Game Plan

Euanicorn:

The Magical Rainbow

Contents

1 One Page Game Design

* The Title
* The Genre
* The Target Audience
* The Platform
* The Big Idea
* The Summary
* The Unique Points
* The Mechanic
* The Competition

2 Game Designs

* The Idea
* The Motive
* The Implementation
* The Interaction

3 Hardware

* The Thought
* The Choice
* The Reasoning

4 Software

* The Chosen Ones
* The Explanation
* The Reasoning

5 Graphics

* The Thought
* The Reasoning

6 Audio

* The Thought
* The Ideas
* The Reasoning

7 Game Designs

* The Structure
* The Behaviour

8 Code Designs

* The Decisions
* The Explanation
* The Reasoning

9 Game Features

* The Choices
* The Explanation
* The Extras

10 Time Management

1. *Bibliography*
2. *Appendix*
3. *Credits*
4. *Feedback*

One Page Plan

Title Euanicorn: The Magical Rainbow

Genre Platform, 2D side-scrolling game.

Target Audience Children between the ages of 6-12

Platform(s)

Android Tablet. If this goes well I will hopefully expand to other Android devices, such as mobiles and possibly port it to iPad/iPhone. However I will probably focus on extending the game for one device first.

The Big Idea

The goal of this game is to help improve child interaction with touch screen devices. The game will be made up of gesture controls, displaying the possibilities to young children. There is a big demand for technology now and teaching youths how to use them is a good thought.

Game Summary

A 2D platform game aimed at children. A mythical creature, a Unicorn, has been tasked to travel to the end of a Rainbow. This Unicorn is named Euan and he must complete multiple tasks through each level to continue his quest to the end of the Rainbow where he will be showered with riches.

Unique Selling Points

* Childs game for tablet
* Fully gesture based
* Fun teaching tool – learn to use a touch screen device
* Upgrade potential
* Time filler

Play Mechanic

Children will use the touch screen to control the main character and complete tasks. Controls will be made from the gestures designed for touch screen devices.

Similar Products

There are many games that fall under the platform genre such as Super Mario Bros. and Robot Unicorn Attack. These are very popular games and have good reviews. Looking at the style of Mario I can see why it is popular throughout all age groups.

Game Design

Game Idea

Euanicorn: The Magical Rainbow is the adventure of a Mythical creature journeying to the end of a Rainbow. Euan, the Unicorn, must venture through the different continents in the world of Esperia to retrieve the magical fortune found at the end of the Rainbow. Esperia is having trouble now that they have lost their King. Euan must help troubled Esperians with tasks to complete levels and continue his quest. Euan enjoys helping the people of Esperia but what lies at the end of the Rainbow? Will it be Euans greatest dream?

Game Motive

The idea behind creating this game is mainly that I have several young cousins who I really enjoy spending time with and would love to give them something fun to play during their free time. They are slowly growing up and asking for touch screen devices so I thought that designing a game that would help them adjust to newer technology would be a great idea. A lot of schools have also started to introduce new teaching methods that involve technology. I plan to introduce a fun way to teach children how to use these devices.

Implementation

This is going to be a game for children so the best way to design a fun interactive game is to talk with children and watch how they interact with their surroundings. I have had a few chances to talk to my younger cousins and discuss my idea with them which has helped towards the designing of this game. Another way I decided to research game techniques was by looking at games already on the market that are also side-scrollers or child games.

* Super Mario
* Terraria
* Animal Crossing
* Rayman
* Lego Games
* Little Big Planet
* Pokemon (versions after Gameboy colour)

and a few more. Looking at these games I was able to tell the differences between adult and child games and what I should be aiming to achieve when making this game.

The game will have a small learning curve; however I will try my best to implement an understandable user guide system so children do not get confused by any of the game mechanics. When I visited a few toy stores I was able to see that there are already a lot more touch screen devices for kids. A lot did require a stylus which I am planning on not using although I may make that an option at a later point.

