GAME DESIGN DOCUMENT

Kick The Cone

Don’t save him

Prepared By:
Andréia

Last Updated:
26/04/2020

TABLE OF CONTENTS

GAME ANALYSIS 3

PITCH 3

GENRE 3

PLATFORM 3

TARGET AUDIENCE 3

STORYLINE & CHARACTERS 4

GAMEPLAY 7

Overview of Gameplay 7
Player Experience 7
Gameplay Guidelines 7
Game Objectives & Rewards 7
Gameplay Mechanics & Code Organization 8
Level Design 10

CONTROL SCHEME 12

GAME AESTHETICS & USER INTERFACE 14

SCHEDULE & TASKS 18

Game Analysis

Lost within random maps of Copacabana Layla finds herself in a crusade to kick every single cone she finds in her way. The longer she is in this place the more she feels the need to go faster. In this game you need to help her kick as many cones as possible without falling off the road or hitting a dangerous obstacle. The obstacles are randomly placed in those maps and restart every time they appear, also, the maps are randomly chosen pre-made maps.

The player must use his controls to help her dodge/jump obstacles and **slide his way** into those orange bastards. It's not an easy task so don't forget to fuel your hatred toward those traffic cones.

Pitch

Kick The Cone is an endless run made for Android with the main idea being not just to survive, but to kick your traffic enemies.

Genre

Endless running.

Platform

Android.

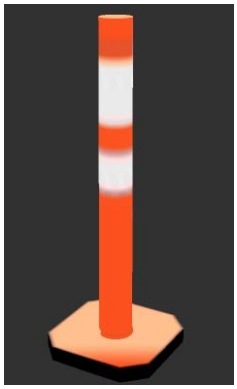
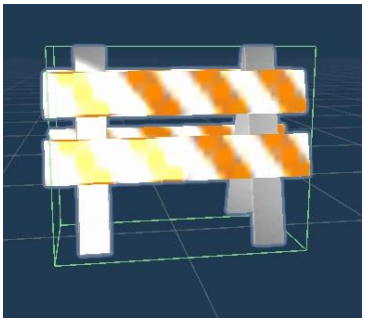
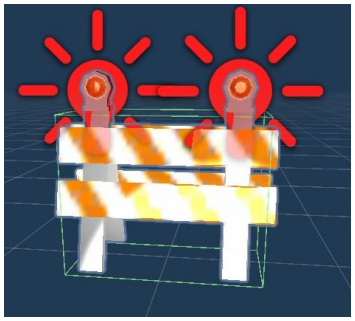
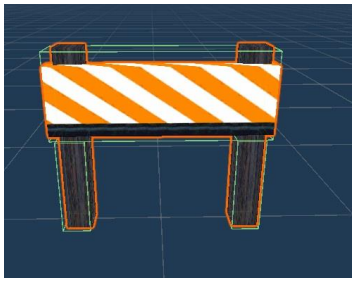
Target Audience

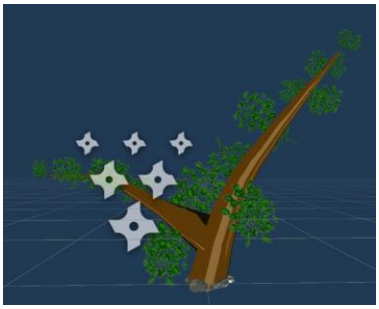
Android users bored of other endless running games and looking for a challenge. Really anyone with the age from 6 (you might need to read) to infinite.

Storyline & Characters

Within the Copacabana realm Layla found her attention being drawn by an orange entity that had stripes on it. She is aware of the 4th wall so she dislikes going backwards. Knowing her destiny as a game character she just wants to let out all of her rage.

Character	Description	Characteristics	Misc. Info
Layla 	Layla is a playable character that can jump incredibly high and slide in a very weird way.	Adorable anime look will make you feel very sad when fails to complete her task.	Don't be fooled by this cute look, this woman is a cone killer.
Cone 	A target that don't move.	Its orange and white and it flies away when kicked.	Very scared of its destiny.
Big Cone 	Can't move too.	Still orange with stripes.	Has been kicked so many times it doesn't even care.

VERY THIN Cone 	Still can't move.	Well, orange with stripes.	Hard to kick.
Hurdle1 	It is challenging you.	Might have stripes but it does not fly.	Don't touch it, it is an introvert and will return the favor as the third Newton's law.
Hurdle2 	It used to have a red shining light but can't shine anymore because your mobile can not handle too many lights.	Has stripes and does not fly too.	It lost its shine, it is mad and does not like touches.
Hurdle3 	You think you can go under it? I don't feel like this is a good idea.	Striped shirt team, but not on the mood for flying.	It's boring and envies his brother that has more wood.

