3. Engaging Attention : The clear sound of a real metronome can catch the ears of an entire group of children.
4. Sustaining Attention : Entertaining sessions, visuals and props can keep an entire class engaged in memorising together.
5. Be Fit & Control Movements : Using movement to a rhythm can help with fitness and learning ball control.
6. Recall & Use of Numbers : Learning to a Rhythm builds strong and confident counting, as well as visual and spatial understanding of patterns.
7. Identify Word Sounds - Syllables : Learning letter sounds & phonics can be easier when learning to a rhythm. Understanding of syllables is hugely important and should understand before a child is 4 years old.
8. Identify & Create Rhymes : Working with rhyming syllables is a natural progression after the word sound games.

Appendix B : Phase Specification Breakdown

<https://jackdotgriffiths.wixsite.com/jacksinthebox/specification>

B.1 - Prototype Phase Breakdown

UI Design	Landing screen with Play Button
	Main Menu - Story Mode, Create Beat Mode, Settings and Shop Links
	Level Selection Map
	Beat Creation
	Play screen - Settings, pause, layout of UI
UI Functionality	Landing Screen with play button
	Main Menu (Navigation not Functionality) - Story Mode, Create beat mode
	Level selection map - 3 easy to access buttons.
	Play screen - pause
Features	Metronome
	Tap Input Detection
	Song Score Generation
	Example Beats for Testing
	Choose Display mode for implemented songs - Fitness, Bouncing ball, Music mode

Sounds	The sound of a basic beat
	The sound of a slide (held beat)
	Intro voice ("Get ready copy me")
Art	Beats - Basic Beat, Slide Beat, Rest

B.2 - Initial Project Phase Breakdown

Bug Fixes	Fix all known bugs from Prototype Phase build
UI Functionality	Landing Screen with Play Button
	Main Menu - Settings Cog, Shop Link
	Custom beat creation
Game Features	Creating and playing of user created beats.
	Saving of user created beats
	Random generation of beats that you can play and save
	Add more songs to Predetermined beats
	Haptic feedback for mobile devices
	Highlighting in time so the player knows when to tap
Sounds	Background Music
	Sounds for UI - Buttons/Transitions
Art	Final art should be drafted and in the project. - Song Selection screen background, Landing screen background, Train (Moving between songs) and UI Art

B.3 - MVP Phase Breakdown

Bug Fixes	Fix all known bugs from Initial Project Phase build
Working UI	Scrolling level select screen for story mode
Game Features	<p>Tutorial - For "level 1" create 3 part tutorial. Part one should be done in fitness mode and only show simple beats and rests. Part two should show off the bouncing ball mode and have more complicated notes such as long and short notes. Part three will use the music mode and test rests as well as long and short notes by potentially incorporating the random generation function.</p> <p>For "level 2" create another 3 part tutorial. Part one should show off divided notes (quavers) as jogging in fitness mode. Part two should incorporate long notes into the newly established jogging notes and should use the bouncing ball mode. Part three will use the music mode</p>

	and test “jogging” beats, rests as well as long and short notes by potentially incorporating the random generation function.
Additional features	Unity Analytics for anonymous data collection.
Art Final Pass	All placeholder UI elements replaced with final art

B.4 - MAP Phase Breakdown

Usability Testing	Project needs to be formally testing on a range of devices by a range of users. We will have done a lot of this during development but it needs to be formally written up and tested to ensure the app is fully functional.
Fix Existing Bugs	Taking results from the Usability Testing, we need to go through and fix the bugs that were found. Theoretically after this, the app should be fully functional as intended and ready to release.
Refine Animations/Art Implementation	IF spare time is available during this Phase, developers should go through all art/animations in the project and add ‘juiciness’. This will include refining bouncing animations, making page transitions smoother and maybe making small alterations to some art assets.
Trailer Made	To get ready for an app release, we should have a trailer uploaded to youtube displaying the apps functionality.

B.5 - Polishing Phase Breakdown

Usability Testing Response	Analyse the results from our usability testing sessions, draw up some conclusions and possible changes for the future.
Google Play Store Ready	We should research all documentation, testing and graphics that are required for an app store release and have that ready to go for the client to add to the Google Play Store.