The Interaction

The controls for this game will mainly be gesture based, so far, gestures that have been assigned are as follows.

|  |  |
| --- | --- |
|  | Currently tap is set to select objects on the menu and in game. Double tap will break objects. |
|  | Swiping your finger up on the screen will make the character jump. Swiping down will make the character stomp the ground. |
|  | This will be for a special section of the game. Will be used for the final task on a level. |
|  | Grabbing objects in game. |
|  | This will be the character movement system. It can be interrupted to jump. |
|  | This will shuffle loose objects in game. |
|  | This will also be used for a special task at the end of a level. |
|  | This will be used to move items that are blocking the way if it is not related to a task. |

Hardware

Thoughts

I had thought of multiple options of hardware before making my final decision. My original plan was to develop for the Nintendo DS. I liked the idea of using the double screens so the child could use the touch screen and still see everything that was happening; however Nintendo DS also have console controls which would probably be favoured more over the touch screen. I had also considered using an iPad because they are popular. They are fairly expensive in comparison to other tablets. If broken they would be less likely to be replaced. I then moved to the idea of using an Android tablet.

Choice

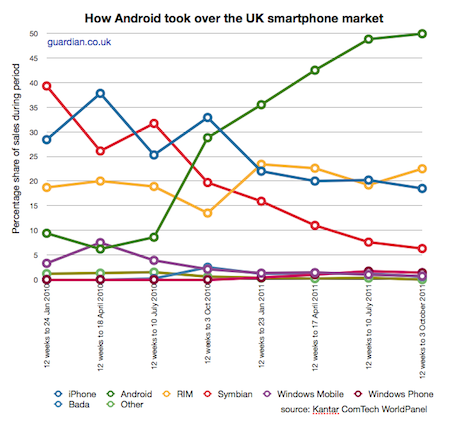
I decided upon using an Android tablet but the next question was what make of tablet should I be using. Luckily I had a friend who recently purchased an Android Tablet, the Google Nexus 7. After watching many reviews and playing about with the tablet I finally made my choice.[[1]](#footnote-1)

Explanation

The Google Nexus 7 was hyped up before its release and it lived up to the talk. It is a much smaller tablet in comparison to the iPad but considering that child hands are much smaller the seven inch screen was slightly more appropriate for my game. I also noticed how sturdy this tablet was. I did not personally test it myself however watching several reviews on YouTube[[2]](#footnote-2). Children can be pretty rough with toys they are given so this is a good feature for designing child games. Another bonus is the expense of the Nexus 7. The price for this tablet is very good and you get a user experience above what you paid for. A lot of the other applications designed for Android are free so there is more choice for games with less cost. Because Android has been used by a lot of companies there is a wide range of potential users. Android is one of the big operating systems now alongside iOS and soon to be Windows.

Bonuses:

* Small size, for small hands
* Sturdy, for accidents
* Cost effective

This graph is from a biased site however this is exactly how Android are sky rocketing in popularity[[3]](#footnote-3).

Software

Choices

The software I will be using to develop this game are:

* Android 4.2
* Android SDK
* Eclipse
* OpenGL
* Box2D
* JOGL

Explained and My Reasoning

Firstly, I will be using the latest version of Android, 4.2. There are a lot of tablets and smartphones with older versions however the tablet I am using has been updated. Using the latest version will allow me to code to the fullest capability at this time.

Android uses mainly Java and C++ for development. I have luckily used both of these previously but not to the extent I would have liked. Android is known for being quite tricky to understand right away so having this background experience with these languages should give me a head start as oppose to learning a brand new programming language. I have been eager to extend my knowledge of these languages after beginning to develop an Android application for another module. The app for that module simply calls on information from the internet and displays it as the client has asked. Creating a game will be a bigger challenge but having experience with this other modules app should allow me to pull thoughts from one to the other and overcome problems with a bit more ease.

