<u>Analysis</u>

"aa" is a strategy-based game application for both android and iOS users. The game was published by General Adaptive, last updated in December 2017. It has over 50,000,000 installs on android whilst ranking 38th on the apple store.

General Adaptive are known to create minimalist apps. They describe this game as "*the "hello world" app for Android Phones & Tablets*", "*i.e. aa is like snake on an old brick phone, it's fundamental.*"

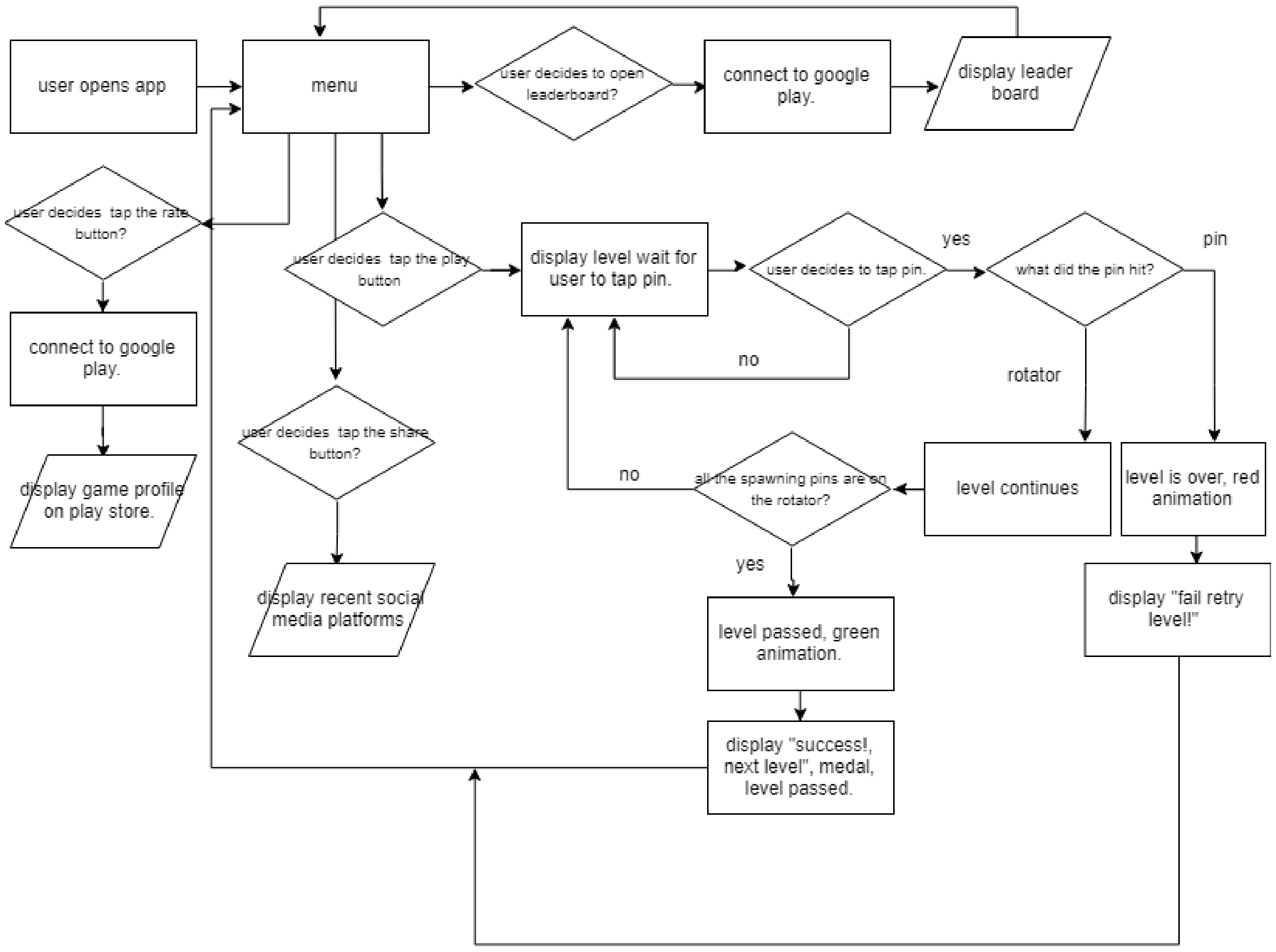
The main mechanic of the game is tapping so that the spawning pins must hit the rotating circle in the middle without hitting other pins in the process. The player proceeds to the next level automatically with a green animation, else if the player hits another pin a menu with a retry button appears after a red animation.

