

Rota do Tropeiro: a cultural entertainment game for Android platform

Estevan Strapazzon Ruschel ¹
Rafael Rieder ¹

Abstract: This paper presents a cultural entertainment game entitled “Rota do Tropeiro”. The aim of our article is to briefly present well-known tools that can be integrated and used by beginners to develop games with quality and optimized performance, especially for the Android platform. The document describes our game prototype, showing techniques and methodologies applied and highlighting new challenges. As a result, it was produced a mobile game for end-users and creative stuffs like animations, geometric models and artificial intelligence scripts.

Keywords: Android, culture, endless runner, game, mobile devices, Unity.

Resumo: *Este artigo apresenta um game de entretenimento, de cunho cultural, intitulado “Rota do Tropeiro”. O objetivo deste trabalho é apresentar, rapidamente, ferramentas bem conhecidas que podem ser integradas e utilizadas por iniciantes para desenvolver games de qualidade e de desempenho otimizado, especialmente para a plataforma Android. O documento descreve o protótipo, mostrando as técnicas e as metodologias empregadas, e os desafios encontrados. Como resultado, obteve-se um game interativo para dispositivos móveis, destinado ao público em geral, com materiais criativos, como animações, modelos geométricos otimizados e scripts de Inteligência Artificial.*

Palavras-chave: *Android, cultura, dispositivos móveis, endless runner, game, Unity.*

1 Introduction

Games are entertainment software consisting of various Computer Science methods of different fields, such as computer graphics, artificial intelligence and computer network. Techniques from different areas of knowledge such as arts and linguistics are also used in their design.

The game market has been growing around the world, especially for education or training purposes. For example, America's Army (<http://www.americasarmy.com>) is a training game produced and used by the US military [1]. Students and teachers can create own “educational games” using Scratch (<https://scratch.mit.edu>), a free programming tool created by MIT Media Lab [2].

According to the ESA [3], consumers spent more than US\$ 21 billion in 2013 in the acquisition of games. Specifically in US, the survey indicated that 59% of Americans play games, using consoles (68%), smartphones (53%) and/or wireless devices (41%).

With the popularity of game apps in smartphones, the mobile gaming market is seductive. Game developers can present your casual games with minimal cost and achieve a high number of users in app stores like Google Play and Apple Store. Moreover, this kind of game does not need large teams or long periods of development time, and provides a significant monetary return [4].

With this in mind, this paper presents a cultural entertainment game prototype developed for the Android platform. Our solution explores scene elements related to the culture of the Brazilian gauchos, the cowboys of the South American pampas.

Our purpose is the endless runner game and provides some resources like 3D animations, audio, collision detection, accelerometer controls and artificial intelligence. In this game, the player character is constantly running until the end of stage, dodging obstacles and executing tasks along the path.

¹Universidade de Passo Fundo, UPF, Campus 1 - BR 285 - Passo Fundo (RS) - Brasil
{tevors@gmail.com, rieder@upf.br}

This document is organized as below: Section 2 presents related work using Android platform; Section 3 shows materials and methods applied in game creation process; Section 4 describes our game; and Section 5 presents our conclusions about our solution.

2 Related Work

Kim, Kang and Song [5] show the development process of the Konu game, a Korea's traditional board game, in mobile version. Authors make a comparison between the use of the game in the traditional way and the smartphone. As a result, they point out that the mobile approach was well accepted by the users, and it is interesting form to preserve the Korean culture and the original rules of the Konu. The survey also explains differences between developing games on the PC and smartphones.

Bangun and Haswanto [6] present an Indonesian game-play named Tari Lima Serangkai, which aims to support the learning of Karonese culture. This study only includes visual and musical aspects of the game-related content of the culture.

Tag and Seek [7] is location-based game which leads a traveler through Tainan City in Taiwan, using Near Field Communication (NFC) technology. The traveler's task is to find friends who are hiding at tourist sites in the city marked with NFC tags. According the authors, the game lets the traveler experience culture, gain knowledge about sites in the city and meet local citizens, without necessarily visit it physically.