The main software components I will need are kept all together in the Android SDK. This is a Software Development Kit designed to hold up to date libraries, debuggers, sample code and tutorials. This holds everything a developer needs and an added bonus is that previous versions are still included in the SDK so I do not necessarily have to develop for a new version of Android.

In my other module we are using an IDE, Integrated Development Environment. The IDE we are using is Eclipse. I have decided to use this mainly due to using it in the other module and that I have experience using it. It is a very popular IDE to use for Android development. The IDE helps me to write more efficient code as it is Object Oriented, like most Java development software.

OpenGL is an API, Application Programming Interface which renders 2D and 3D graphics. My game is a 2D side-scroller so this will be handy for making my images appears appropriately. I plan to use Eclipse and OpenGL together to create good code structure code with accelerated graphics. The extra piece of software I will need to include is JOGL, Java bindings for OpenGL. This software will allow me to combine code from each Eclipse and OpenGL so that they will run alongside each other.

Box2D is another piece of software I will be using however it is going to be used for one of my features. It is a physics engine to control the simulation of the game, how characters move, gravity etc. It keeps the realistic elements in a game to avoid it becoming too unbelievable. I will also be using it for collision detection in certain areas of my game.

Graphics and Audio

**Graphics[[4]](#footnote-4)**

For the style of my game I decided to begin by looking at games that are suitable for children and side scrollers such as Animal Crossing, Cooking with Mama, Super Mario etc. it's quite easy to notice that there are differences between games directed at children and those for adults.

Young children enjoy games and toys that are big and bright. Also when children are playing games they should be learning at the same time. By taking these thoughts I have been able to collaborate with a design student to make appropriate graphics for children.

The style I went for is quite simple without much detail. The images are vector based and coloured well for children to enjoy. Also adding elements of fantasy a child should enjoy it more as they have fairly a creative imagination.

I have asked my cousins what they think of the artwork and had a response that they quite liked the style.

**Audio**

A lot of the games I have researched tend to have fairly repetitive music that runs in the background. When the main character changes to a different are then it may differ slightly. When playing racing games the music is usually fast paced to give the player the feeling of speed. My game is fairly casual and there is no time limit on levels so I am planning to use a normal paced music tune. The track will only be music no lyrics. The music will be optional to the user. So children will be able to play the game while parents focus on work without distracting music.

I plan to add sound effects to the game also. These will cover:

* Footsteps
* Background
* Tasks
* Achievements
* Extras

There are multiple websites that offer free sound effects. I will be using multiple sites to find the perfect sounds that will match my game.

Game Designs

Structural Diagram

Behavioural Diagram

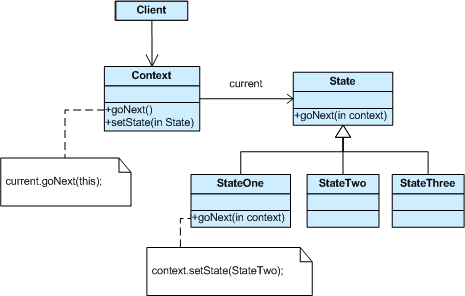
Game Patterns

The design patterns that will be used in the coding of this game will include the MVC, Model View Controller, the State pattern and the Observer pattern.

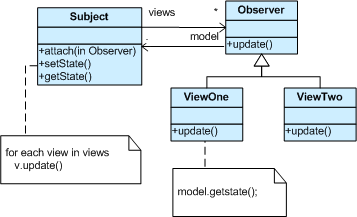
The MVC pattern has been assigned to us and must be used however I find that this is the best way to layout my main code structure and avoid getting messy code. The MVC pattern is perfect for game development. The user can see what is happening on the screen whilst it is changing however each section has been separated to avoid concurrency issues. The user only accesses the controller to tell the code what to do. The events are recognised in the controller and sent to the model to be executed. When the tasks have been executed the view is then updated to show the user what has happened.

My game will need this because it will need to update the image constantly and this is the tidiest way of separating the users’ actions from the codes actions.

