**Pizza Bounce!**

****

**Student: Amerie Lommen**

**Date: May 29, 2015**

**Introduction:**

Android applications can be used to get things done and increase creativity, but many users use apps simply to entertain themselves. Maybe they are waiting at a bus stop or cooking food in the microwave and need to pass the time. An app on their phone is perfect for this purpose. Therefore, this project is an interactive app that people can play with for a short amount of time, based on <http://cat-bounce.com/>.

Animation in Android apps is done using custom view classes and threads that redraw objects on this canvas periodically (e.g. 50 times a second). Stanford University has released code[1] for the foundation of these classes and threads, which provides the mechanisms for creating a single simple object with a random, unchanging velocity that will bounce off the sides of the screen. In addition to this, the project implements the following features:

1. The user can create new objects by tapping the screen. (There are at most 10 objects on the screen at a time).
2. When an object is created, it falls from the initial position where the user tapped and bounces off the bottom of the screen.
3. Falling objects accelerate as they fall and decelerate as they rise to mimic gravity.
4. Users can change the speed and direction of existing objects by swiping them.
5. The objects are circular images of pizzas.
6. The objects rotate at an initial angular velocity, which can be changed by swiping the object.
7. There is a “clear” button to remove all objects from the screen.
8. There is a “Help” button to show instructions to the user in a separate activity. (Note: this feature was not included in Version 1 of the app due to time constraints, but will be included in the final version.) **Added since version 1.**

This project has a simple user interface, but programming the bouncing objects was challenging. It was mostly based on the bouncing ball program from Stanford, but also implements rotation (using matrices[2]) and the ability to change velocity from user input (via a Velocity Tracker object[3]). Another challenging aspect was saving the state of the Canvas so that the existing objects remain when the user rotates the screen. This involved making the “Sprite” (pizza object) class Parcelable by implementing several functions.

Since version one, the app has been heavily debugged, and now the pizzas do not behave strangely when they are tapped and held in place, nor does the app crash sporadically when the user creates a new pizza. Additionally, according to user suggestions, several user experience improvements have been made: the target area for catching existing pizzas is now slightly larger, and the pizzas are limited to a lower maximum speed.

**References:**

[1] Stepp, M. 2D Graphics, Animation, and Games [PDF document]. Retrieved from Lecture Notes Online Web site: http://web.stanford.edu/class/cs193a/lectures.shtml

[2] Sechan, G. (2013, April 12). Android: Using matrix to draw bitmaps. Retrieved May 19, 2015, from http://stackoverflow.com/questions/15978721/android-using-matrix-to-draw-bitmaps

[3] Tracking Movement. (n.d.). Retrieved May 19, 2015, from <https://developer.android.com/training/gestures/movement.html>

**User Interface Design Screenshots:**

|  |  |
| --- | --- |
| **C:\Users\Amerie\Desktop\lommen_final_project\pizzabounce_blank.png**  **C:\Users\Amerie\Desktop\lommen_final_project\pizzabounce_several.png** | **C:\Users\Amerie\Desktop\lommen_final_project\pizzabounce_single.png**  **C:\Users\Amerie\Desktop\lommen_final_project\pizzabounce_help.png** |

**Program Flowcharts Design:**

View.onDraw

Sleep for given amount of time

Start Drawing Thread

Add default sprite to list: x and y positions in center of screen; x, y, and angular velocities 0.

Create empty list of sprites

Custom Canvas View included in XML

Start drawing thread

App Launched

View.onTouchEvent

Is the touched point contained by existing sprite?

No

Yes

OnPause

getXVelocity

getYVelocity

Add sprite to list with 0 initial velocities and at position where touch occurred.

Set sprite’s x and y velocities to obtained velocities.

Set sprite’s angular velocity to

x\_velocity/y\_velocity

Return

onPause

HelpButton.onClick

ClearButton.onClick

Return

Empty list of sprites and refresh view.

Switch to Help Activity layout that includes TextView with help text and back button.

Start Help Activity

View.onDraw

Update sprite’s x and y positions and angle according to velocities.

Yes

Switch direction of appropriate velocity.

No

Is the sprite’s edge touching the side of the screen?

Return

No

Yes

Increase sprite’s vertical velocity according to gravity.

Translate matrix according to sprite’s x and y position fields.

Use matrix to draw sprite on canvas.

Rotate matrix according to sprite’s angle field.

Sprites in list that haven’t been updated?

**User Evaluation and Suggestions:**

**User 1:**

User 1 is a third year computer science student who has no experience with Android (or other mobile application) development.

Praise:

The user really seemed to enjoy the app, she laughed the entire time she was using it. She commented that she really liked the title banner (i.e. the rainbow text at the top).

Suggestions/Criticism:

She didn’t offer any criticism, but she needed to ask me what she could do and how to do those things, which means that a help screen with these instructions would be helpful.

Response:

The help screen is now included in the final version.

**User 2:**

User 2 is a senior computer science student who has no experience with Android (or other mobile application) development.

Praise:

The user was impressed by the graphics. She said that they were high-resolution and that the animation was smooth and responsive. Like the first user, she had a lot of fun with the app, and tried to make the pizzas go as fast as possible.

Suggestions/Criticism:

She mentioned that it would be nice if it were easier to “catch” existing pizzas on the screen, i.e. the target size was larger. She would often try to catch a pizza but end up creating a new one by accident. She encountered several bugs when she tapped and held or dragged the pizzas: sometimes the pizza would disappear, stop moving entirely, or spin very, very fast.

Response:

The target size for existing pizzas is slightly larger, and the onTouchEvent function no longer causes crashes or strange when the user taps and holds a pizza. This was caused by referring to a VelocityTracker object that had not been initialized properly.

**User 3:**

User 3 is a graphic designer who works for the University. He has taken introductory programming courses, but has very little experience developing for web or mobile.

Praise:

The user liked the title text (i.e. the rainbow graphic at the top) and was impressed by the animation. I.e. he thought that the pizzas moved slowly and thought that the “physics” was impressive.

Suggestions/Criticism:

He thought that the rainbow graphic at the top of the screen should be bigger. He also had a lot of trouble creating new pizzas because he would try to tap too close to the edge of the screen.

Response:

Users can now create pizzas closer to the edge of the screen. Originally, the sides of the screen close to the edges were a “dead zone” where no pizzas could be created so that they didn’t accidently get initialized slightly off screen. Now, if a pizza appears off screen, it is immediately be moved on screen instead.

**User 4:**

User 4 is a senior majoring in Japanese at Western. She has no experience at all with any kind of programming or development.

Praise:

The user really enjoyed the app. Her feedback was “It was fun! But it didn’t seem to really have a point.” She made a game out of making the pizzas go very fast and then trying to “catch” them by tapping them, which worked fine. She also commented that she liked how the different kinds of pizzas appeared randomly.

Suggestions/Criticism:

One time, the app crashed when she tried to tap and *drag* (instead of swipe) a pizza.

Response:

As stated above (User 2), the onTouchEvent function has been debugged.

**User 5:**

User 5 is an exchange student studying business at Western. She has no experience with any kind of programming or development.

Praise:

The user was impressed by the animation and how fast it could go.

Suggestions/Criticism:

She said that the pizzas were a little too fast to keep track of and that they looked strange when they spun too fast. She needed help figuring out how to create new pizzas and move existing pizzas.

Response:

The help screen is now included in the final version of the app.