Appendix C : Social,Legal & Ethical Report

https://docs.wixstatic.com/ugd/fb308b_f323c3bb8dae468ba3418bf5d66bc1eb.pdf

C.1 - Team Legal Agreements

As a team, we have put in place an internal Memorandum of Understanding (MOU). The MOU outlined all of the work times, team roles and what happens if one of us is ill or decides to leave the project. The MOU is purely internal as our client already has an agreement with the university as to the required meeting arrangements in order to get students to develop their prototype.

Our MOU states that we have all agreed to these terms :

1. To meet once a week for a chance to update all team members and make sure the project is still heading in the right direction. This will be an in-person meeting. During this meeting a code review should be undertaken.
2. To follow a set of code conventions for all code written. These conventions can be found in a separate document.

3. Commit notes and titles will have a standard that includes the sprint number and good information. The description should accurately detail what was changed and what files were touched. These will feed into the merge notes at every meeting.
4. To complete a minimum of 10 hours worth of work a week. A week consists of Monday to Friday and also will not include the Easter holiday.
5. Each member of the team should keep track of how much time they have worked.
6. The client liaison should contact the client at least once a week with updates on the project.
7. If a member of the team cannot complete their minimum points worth of work or cannot attend a weekly scrum their work will need to be picked up by other team members. h. Tests should be undertaken during each sprint to ensure all functionality works before the next week's tasks are selected.

These 7 points cover all module requirements, as well as some further requirements in order to develop a successful project.

C.2 - Client Legal Agreements

It was made clear at the start of this project that our client wanted a Non-Disclosure Agreement to be put in place in order to protect her Intellectual Property. A Non-Disclosure Agreement (NDA) is a legally binding contract that recognises that there is a confidentiality agreement between two parties, in our case, Jacks-in-the-box and Rhythmic Learning.

Our NDA confirms to our client that we are going to enforce a confidential form of development, where progress, goals and phase results are only shared with her and the University as required by our assignment specification. This agreement protects the Intellectual Property of Rhythmic Learning as well as the code that Jacks-in-the-Box creates.

We also made sure that our agreement covers our final result, as we need to provide it for our assignment hand-in, and as a portfolio piece. We have established the portfolio piece only needs to take the form of screenshots and client testimonials, and if the app gets published, a link to the relevant store page.

C.3 - User Testing Ethics

The main ethical considerations from this project come as a result of our target audience. At some point during the development we have to be able to do Usability Testing on our Target Audience, otherwise we are not going to be able to get reliable testing responses.

In our University, like every other establishment, we would be required to fill out extra forms that allow us to run usability testing on people under 18 years old. This would apply to us, as we would ideally get a range of children to test the app while developing it so that it meets the target audience's expectations when released.

Filling in these forms/getting them approved would add to our development time, and hence we are going down an alternate route of User Testing. Caroline Stephenson, our client, stated during our first meeting that she would be able to take the app to various sessions that she runs, as to gain some feedback on the project's functionality. This situation does however mean that we will have to adapt

our method of taking feedback on the app to accommodate our client asking the questions, and the children's responses to questions being useful to us.

The easiest way of getting a quick and reliable question from children is to ask mostly closed-ended questions (Questions with Yes/No Answers). This will be something that we have to look into when we get to the MVP/MAP stages of development.

C.4 - User Rating Guidelines

The current plan for the project is to not release for either iOS or Android App Store, as that does not meet the requirements of the module. We are however going to intentionally develop for the Android System, and use that as a basis for our testing as we go through, as our client wishes to release this on both App Stores in the future. Android SDK is also free to download, so hence is better for development and Initial testing. However, despite the fact we will not be releasing the app during the PRCO203 Module, It is still necessary to take relevant action to adapt our development to suit the age rating of the target audience. As our Target age rating is 5-10, we will be going for the most inclusive possible age rating, also known as PEGI 3 in Europe and ESRB E (Everyone) in the US.