My first choice is the State pattern. This is pattern allows objects to change how it behaves when the state has been changed. Objects can have multiple states and this pattern can switch through all of them informing the main class to update as the states change.



This pattern is perfect for my 2D side scrolling game because there are several things on screen that will need to change constantly depending on the users input. By having this pattern as soon as the states change the controller will be notified and any updates will be made to the view screen.

My second choice is the Observer pattern which is very closely linked to my choice of the State pattern. The observer is much like the State however where the state only updates itself the observer informs all classes that are important to that hierarchy. So all classes related to each other are then updated automatically and sent to the view.

I will have multiple objects on screen that will need to interact with each other so linking them via the observer pattern will help the game remain concurrent. An example: On one of my levels there is the task to break ice blocks. When tapping to break the block there will be several states changing at the same time by linking one state changes to the others then I can make sure they are concurrent.

Game Features

The main game features are Real-time gameplay and the use of a Physics Engine. These are the features I will be spending most of my time working on as they play a big role in game.

Real-time games are pretty easy to explain and understand. Real-time is used a lot today throughout Computing and general day to day practices, for example Clocks and Televisions. Real-time gameplay is the same however is used for generating images. Instead of the user selecting an option and waiting for it to happen, real-time allows the player to take control of the game and make events happen consistently while moving keys. Such as in PC games using the WASD keys allows you to move left, right, forward and backward as soon as you touch the key. Real-time, I feel, encourages the player to interact more with the game that they are playing.

Examples:

* + Age of Empires
  + StarCraft

My game is completely real-time. When the user plays the game, their gestures will determine what the main character in the game will do without any pause.

Euan needs to make his way through each level and can only achieve this via the users’ interactions.

My second feature is that I will be using a physics engine. The physics engine is very important for most games that have real-time elements. It allows games to simulate physical systems and create a feeling of reality within the game. It controls things like the gravity when jumping and falling, knocking in to obstacles and so much more.

Box2D is the engine I will be using. I am using it to manage the way my main characters sprite will move. It will be jumping; walking, running etc. and I want to make sure that it looks relatively realistic in game. I will also have obstacles in my game that will prevent the main character from moving forward. The physics engine will also be used for collision detection.

The collision detection is not necessary but to ensure the children benefit from playing I am adding obstacles so they do not miss tasks that must be completed. Another feature I will add if there is time is some AI (Artificial Intelligence). I think when children play the game they would enjoy seeing some NPC(Non Player Characters) moving of their own free will and let them interact with each other.

Time Management

Bibliography

**Game Design**

* www.evolus.vn/Pencil/Downloads-Stencils.html
* favbulous.com/post/695/6-multi-touch-gesture-icon-sets

**Design Patterns**

* en.wikipedia.org/wiki/File:MVC-Process.png
* sourcemaking.com/design\_patterns/state
* sourcemaking.com/design\_patterns/observer
* en.wikipedia.org/wiki/Model–view–controller

**Graphics and Audio**

* www.freesfx.co.uk/sfx/
* creativecommons.org/

**Reading**

* Head First Design Patterns – Eric Freeman, Elisabeth Freeman, Kathy Sierra, Bert Bates.
* Level up - http://site.ebrary.com/lib/dundee/docDetail.action?docID=10469788
* The Computer Game Design Course: Principles, Practices and Techniques for the Aspiring Game Designer – Jim Thompson

