|  |
| --- |
| ESP 2015 |
| Flying Man APP |
| Android Mobile APP Report |
| Pierre Grant  Wasee Sameen  Ryan Brasington  6/29/2015 |
|  |
|  |



|  |
| --- |
|  |

**Purpose:**

The purpose of the app was to create a fun game for all age groups.

**Background:**

The name Flying Man originally came from our first idea which was about game centered on the original game Flappy Bird. But instead of a bird our character would be a pixie man named Joe. Joe was framed and sent to prison and he could not take it anymore, so Joe escaped. The point of the game was to get passed the pipes which were prison bars. What a great idea that was until we started the coding. Our coder had errors when we tried to run the program and he asked the TA, but all the TA suggested was come up with another project since we did not have much time. So from this we moved on to our new game and we kept the name Flying Man in remembrance of our original idea. We used a code we found and we referenced it in the References section. We thank Mark Nicolosi for his help. So what we did was create an intro page and change the level. Even though our code was correct for the intro page it did not show up when we ran the app. We changed the level because when we first ran the app we could not find a way to make the player jump because the coding was for a keypad which the tablet did not have, so we changed the level so that the player did not need to jump. We changed the character to our pixie man and that is our app.

**Abstract:**

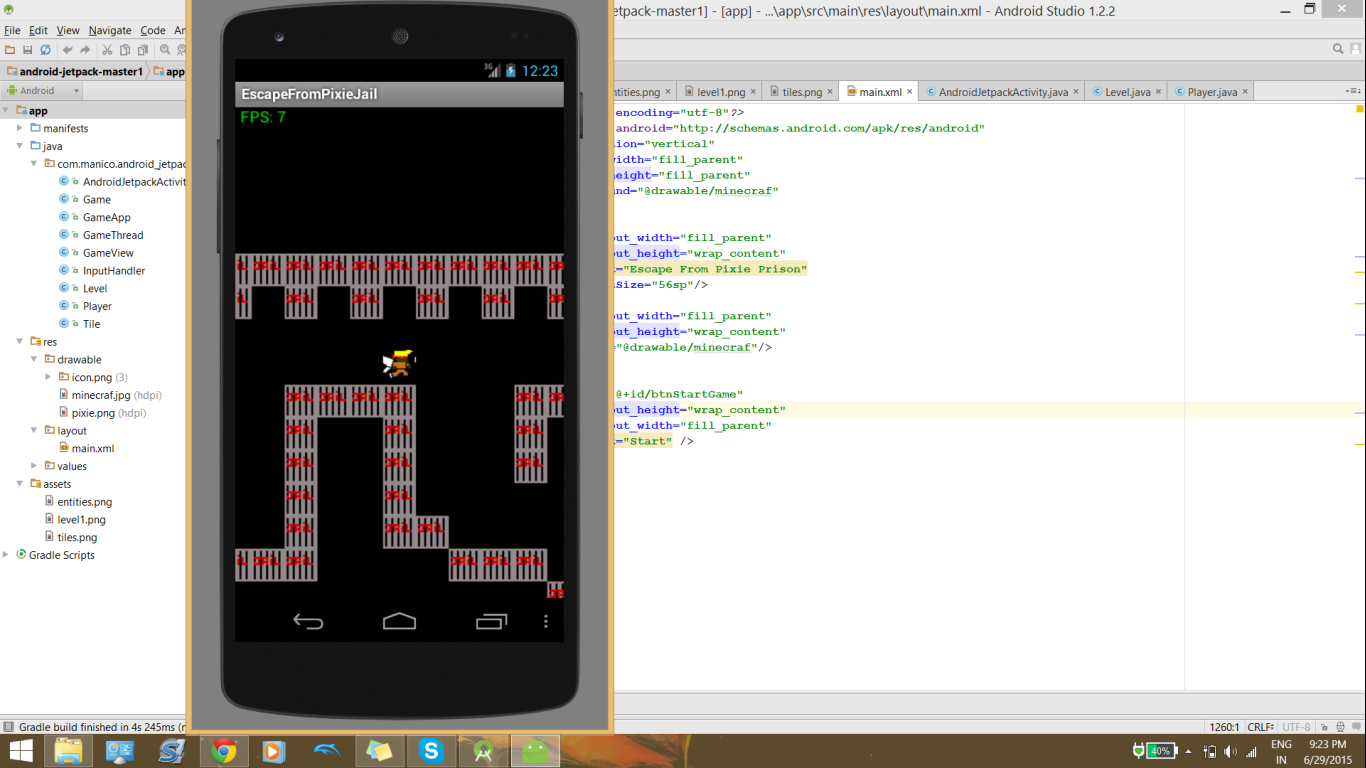
The purpose of the app was to create a fun game that is intended for all audiences. The app was created through Android Studio. We had the code for the accelerometer which would change the background when the positon of the tablet is changed, but we could not integrate it into Java because of time. We did use a code from github and we did make changes in the layout and we changed the overall game because we changed the level. Our graphics partner changed the level by creating another level through Photoshop and changed some of the coding so that when we ran the app the level would change. We changed the level because the original code only allowed the player to move left and right, but the only way to up is to use a keypad or board, but our tablet didn’t have that and we have had to change most of the code because the original creator used the keyboard command. The game was created so that the player won’t need to jump but instead just move left and right. We made the game into a maze so that when the character gets stuck he would technically lose because he was trapped. Due to time we were not able to put a game over screen. Our app was able to launch and actually worked. The app was should run on most tablets with an sdk of 18 or greater.