The PEGI 3 Rating is what we will be developing for, due to the bulk of target audience being based in the UK:

PEGI 3


The content of apps with this rating is considered suitable for all age groups. Some violence in a comical context (typically cartoon-like – Bugs Bunny or Tom & Jerry – forms of violence) is acceptable. A child should not be able to associate the character on the screen with real-life characters; they should be distinctly fantasy. The app should not contain any sounds or pictures that are likely to scare young children. No bad language should be heard.

The PEGI 3 Rating is very achievable for our development, as our only representation of characters will not be shown in any kind of violent situations, the sounds will not be scary to young children and there will not be any bad language.

Our client, Rhythmic Learning, has had years of experience working with small children of this age group and will be providing all of the sound/art assets, and hence all content will be previewed/vetted by an experienced source.

Appendix D : Design Moodboards


Existing Similar Games



[Rhythmic Village](#)


This game is a good example on how to represent accurate musical notation while remaining child-friendly. It's UI layout is very intuitive and gameplay (Video Linked above) looks fun and engaging.

There are also lots of different game modes in this app that may serve as inspiration.



[Musilla Music School](#)

Musilla Music School is another brilliant example of a similar functioning app. The Game features levels that get increasingly harder, where each level displays a piano and some sheet music that the child must play to continue. The actual game mechanic isn't representative of what we're going for but this serves as a brilliant display of level progression and UI design.



[Rhythm Party: Kids Music Game](#)

This Game is less educational than the project we are wishing to make, however, our client said that they wanted some way of keeping track of the 'success' of the player's game play.

This demo features scores, accuracy measurements and a visual reward of the player being successful in the game play. This will be very appealing for children and will help with keeping them engaged with progressing further in the game.

GUI Design/Planning





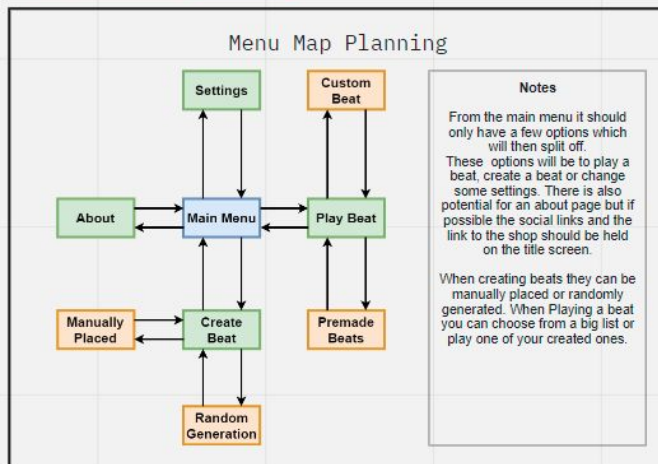


Notes



All the UI in the game should be large and obvious, for the landing screen nothing but an obvious play button is necessary

For the main menus large buttons that take up the majority of the screen will insure that children understand how to use them. For the progression of the game we were thing a star based reward system that congratulates good rhythm with higher points, the UI for this will be a path like screen with different levels and the score achieved

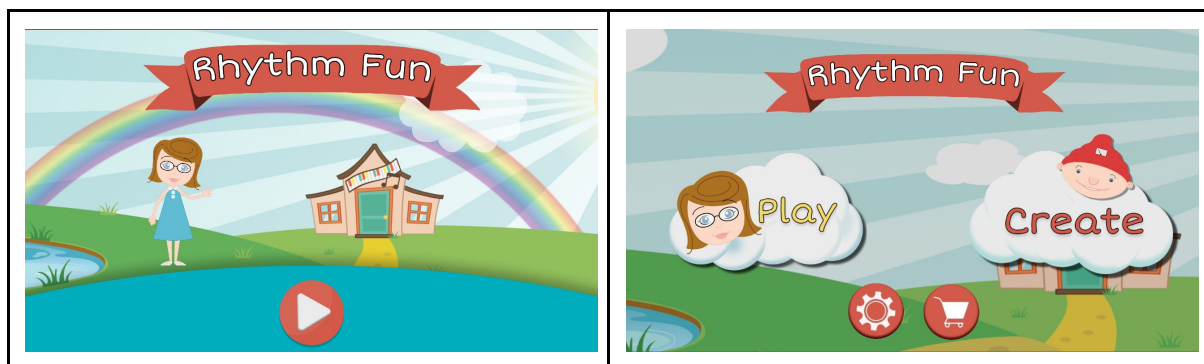
The game play will have large UI showing points and a pause button that is out of the way but easy to hit with little precision.