Regarding cultural games, we can highlight Sok & Sao's Adventure produced by DirexPlay [8]. It is an action-adventure game for dengue prevention and control, where the aim of the game is to eliminate mosquitoes. In addition, the scenarios refer to the Southeast Asian countries, providing to the player the contact with elements of culture of these regions.

Other commercial cultural games are Plants vs. Zombies [9] and Total War: SHOGUN [10]. The first is a tower defense game where the player defends your home of zombies using a variety of plants. The game explores the musical culture, associating the behavior of plants with rhythms of the pop classic songs. The second game is based on the famous series of strategy games and explores the medieval Japan culture. Its interface allows organizing troops, managing combats, controlling economy, and playing in multi-user model.

However, we noted that the study and the development of the culture games are not a common practice in scientific literature and/or in app stores. It is plausible that researchers and industry are not interested in cultural games, since the public who may consume them is restricted and regional, and it requires content domain. Moreover, this kind of game can be a promising market niche, especially for new developers, if adequately exploited.

In this context, it is important to consider aspects of globalization. For example, foreigners could access a cultural game application and remind values and traditions of their homeland. Another possibility is the dissemination of the region culture for people of different nationalities, encouraging sectors such as tourism, education or entertainment. These were some reasons for the study of well-known tools and the development of a Brazilian cultural game, presented in the following sections.

3 Materials and Methods

Android operating system had been chosen for this project because it is the market leader in smartphones and tablets and, consequently, a considered number of consumers. According Gartner, Android is the world's leading sales, with 80.7% of the amount, followed by iOS with 15.4% and Windows Phone with 2.8% [11].

As development environment, we used Unity engine (<http://www.unity3d.com>). This game engine has a multi-platform interface that allows the creation of interactive solutions using OpenGL ES for Android. It has rich documentation on samples and an extensive and active community. For coding, we adopted the use of C# language in MonoDevelop.

For geometric modeling, we used Blender (<http://www.blender.org>), an open source tool used to create animated films, visual effects, animations and games. Routes and animations were created using ITween library (<http://itween.pixelplacement.com>), a free tool for Unity developed by PixelPlacement. Finally, we opted

for the T4M library (<http://www.azert2k.fr>) to import and convert terrain models.

4 Game Development

Our prototype is entitled “Rota do Tropeiro”, an endless runner game which explores scene elements related to the culture of the Brazilian gauchos, the cowboys of the South American pampas. The player can control the main character by simply tilt your device sideways. The goal is simple: get the highest possible score.

4.1 Virtual Environment

During the development process we created several versions of the virtual environment, based on landscapes of Southern Brazil. Initially, terrain and background scene were built from two rectangles with light textures. Including 3D models, the character perspective in relation to the scene had no desired realism. In doing so, we decided to use the Unity terrain object. Figure 1 shows the first versions of terrain.

The use of this resource presented significant visual results, but the 3D model needed a complexity reduction for mobile devices. In order to correct this problem, we used the T4M tool, converting to a new simplified terrain (Figure 2).

During this process, problems in scene perspective also were detected, related to the use of a simple textured cube as sky object. As a solution, we adopted to use the Unity skybox. This effect can also be seen in Figure 2.

In an effort to provide a rich scene, we included sprites of plants and rural objects in the virtual environment. Grass tussocks, wildflowers and sections of farm fences were dispersed in the scene (Figure 3). These objects and the terrain are loaded only once, and relocated throughout the experiment, in order to improve game performance.