Hurdle4 	Hurdle3's brother.	Many stripes but no flying capability.	Woody wood wooding.
Tree 	Especially made for you to be able to jump/slide over/under its branch.	Can be spawn in different rotations. Lovely tree well rooted in the ground.	Scent of nature.

Gameplay

Overview of Gameplay

Alike others endless running games, here the player needs to dodge different elements but it does not count score as the time runs. The way to win points is to hit the enemy (traffic cone) while sliding. The game is made for Android with the use of shaders and object pooling to optimize it.

Player Experience

You will start in a menu screen, with the following buttons:

- A config button that will lead to some volume sliders and if you want to reset the tutorial, a button to do so (only available if the tutorial has been done before);

A play button that will do one of the following: if it is the first time playing, it will load the tutorial. Else, it will load the main scene (real deal);

- A high score button that shows the highest score ever archived by someone on this instance of the game, and by checking the score, a new button appears that allows resetting the score;
- A quit button.

As for the tutorial there's explaining sentences on the bottom of the scene that will guide on what needs to be done to surpass the obstacles.

At the end of the tutorial a new in-game menu appears allowing you to finish the tutorial (skip). Also, this button can be reached by pressing the stop button at the top right corner. This will stop the game and load this in-game menu. To return, to the game the player needs to press the pause button again.

Only when you end the tutorial and play the real deal, the game will store the score made.

Remember that at anytime you can pause the game by pressing the pause button on the top right corner.

Gameplay Guidelines

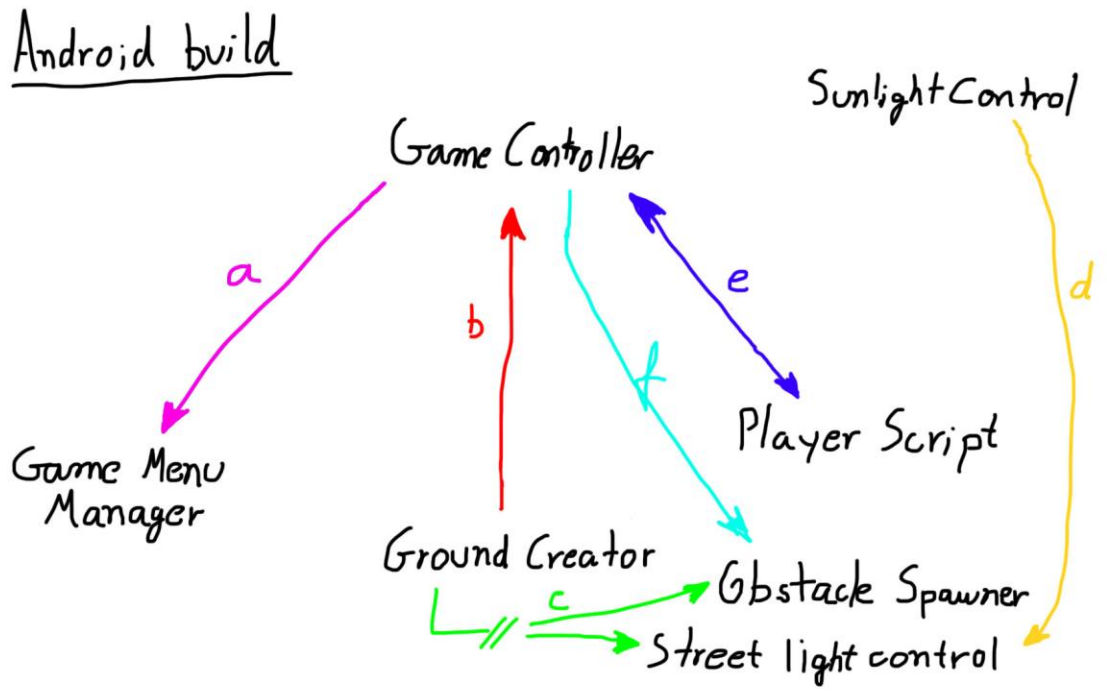
This game does not have any complex interactions besides the game over menu mocking you for the reasons you died.

Game Objectives & Rewards

The objective is to kick as many cones as you can, as the game starts (real deal, not tutorial) there will be no obstacles for at least 2 seconds and then the chances of an obstacle (or a cone!) appear go from 50% until 5 seconds, 70% until 10 seconds and 84% later on. This can be a great reward as there will be more chances that cones will appear. On the other hand, deadly obstacles will appear too.