**Methods:**

Coming up with a game at first was to us not that hard of an idea until we released how important coding was. Our graphics guy knew how to create some great images, but our coder didn’t know much and I was the processor, so I had to help him. I watched over 7 videos on how to create a 2d game. The video helped me understand the coding part better. Then we had to come up with an idea and the idea we had first chosen centered about the game flappy bird. But we encountered an error, so we had to abandon the idea and quickly come up with another one. Pierre did some research and found a code off of github that worked back he did not understand the controls because the character could not jump and after some inspection of the code we realized that the game needed a keypad to jump. To deal with this we changed the makeup of the level so that no jumping was involved and we were successful. Then after we were done with that we tried to change the layout. The code for the layout was correct, but it did not show up when we ran the code. We still had to add our processing code (accelerometer), but we were unsuccessful due to time. If we had the processing part working then the background would have changed if the tablet was in a different positon. Even though we had some errors we were still able to run the app and it is enjoyable.

**Results:**

In the end we were able to make the app the run and below is a screenshoot of the app running on an emulator. When the app finally ran my teammates and I were happy because at least had something and it ran. I looked at several tutorials to understand the code and Pierre did some great graphics. Ryan did not do much except help film the video. So I had to do most of the coding. We basically worked our butts off the day of the presentation, but right after we got everything to run we had to create a video, so we were able to incorporate processing, but we did have the processing code. If we were able to incorporate processing then it would have made the game better because our sensor was an accelerometer which would change the background if the tablet was in a different position. Sometimes the app might crash we tried to fix it but we could not. In the end, some of the errors did not stop the app from running and we had what we wanted an app that worked.



**Discussion:**

If we had more time I think we would be able to fix some of these errors. Also, if we had

more time we would have tried to create more levels. What I thought at first was if I used the code from the levels class and created another class called level2 than it would create another page with another level, but I was sadly wrong. One thing that I did realize was that if any has android studio they can use our code and create a level of their own which makes the app more user friendly. Even though with all these issues we were still able to make a basic game that ran and the app incorporated our ideas like the character and the level. We enjoyed working on this app together because it showed how difficult it is to create apps and we won’t take any apps for granted. We learned a lot from the course and will continue to cherish what we have learned.

**Conclusion:**

We all started from not knowing anything about creating apps to a basic understanding. From what we learned there are three components: Java, Processing, and Graphics. We were not that disappointed with the errors because we knew it was not going to be that easy. There is much room for improvement in our app. We could make multiple game apps out of it. If we had more time there would be more levels an intro screen, a game over screen, and our sensor. We intend to build on our app. Our app might not sound like much but it is really fun.

**References:**

These are the websites for one who would try to replicate our app.

Where we got the code:

<https://github.com/manicolosi/android-jetpack>

We thank Mark for allowing us to use it.

These are the tutorial we used:

<https://www.youtube.com/watch?v=rJcm5Oyi3YA>

There are more of these videos, but this is the first one in the series.

Our Github

<https://github.com/ESP2015/ESP2>