Figura 1: First versions of the terrain

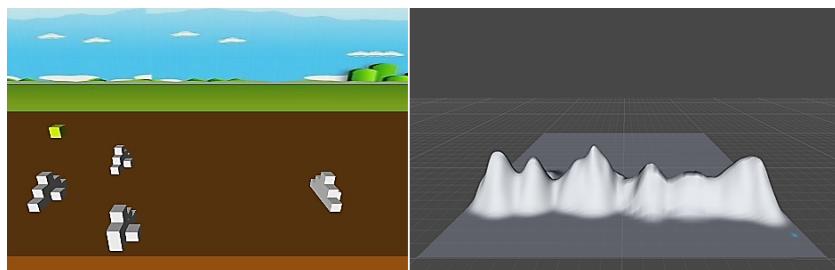


Figura 2: Final model of the game terrain



4.2 Obstacles

The presence of obstacles is one of endless runner game features. In order to develop this solution, we used 3D models available on Unity Asset Store (<https://www.assetstore.unity3d.com>). These models were combined to create complex obstacles, representing camps, log fences and haystacks, presented in Figure 4.

Figura 3: Terrain details

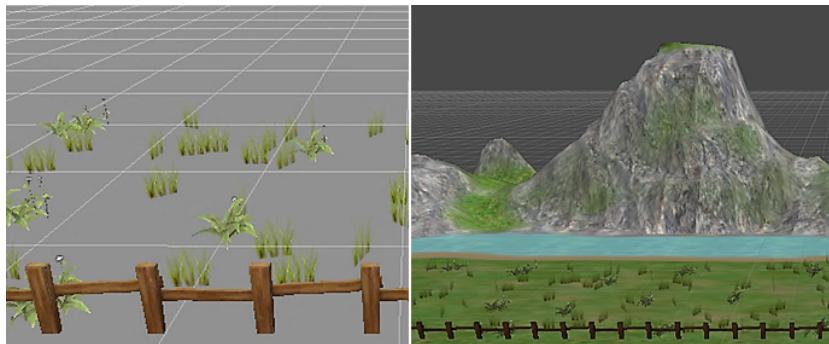
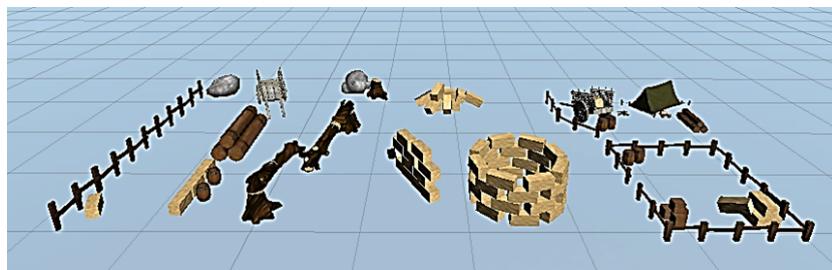


Figura 4: Game obstacles



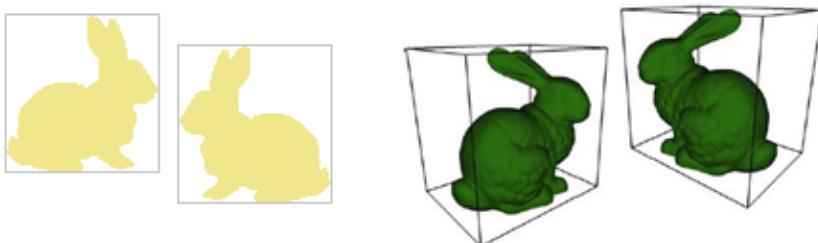
A script controls the creation and the positioning of obstacles, considering the main character's position or the distance traveled by it. Randomly, the objects are chosen and rendered in the next frame. After the main character pass a certain section of objects, another script detects which elements are outside of the field of view, and stops their redraw.

Each model has its own elements to detect collisions. Usually, these elements can be rigidbody and collider components, which are Unity resources enabled to control collision detection. Edges of bounding volumes can be used to detect collision between objects quickly. Bi-dimensional models use envelope structures and 3D models use bounding boxes. Figure 5 shows how the collision detection techniques are represented.