Gameplay Mechanics & Code Organization

Character Attributes	
Character	**Actions Available**
Layla	Jump: slide your finger up. Slide: slide your finger down. Rotate: slide your finger in the direction you want to go (left, right). Move sideways: rotate your device left or right relative to your wrist's axis.
Cone	None, he is just born to suffer endlessly.
Game Modes	
Tutorial	On your first game run, you will encounter this tutorial, which is supposed to help you understand the mechanics. After doing it you will access the Real Deal. At any moment you can remake this tutorial by going on settings and resetting it. Also, you can skip it by pressing pause on top and the skip button.
Real Deal	The real deal of cone killing.
Scoring System	
Cones Kicked	**How it's Awarded & Benefits**
The number of cones hit by the player while using the slide motion.	You need to slide your finger down the screen to perform the motion to slide. Don't forget to aim that motion with your phone's accelerometer.



On the real deal scene, we have some codes, and here is how they connect:

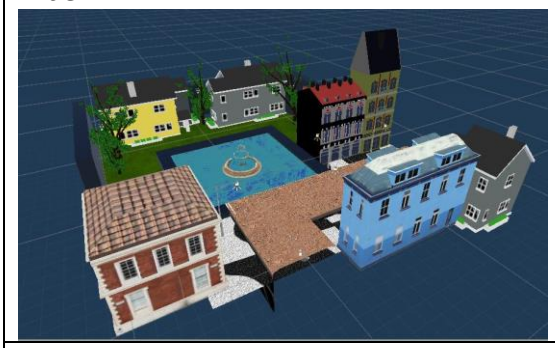
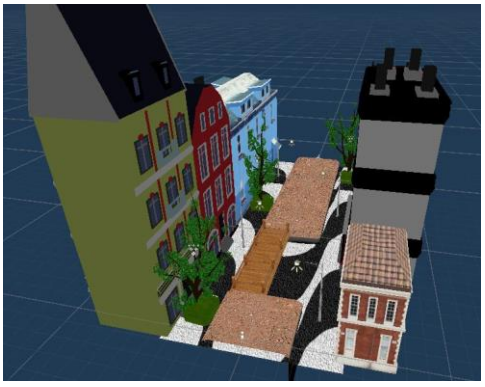
A) The *Game Controller* receives the collisions and sends to the *Game Menu Manager* so we can have context related game over sentences and also have the score on the screen's bottom.
B) The *Game Controller* receives the track where the player is standing on to check if the player is moving in the wrong way.
C) The *Ground Creator* starts two clones of each track and place them randomly in front of the player as he moves. At the time each ground of the track is activated the *Obstacle Spawner* instantiates an obstacle randomly, if deactivated it will delete the spawned obstacle and wait for another call. As for the *Street light control*, it will check each time his game object (light) gets activated if the sun is down. If it is, it will activate the lights, and the opposite too. Unfortunately, this last code is almost useless since a moving source of light would be very consuming on Android so I did not allow the light to move in the scene.
D) The *Sunlight Control* was a script made for moving the sunlight and making a day-night cycle. It also did change the skybox when it was night, so we would have a starry night. This script has a public function where *Street Light control* could receive the boolean representing if the light should be on.
E) Here the *Game Controller* receives the player's center of mass, since the game does not have a walking upwards or downwards mechanic I can assume that if the player is under a certain level he is just falling out of the track. Also, the *Game Controller* receives the player's public variable forward speed and manipulates it in a way that it increases as the time in-game does, both scripts have limits for the speed, being in the *Game Controller* the max 6 and in the *Player Script* the magnitude of the Rigidbody's velocity will max at 8. Another connection between these two is that the *Game Controller* receives the player collisions and sends back: the respective animation that the player should do, disables the

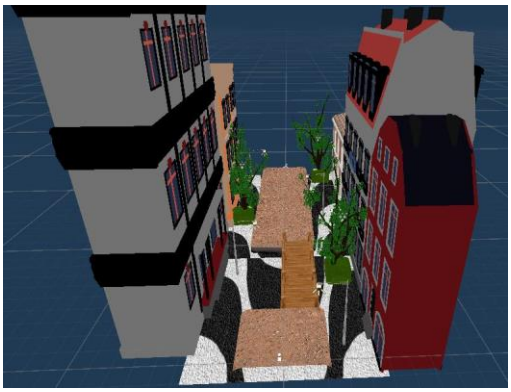
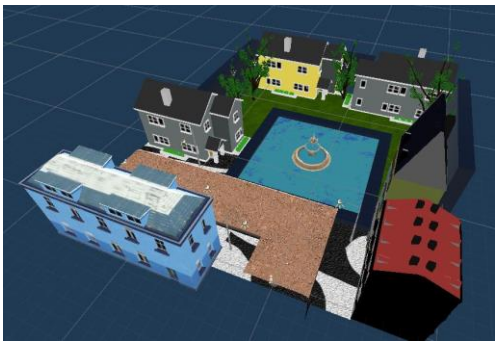
Player Script update in case of game over. It also checks if the player has hit the cone while sliding.