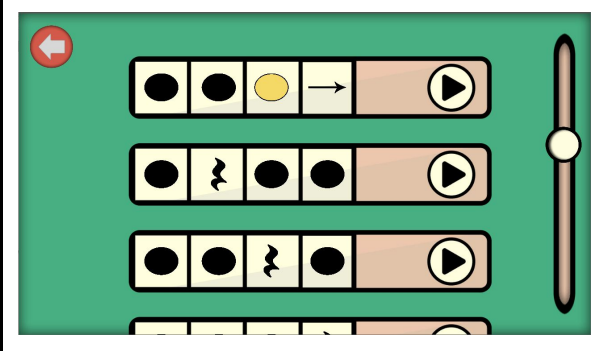
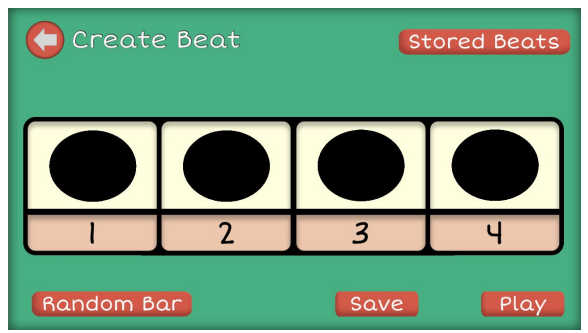
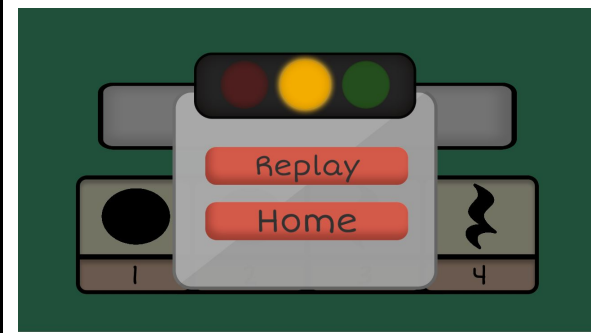
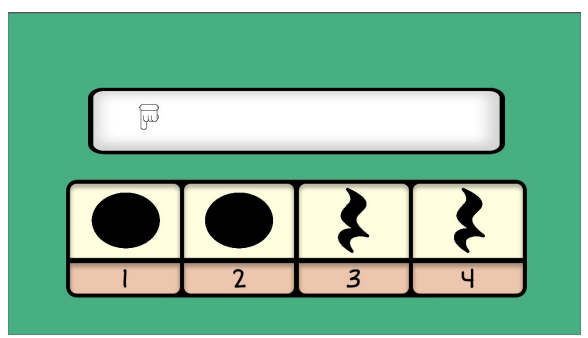
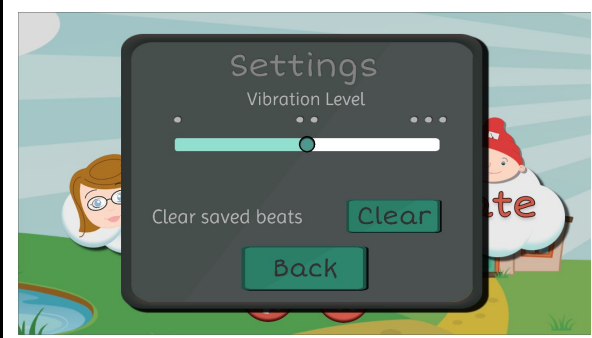
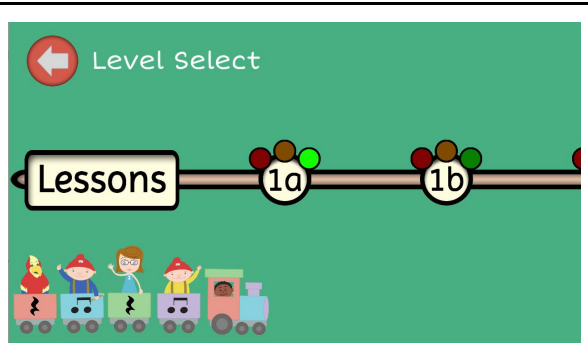
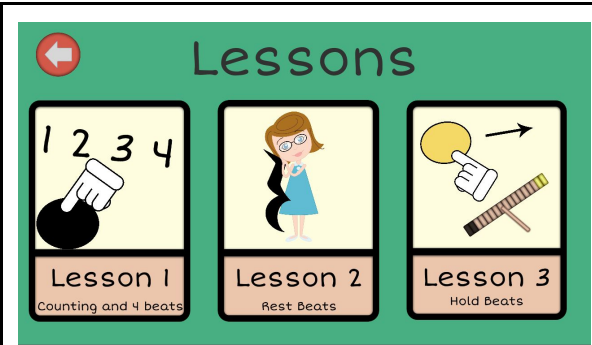