During the gameplay the player has a number of pins which its number is shown on the spawning point. This number decreases when the pins are launched. Each level is different some examples are; increasing the rotator's speed when a pin is launched, changing rotating direction each time that a pin is launched, having pins already with the rotator and also having different rotation speed. The music gives the player a sense of accomplishment whilst playing as it starts slow whilst gradually increasing the sound of a grand piano, conveying feelings of power and success.

When the player wins the level the pins and rotator sort of expands giving the impression of joy and accomplishment. The greens colour is also an indication of proceeding similar to traffic lights. There is no victory music only two beeps. It also displays "**SUCCESS!** NEXT LEVEL…" in green, showing a medal and the level next to it. There are also the leader board, play, share and rate buttons.

However, when the player loses, the pins and rotator shrink in conveying that the player really messed up and move downwards. The red colour is also an indication that something went wrong. The sound effect for this is as if all the pins fell down. After this animation it displays "**FAIL!** RETRY LEVEL…" in red with the buttons as previously mentioned before.

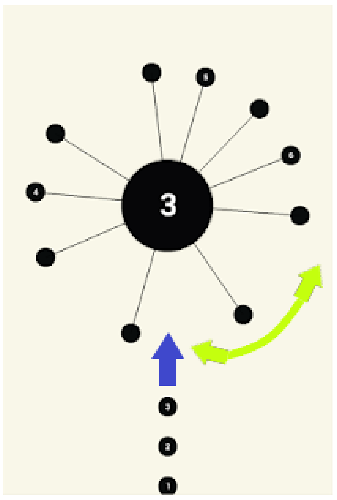
Below is an example of how I think the flowchart of this game is:


user opens app
menu
user decides to open leaderboard?
connect to google play.
display leader board
user decides tap the rate button?
connect to google play.
display game profile on play store.
user decides tap the play button
user decides tap the share button?
display recent social media platforms
display level wait for user to tap pin.
user decides to tap pin.
yes
no
what did the pin hit?
pin
rotator
level continues
level is over, red animation
display "fail retry level!"
all the spawning pins are on the rotator?
no
yes
level passed, green animation.
display "success!, next level", medal, level passed.

Design Documentation

Target Device: Mobile and Tablet devices as it requires the player to tap.

Game Mechanics: The main mechanics of the game are that;



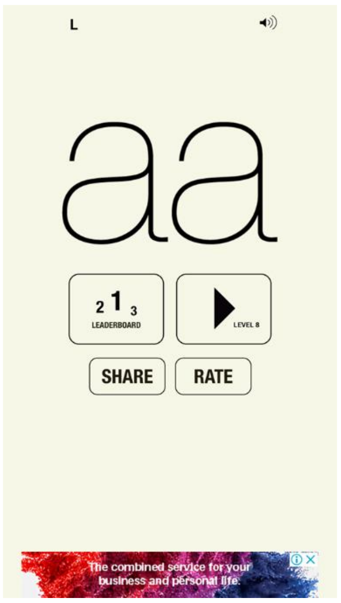
- The rotator rotates depending on the level. (Green arrow as shown in the picture.)
- When the player taps the front spawning dot goes upward towards the rotator. (Blue arrow as shown in the picture)