F) The *Game Controller* sends the probabilities of the obstacles to spawn to the *Obstacle Spawner*.

Level Design

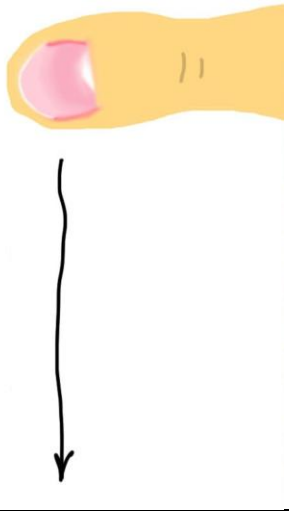
The game is made of prefabricated tracks that randomly appear and the obstacles in them also appear randomly. The building and trees give the environment life, at the same time that they offer danger for the player. We also have street lamps that in the initial versions served the purpose to light the street at night time.

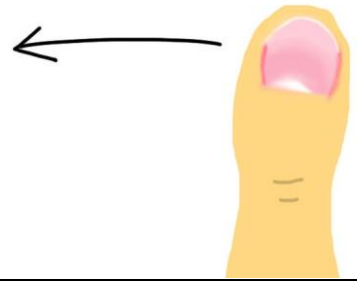
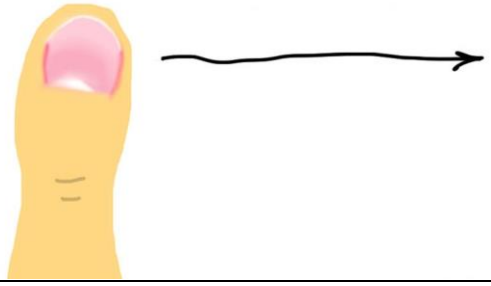
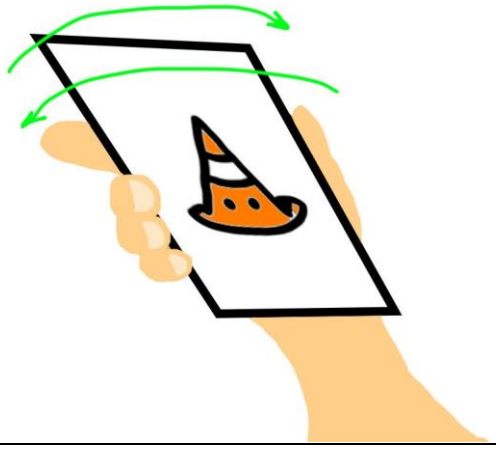
Tracks	
Track1 	This first track has a big bridge, he does not seem dangerous like this, just wait until we fill it up with obstacles.
Track2 	This track has a curve, it goes right, and if you don't go as it demands, it might as well make you wet.
Track3 	In this one we have a bridge on the left side.

Track4 	The bridge is on the right side so you might as well be there.
Track5 	Ooh, so you think you escaped the right curve? Now take this left.
Track6 	DOUBLE TROUBLE. Two bridges just because I felt like doing so.

Control Scheme

This game uses touch input as well as the accelerometer.

Input	Action it Performs
Touch the screen and slide up. After sliding, remove the finger from the screen. * 	Jump.
Touch the screen and slide down. After sliding, remove the finger from the screen. 	Slide.
Touch the screen and slide left. After sliding, remove the finger from the screen.	Rotate left.

	
Touch the screen and slide right. After sliding, remove the finger from the screen. 	Rotate right.
Rotate your phone around your wrist (accelerometer). 	Move sideways.

*Why does the user need to remove the finger from the screen? The script saves the vector representing the first touch you did and then it waits for you to remove your finger so it knows you have completed the action you wanted. Then it makes the difference between the two vectors and check which direction you moved your finger the most.