Camps are organized in a circle format, in order to move the main character sideways, blocking jumps. In contrast, log fences block the travel in z-axis, forcing the player to use the jump to pass them.

Haystacks are created in the scene only when the supporting character (cow) shows up. They have various shapes and consist of grouped cubes. These elements can be displaced by the supporting character after a collision. In these cases, the haystacks are thrown in different directions, and they can hit the main character.

Figura 5: Collision detection using bounding volumes: envelopes for 2D models (left side), boxes for 3D models (right side)



4.3 Main character

The main character of the game consists of a riding set, as can be seen in Figure 6. The horse use the skeletal animation technique, in which a character is represented in two parts: a surface representation used to draw the character (called skin or mesh) and a hierarchical set of interconnected bones (called the skeleton or rig) used to animate (pose and key frame) the mesh [12]. Figure 7 shows quickly two horse animations (running and jumping) created in Blender.

Figura 6: Horse and rider models

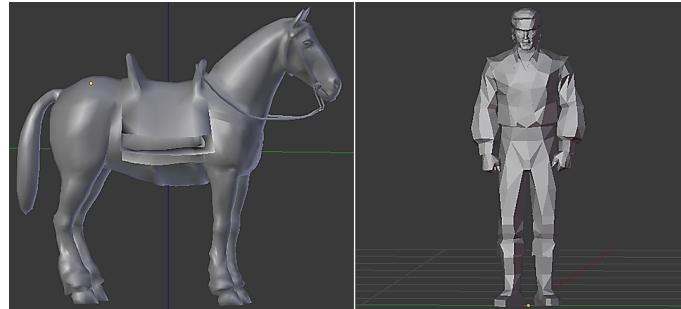
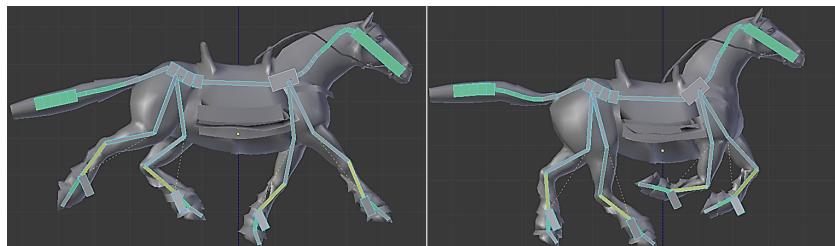


Figura 7: Skeletons for horse running and jumping



In Unity project, we assigned a script for the horse model, in order to start/stop animations, control the speed and distance traveled, add points in score, determined behaviors after a collision, and control horse movements using accelerometer.

To detect horse collisions, we added collider components in the skeleton bones. In every bone, we also added a simple script that sends a message to other points of collision, in order to disable new collision tests and stop animations. In this moment, the ragdoll is enabled.

Ragdoll is a type of procedural animation that considers physics in order to determine the model behavior. To apply this technique, we used rigidbody components associated for each bone, and assigned different body mass values for the horse building blocks (e.g. horse paws should be much lightweight than the head or body). In order to move the skeleton according the physics, we added character joint elements, limiting specific movements and applying balanced forces to the animation does not seem surreal.

Particle system and audio resources were also added to the horse. In these cases, we applied dust simulation and horse gallop sound effect. Finally, we also considered the use of less material to optimize the graphics processing in smartphones. Figure 8 presents the final version of horse model.

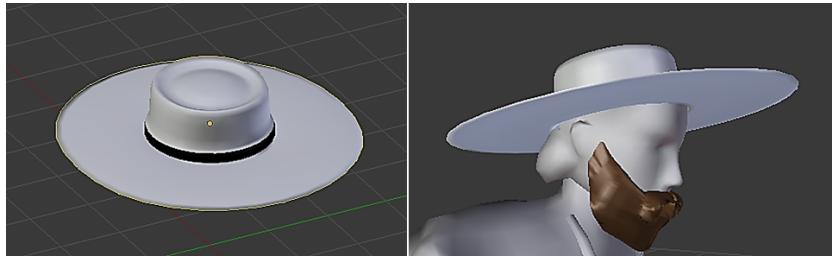
On the other hand, the rider model used in our project was found in a free stuff repository. In order to adapt it for our purposes, we added beard and hat (Figure 9), and manipulated its pants, as a means of simulate the traditional pants used by Brazilian gauchos. We made these changes in Blender.