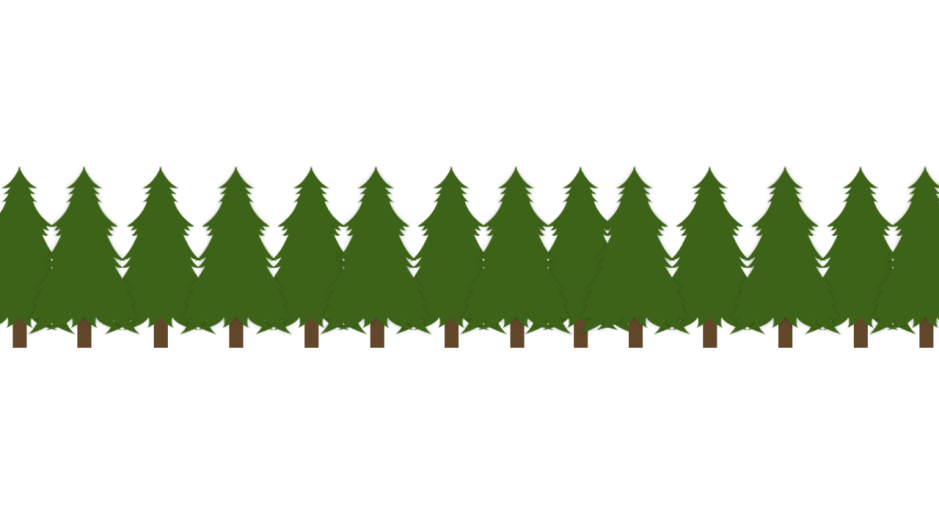
Appendix

Appendix A

These are samples of what the game will contain.

Backgrounds:

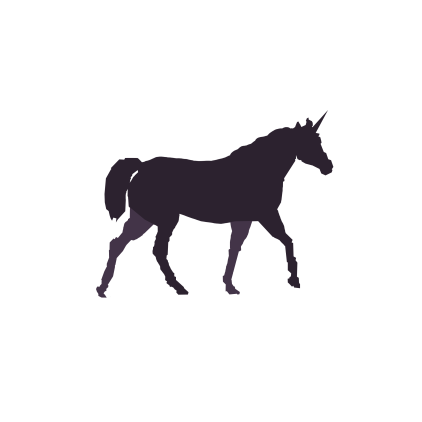
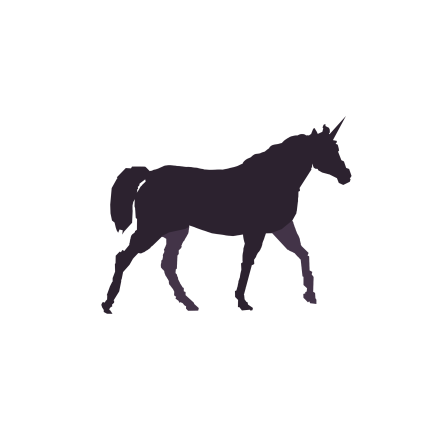
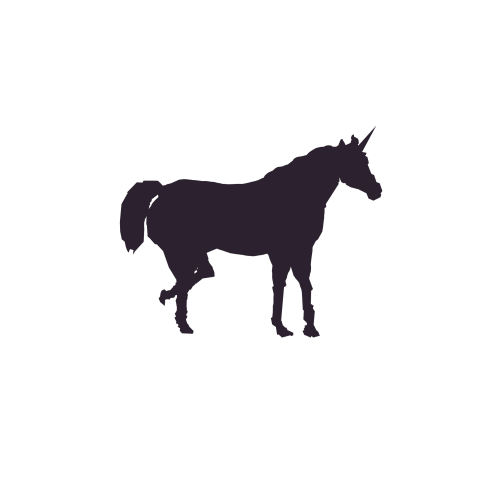
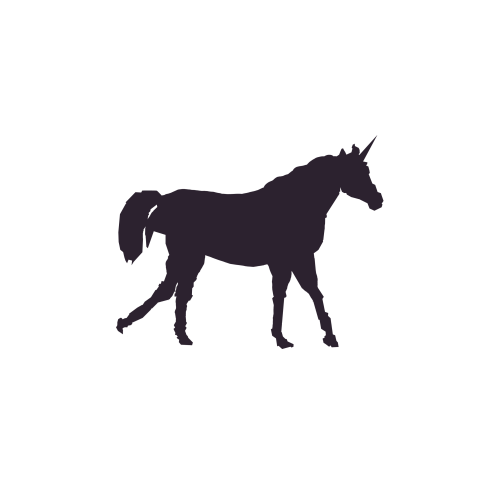
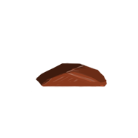
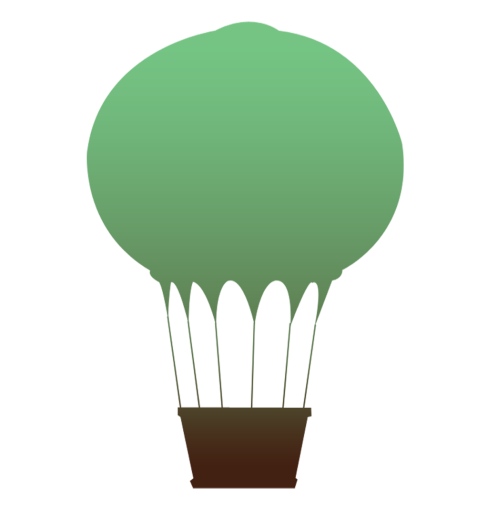
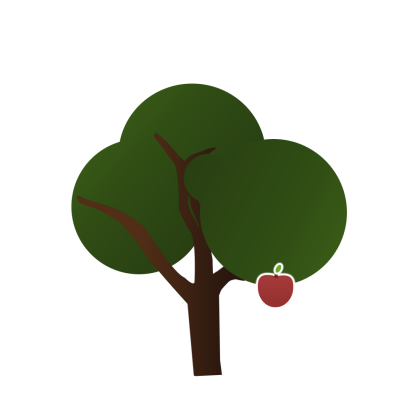
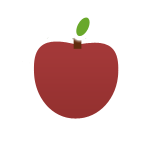
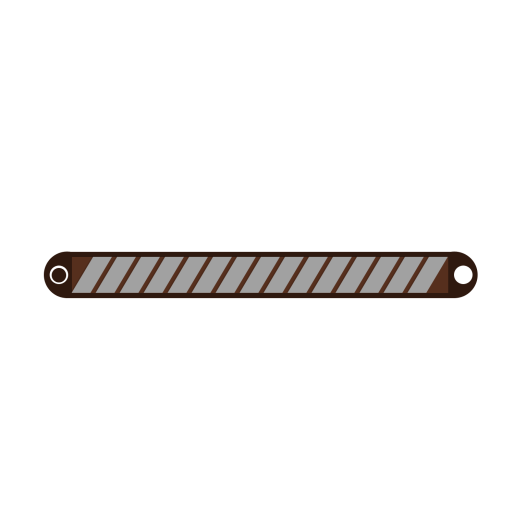