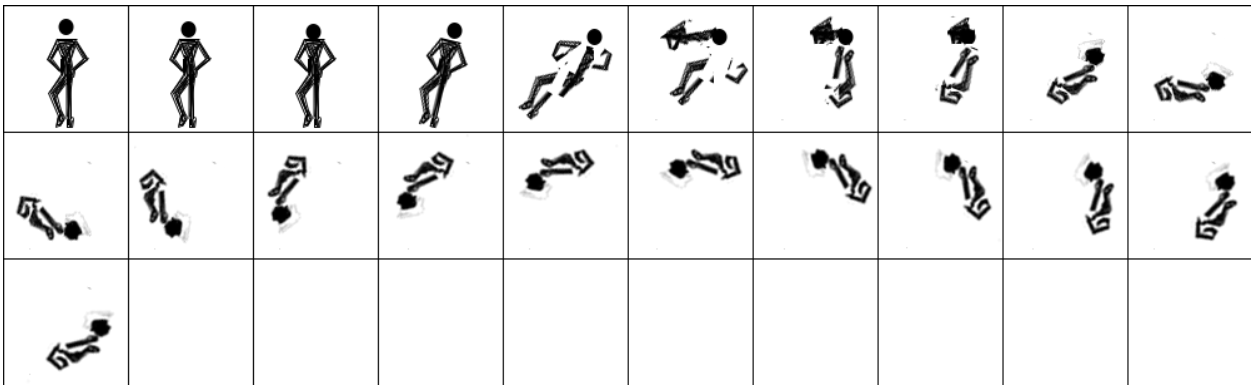
Game Aesthetics & User Interface

The game takes inspiration in a city near the beach, even if you can't see the sea.
This is the representation of the sidewalk on Copacabana's beach in Rio de Janeiro:

In the first image we have just the layout, on the second I added some noise to make it look more realistic.

Here I use some comic inspiration on the drawn by hand main menu buttons as in the animations frame by frame made for the in-game menu.

Button background for the main menu.



The frame by frame grid made for the Restart button.

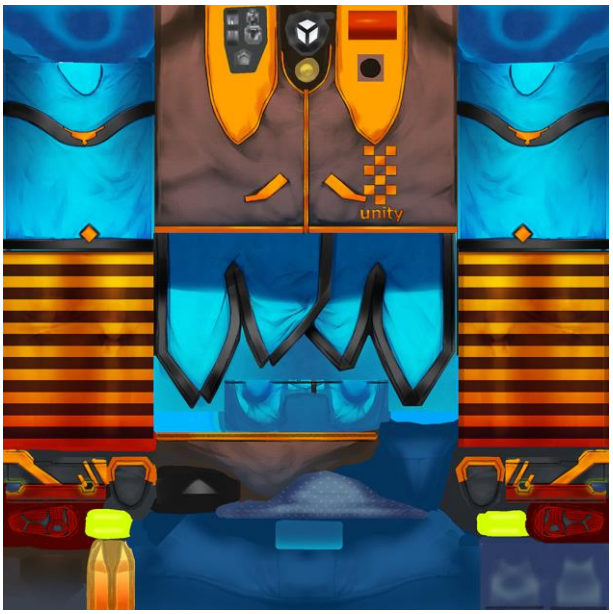
The frame by frame grid made for the Main Menu button.

This game uses Cel shading.

Cel shading example.

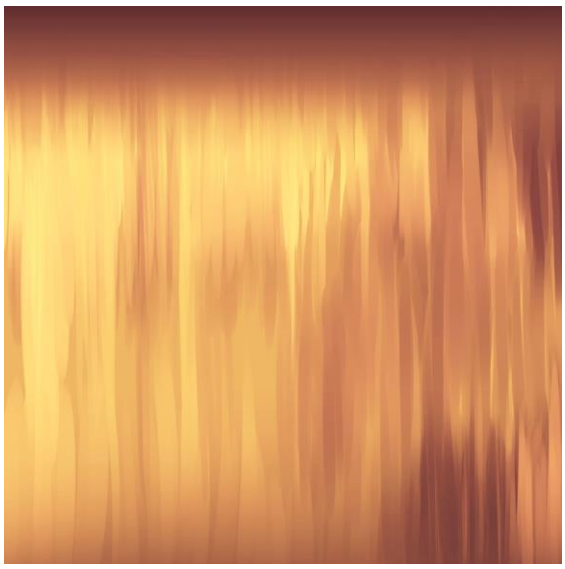
The look on Layla (formally known as Unity-Chan) was repainted to darker colors, since the rest of the game is very bright.

Body look:



Old x new body texture.

Hair look:



Old x new hair texture.

This happy environment allows the player to enjoy his ride kicking cones.

Schedule & Tasks

Since this game was made with the intent of learning and I (dev) expected nothing from it. Also, when I started, we did not need to make a GDD so I didn't make a schedule. **But for the sake of documenting, this game took me about 5 months to make.**