

Introduction

This is a 2D game.

You are a cardboard box. Your goal is to collect as much tape rolls as possible without catching on fire. You can only jump forward. You will change the direction only when you reach a wall. Player can touch the screen to charge a jump, and release to jump. Player can also shake the phone to hop over a small distance.

Description

Classes:

MainActivity: the activity holds the panel

Panel: the panel holds the game

GameThread: the thread holds the game

GameObject: a class that define the game objects

GameObjectSize: a class that define the size of a game object

Coordinates: a class that define the coordinate system

Floor: a game object that define the floor of the game

Player: a game object that define the cardboard box (i.e. the player)

TapeRoll: a game object that define a tape roll

Fire: a game object that define a fire object

FireWall: a game object that define a wall made of fire

PlayerOnFire: a class that define how it behave when the player is on fire

SoundEffect: a class that define the sound effects for the game

GameText: a class that define the texts shown in the game

Button: a class that define on/off buttons for the game

APIs:

android.content.SharedPreferences: to get and set the game data in the device (e.g. settings, highest score)

android.graphics.Bitmap: to hold the graphics for the game

android.graphics.BitmapFactory: to change the graphic files to the game graphics

android.graphics.Canvas: to draw the graphics and texts

android.graphics.Color: to set the colour of the game objects (e.g. the floor, the background)

android.graphics.Matrix: to flip the graphics

android.graphics.Paint: to define a paint for the game objects (e.g. the floor, the background)

android.graphics.Rect: to define a rectangle for drawing the graphics

android.hardware.Sensor: to detect the shake motion

android.hardware.SensorEvent: to detect the shake motion

android.hardware.SensorEventListener: to detect the shake motion

android.hardware.SensorManager: to detect the shake motion

android.media.AudioAttributes: to allow the application to play audio

android.media.SoundPool: to define the sound effects

android.os.VibrationEffect: to define the vibration effect

android.os.Vibrator: to allow the application use the device's vibrator

android.util.Log: to show debug message while developing the game

android.view.MotionEvent: to check if the user interact with the screen

android.view.SurfaceHolder: to hold a display surface

android.view.SurfaceView: to allow the panel extends SurfaceView

android.view.View: to set the game to full screen

java.util.Random: to generate random number

java.util.ArrayList: to create a list of game objects (e.g. for the fire)

Methodology:

New Features:

1. Use Sprite for animation
2. Check for collisions of objects in animation
3. Add audio in the application
4. Use sensors to control the animation
5. Use SharedPreferences to store data
6. Use Matrix to flip the Sprite
7. Hide the title bar and set the application to full screen
8. Add vibration to give feedback
9. Set to landscape mode

How to play the game:

Aim: to get high score

How to get score: you will get 1 point when the box touches a tape roll 

How to move the box:

- Shake the device to hop
- Hold on the screen to charge a jump
- Release to jump
- The longer you hold, the higher and further you will jump
- You can only move forward. You will automatically change the direction of going when you touch the side walls.



Programme Flow

- Start the game thread when started
- Create a panel that hold and display the game objects.
- Setup the game objects

Repeat the following steps:

- Check if user interact with the screen.
 - If a button is clicked
 - If mute button is clicked, toggle the sound setting and save it into shared preference
 - If shake button is clicked, toggle the shake settings and save it into shared preference
 - If the game is not started, start the game.
 - If the game is started
 - tell the character to jump and the strength of the jump
 - play the sound effect
- Update the game objects
 - Coordinate
 - Visibility
 - Sprite
- Check the collision
 - If the character collides with the tape roll
 - Tape roll disappear
 - Update the score by adding 1
 - Play the sound effect and vibrate the device
 - Re-generate the firewall
 - If the character collides with the firewall
 - Set the player on fire
 - Play the sound effect
 - If new max score, update max score in the shared preference
 - End the game and wait for new game to start
 - If the character collides with the floor
 - Set the acceleration of the character to zero
 - Slow down the x-acceleration according to the friction coefficient
 - If the character collides with the walls
 - Flip the sprite image
 - Toggle the direction of the player going
 - Respawn the tape rolls after the player is away from the walls
- If the phone is shaken, tell the character to hop and play the sound effect

Conclusion

Problems faced and solution:

1. Creating animation

Solution: Use Photoshop to create graphics and sprites. Then use `drawBitmap(bitmap, src, dst, paint)` to implement the sprites.

2. Change the direction of the character graphic

Solution: Use Matrix to flip the Bitmap

3. Check collision

Solution: `return ((obj1.x1 < obj2.x2) && (obj1.x2 > obj2.x1) && (obj1.y1 < obj2.y2) && (obj1.y2 > obj2.y1))`

4. Creating a wall of fire

Solution: create a fire object and use a firewall class to implement a firewall by holding an arraylist of fire

5. Make the player on fire

Solution: use a PlayerOnFire class to implement it by holding an arraylist of fire

6. Unable to check sensor events in the Panel

Solution: check in MainActivity and call function in the Panel

7. Hard to create a Button (that Android provide) on the Panel

Solution: make my own button as a game object

8. Initially facing a problem

Solution: to fully understand the current situation and the current program. Try out different methods and do not afraid of failing.

9. If still cannot try the solution out

Solution: think out of the box and try different solution even it does not sound right

10. Still stuck in the problem

Solution: search on the Internet with the correct keywords or even ask questions on forums.

Understand more. Try more. Learn more.

Further Development

The following can be easily done with my current knowledge but they are not done yet due to insufficient of time:

1. Add a button to disable vibration
2. Improve the Graphics
3. Add more sound effects and add background music
4. Optimise the codes to improve readability
5. Optimise the codes to reduce resources
6. Move the gap of the firewall around when the score is about a specific value
7. Provide different skins for the character and tape rolls
8. Local multiplayer

The following can be improved with some more knowledge:

1. Publish the application to Google Play
2. Synchronise the highest score to Google Play
3. Optimise on the codes to reduce program size
4. Improve the onPause function in order not to just kill the application to prevent game crash.
5. Provide fully customisable skins
6. Online multiplayer

References / Appendices

This game is 100% my own idea. I swear I did not copy the idea of the game from others. If you found any question, please contact me and I can explain everything in my game. (don't just deduct my marks)

Some idea of coding is found on the Internet:

Mirror the bitmap: <https://stackoverflow.com/questions/36493977/flip-a-bitmap-image-horizontally-or-vertically>

Sprite: <http://www.edu4java.com/en/androidgame/androidgame4.html>

Shake detection:

https://www.youtube.com/watch?v=fPa9Sev7il8&ab_channel=SarthiTechnology

Full screen: <https://stackoverflow.com/questions/26543268/android-making-a-fullscreen-application>