Extra Background:





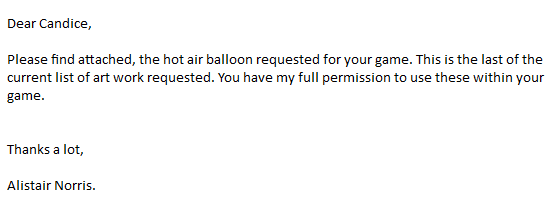
Sprites/In game :



Credits and Feedback

Credits

Alistair Norris – 3rd year Digital Interaction Design Student for doing a majority of my artwork and will be continuing to work with me until the game is complete. He has given me permission to use his artwork via e-mail.



Feedback

I struggled quite a lot to write this report. I understand what I am doing in regards to the program however was unsure how to word it in this plan. I would appreciate feedback for the entire report if possible however if not, what could I do to improve my UML diagrams also is there a rough plan on completing each section?

1. mashable.com/2012/06/27/nexus-7-google-table/ [↑](#footnote-ref-1)
2. https://www.youtube.com/watch?v=nKaPmrXKfO0 [↑](#footnote-ref-2)
3. http://www.androidauthority.com/why-android-could-reach-70-market-share-in-2012-29926/ [↑](#footnote-ref-3)
4. Appendix A [↑](#footnote-ref-4)