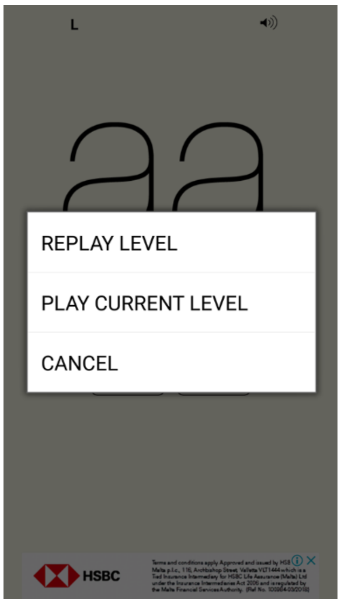
Objective: The main objective of the game is for the player to tap and attach all the pins to the rotator without hitting them against each other.

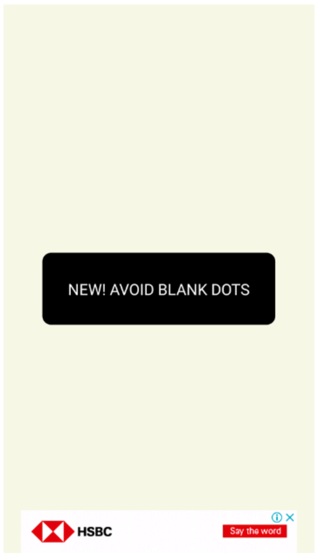
Game Elements: Circle middle rotator, spawning point which transforms into a pin when tapped. The level number is shown in the middle and the number of pins in each level are shown on the circle of each pin.

UI Elements:

- Start Screen/Menu; Shown in the first picture below. Simple logo with various buttons and a grey background. Two big main buttons. leader board, play whilst having share and rate in smaller buttons. Everything is in the centre except for the sound icon which then has a cross if the player chooses to mute it, located at the top right. Whilst on the left there is a small L. This pops up a small rectangle in the centre with the options to replay any level (must be passed before), play the current level or cancel. This is shown in the second picture below. When this pops up the background is faded to black however, having a light opacity. When the player presses any buttons the game just goes straight to the command.