Color pallet

Appendix E : Final Design

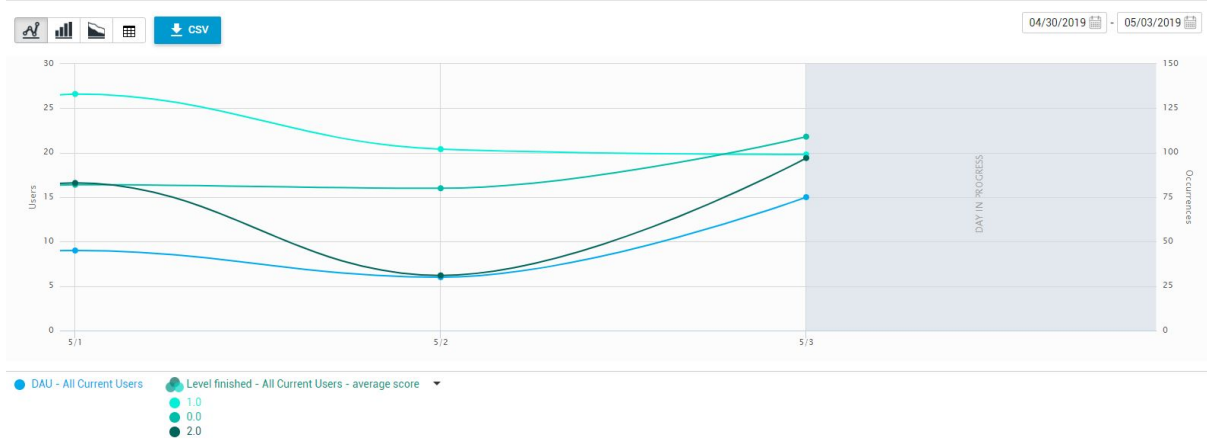




https://docs.google.com/spreadsheets/d/13iyud-1gv_b3QEj9VjvZorfG3XcSPtEXWfWzZPfrAMQ/edit?usp=sharing

Appendix H : Usability Testing Report

https://docs.wixstatic.com/ugd/fb308b_6de4bd6830de4fc1a77780a1104f6a9e.pdf



Appendix I : Task Lifecycle

Jacks-in-the-box

TASK LIFE CYCLE

Rhythmic Fun

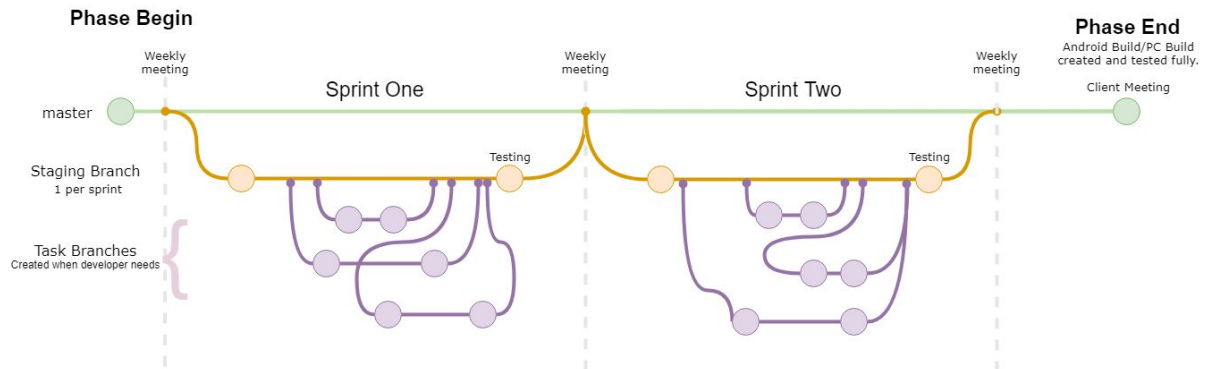


JACKS-IN-THE-BOX
JACK GRIFFITHS - TECHNICAL LEAD
JACK BREWER - CLIENT LIAISON
ADEN WEBB - PROJECT LEAD



Appendix J : Branch Management Diagram

Jack's-in-the-box Repo Branch Management



Appendix K : Repo Pull Request Notes

Aden/Griff - Refine beat animations #61

Edit

Merged JackDotGriffiths merged 4 commits into `Staging-Branch---MVP-Sprint-2` from `Aden/Griff---Refine-beat-animations` 5 days ago

Conversation 1 Commits 4 Checks 0 Files changed 28 +570 -58



JackDotGriffiths commented 5 days ago

Task

The animations that play when an alternate beat mode is selected need to be tightened up a little. This could be looked at in a number of different ways. The first is to try and make the transition between the different animations smoother. The main other thing is to ensure that the timing works nicely and the beats line up with bounces etc.

Notes

- Animations now enable when the beat hits, rather than being already enabled.
- Animations disable once they're complete.
- Animations now start 0.1s before the beat hits, so the boot actually hits when the metronome does.
- Animation positions moved closer together to improve overall flow of beat display

Beat Displays have been refined now so that they appear more appealing to the user of the app. The animations display only when the beat hits, and they have been edited to ensure they hit directly in time with the metronome.

+ @ ...

Reviewers

No reviews

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone


No milestone

Notifications

Unsubscribe

Appendix L : Minisite Blog Post

<https://jackdotgriffiths.wixsite.com/jacksinthebox/blog-1>

 Aden Webb · 5 days ago · 2 min read

Initial Project Phase - Sprint 5

The saved beats now function as desired. The users can play saved beats by using the play button next to each one. This will load them into a play screen that has been populated with the selected beat. When creating beats for saving, it is impossible for a user to create a duplicate as a check is performed to see whether or not the bar already exists. When listing the saved beats we have changed the order so that they are listed from most recent at the top to oldest at the bottom. We felt this would be more understandable to a new user.

Some elements of the code standardisation document had been missed out during coding - to make sure we were up to date on this we went through the current code base and made sure all major functions were commented clearly as per the document.

With our usability session planned, we needed a way to gather data from the testers easily. The best solution to this seemed to be unity analytics. This has

Appendix M : Code Standardisations Document

https://docs.wixstatic.com/ugd/fb308b_749c02b9beda4cf390bbfdde5da32ce6.pdf

PRCO203 Standardizations Document

Variable Names [All Team Members]

For Variable Name conventions, we are going to use Camel Case. This is the same as what we used in AINT254, so will be the easiest for all team members to follow.

Camel Case - words created by concatenating capitalized words except for the first word which is lower case. (e.g. A variable for a player controller will be called playerController)

Naming conventions will be following the typical format based upon a Variables access Level.

Private & Protected Variables -

Format : letter "m_" + the name of the variable.
Example : m_playerController

Public Variables -

Format : name of the variable.
Example : playerController

Local Variables -

Format : "_" + name of the variable
Example : _playerController

Static Variables -

Format : "s_" + name of the variable
Example : s_playerController

Each of our variable names will follow this format :
{VariableAccessLevelPrefix}{Unity/VariablePrefix}{Variable Name}

e.g (private AudioSource called Speaker) : m_asSpeaker

Please follow this table for name prefixes when naming Variables.