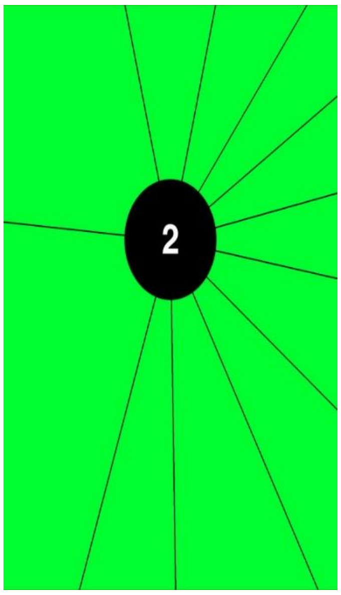


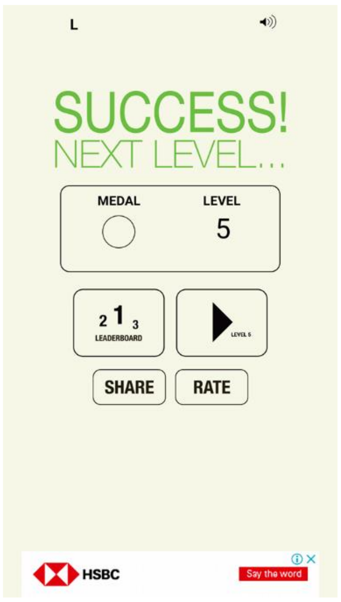




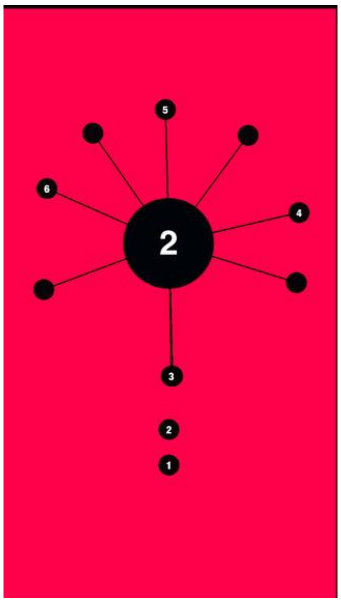
- Tutorial; The first few levels are basic tutorials so that the player can familiarize themselves to the game. The basic commands are shown like the picture on the right. The rotator and pins are not shown in the background whilst the command is shown in a black, centred, rounded rectangle with an uppercase, white text. After a few seconds the level starts.

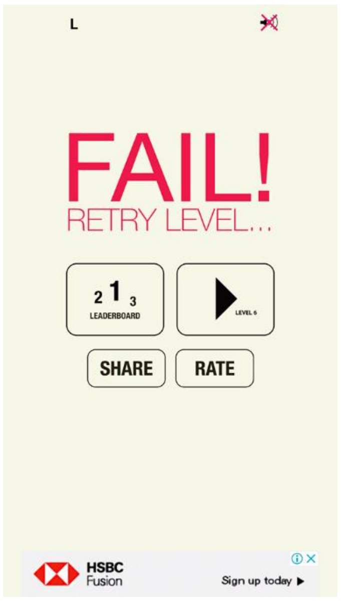
- Winning; When the player puts the last pin on the rotator the whole background of the screen turns green and the pins lines extend to fit the whole screen as shown in the first picture below. This animation then ends and a screen showing that you won the level in green, with a medal shows up, shown like the second picture below. This includes the buttons which are seen in the starting menu.



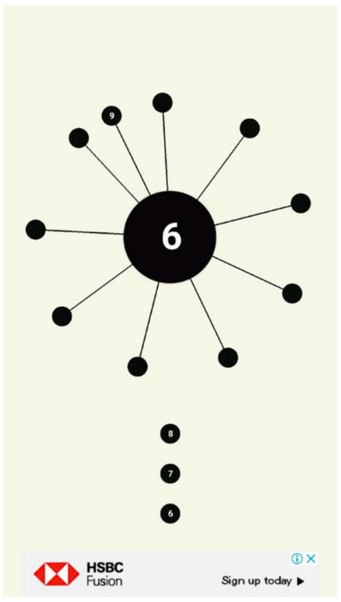


Failing: When the player hits another pin the background screen turns red whilst the pins and rotator shrink with the pins and their lines head to a downward direction, giving the impression that they fell. This animation is then followed with a screen showing that the player failed in red and to retry with the usual menu buttons.





Gameplay: When the player I playing the rotator appears central at the beginning of the level, this can either be empty or with pins already attached to it, depending on the level. The level number is written in the centre of it. A bit down there are the spawning points which have numbers on them showing how many the player must launch to win the game. The maximum amount of points shown is always three. When the pins are launched the number stays on the circle of the pin. This is shown in the picture on the right.



Evaluation

Making the basic game mechanic wasn’t difficult as I followed a tutorial on YouTube which did this basic mechanic making the game into a high score-based game; how many pins can you fit on the rotator.

The problems which I encountered are more related to the game flow meaning how a level progresses to the next and how to separate levels.

I wanted to make 3 levels. One level was that every time that a pin is thrown the rotator speeds up by 10f.
For this I did col.GetComponent<Rotator>().speed += 10f;

Another level was that when I throw a pin the rotation goes the other way. To get that I did col.GetComponent<Rotator>().speed *= -1

Another level was that I had some already fixed pins on the rotator and I had to fit in another 15 pins.

A problem which I encountered doing this was that in the spawner and NumberOfPins I did that I wanted to spawn another 15 but when playing the number of pins was 12. This was because Unity was also counting the three pins which I had with the rotator.