Unity Variable Prefix Examples (Ask Jack Griffiths if unsure)	
GameObject	go
AudioSource	as
AudioClip	ac

Rigidbody	rb
Collider	col

Method Names [All Team Members]

For Methods we are using Pascal Case. This is again the same convention as what we used in AINT254 so should be standard in all of our team members coding.

Pascal Case - words created by concatenating capitalized words.

Example:

```
void TestMethod(){  
    //Do Stuff  
}
```

Method Names should be descriptive of its function.

Commenting Convention [All Team Members]

Comments should be used to the judgement of the programmer, but need to be frequent enough to convey what the code is doing.

ALL methods should have the a summary generated by VS. This is generated by using '///' before the methods. If unsure on how to use this, ask Technical Lead.

Comments should clarify the intention of the following lines of code in less that 1 line of description.

Commit Names [All Team Members]

Commit Names will follow this typical convention in each branch.

Commit Name :
S{SprintNumber}C{CommitNumber} - (Brief Title of Commit)

Example : S4C10 - Character Controller Refined

Commit Description:

Commit Description should be brief but concludes the changes and fixes made in that commit.

Example : Jumping Mechanic in Character Controller refined to match gravity of new environments. New Sounds Created, Imported and Fully Implemented for walking and running.

The Titles are going to be helpful when merging with master, as they will create an easier to follow log of all of our developments.

These Descriptions will be used when writing the Larger Merge notes which I write each week, so it is essential that each of the Commit Notes in the branches are fully cohesive in order to have a well documented development.

Branch Management [All Members]

At the start of each week, you will be assigned 2 or 3 branches to work on your tasks during the sprint.

Merging to master should only be done by the technical lead.

If you require to merge your assigned branches then that can be done through the github desktop, and we went through this during the session. If you have any issues with merging then please contact Technical Lead.

If you need to create a 'testing' branch, ensure you are making it as a copy of the task you are testing. Ensure to merge this once testing is complete before carrying on.

If you are unsure on any merging/branch creating rules then please contact Technical Lead.

Branch Merging Routine [Technical Lead]

At Each Meeting, the Technical Lead will complete the following checklist :

- Check with Developers that their branches are fully up to date and commit notes are coherent, matching the commit notes standard.
- Merge each of the branches into the master branches, while ensuring there are no conflicts.

- Once all Branches are merged, test master branch is fully working and functional as expected
- Delete all branches except from master. Ensure all Repo Users also delete the branches locally as to not confuse later on down the line.
- Create new branches for each of the tasks created in Trello. This can only be done after Project Lead has written out the sprint and distributed tasks.

Appendix N : Weekly Meeting Checklist

https://docs.wixstatic.com/ugd/fb308b_41af8e8f49be4268817a5d1694fd34fd.pdf

PRCO203 Weekly Meeting Timeline

Before the Meeting

All

- Ensure all changes are committed to the Repo.
- Notify Aden (Project Lead) of any uncompleted Tasks that will require more work.

During the Meeting

Griff

- Merge Jack branch with the master and write appropriate comments on Changes made.
- Merge Aden branch with the master and write appropriate comments on Changes made.
- Merge Griff branch with the master and write appropriate comments on Changes made.

Aden

- Fill in Trello story with completed tasks from the completed sprint.
- Take Tasks from "Approved from Backlog" in Trello, Add checklist of Tasks and add completion time. Move completed cards into "Planned Tasks".
- Assign tasks to team members.

Jack

- Assist Aden in Developing cards into tasks that are assigned to team members.
- Start developing the 'Patch Notes' Document from Griff's Merge Notes in order to send to the client.

All

- Discuss Backlog tasks and adapt them to "Approved from Backlog"

Soon after the Meeting

Jack

- Complete and Send Patch Notes document to Client

Griff

- Ensure all team members have pulled from master before adding any more code to their branch.