The rider model also use skeletal animations: riding, jumping and lassoing. The first two animations are associated to the riding set, while the third is a feature fired when the rider use the boleadoras during the game.

Figura 8: Final version of the horse model



Figura 9: New rider's hat



Boleadoras is a type of throwing weapon made of weights on the ends of interconnected cords, designed to capture animals by entangling their legs. In our game, this object (Figure 10) is presented in a pole only when the supporting character (cow) appears. To pick up it, the riding set must collide with the pole in order to attach this weapon in the rider's left hand. After this, a torque animation starts, where the rider gives the balls momentum by swinging them and then release.

When the player decides to release the boleadoras, our application waits for a touch screen to define a throwing force. This force is useful to the next arm's animation to determine the range distance. If the cow is hit, an action is triggered in the cow script; otherwise the boleadoras falls on field. After a while, a new post is instantiated for the rider pick up and tries again the same move. It is important to note that the boleadoras only appear in the scene when the cow is also presented.

The rider model also has colliders, ragdoll, rigidbodies and character joints, as well as a script that controls its actions. Figure 11 shows the riding set finished.

4.4 Supporting character

In contrast with other endless runner games, we decided to insert a supporting character to be pursued by the main character. Considering the gaucho culture, we opted for the inclusion of a cow object in our approach.

During the game, the general game's aim is to get as far as possible before the main character dies, avoiding obstacles in order to increase the score. In addition, new extra challenges to get points are lasso and capture the cow when it appears on the scene. From this moment, the player's aim is to catch the boleadoras and try to hit the cow. In case of slow or no execution of this action, the cow has a chance to behave strangely or flee.

Figura 10: Boleadoras' model

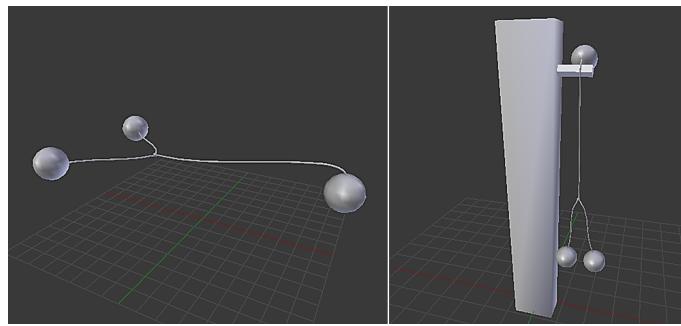


Figura 11: Riding set



The cow model was also found in a free stuff repository, including geometry, texture, animation, gallop sound effect, rigidbody, ragdoll, dust effect and colliders. When it is drawn in the game scene, a mooing sound is triggered, and the obstacles are changed to haystacks. Figure 12 shows the cow geometry.

The cow is the only character with artificial intelligence resources (AI) defined in script. In games, AI is used to build and describe the behaviors of elements that move and act by themselves, be it a non-playable character, a seeking rocket, or a cow in our case. This game has eight cow behaviors (walk, zigzag walk, run, fast run, escape, jump, stand and fall) controlled by ITween functions. The idea is to provide the player dynamic obstacles and spice up his experience.

A finite state machine (FSM) manages the model behaviors, considering task time or action executed. FSM are models of computation used to abstract an object or machine in such a way it can only be in a unique state at a time, picked in a set of states. For game development, it's quite natural to think about an AI as an entity with a set of behaviors or states. In our game, the states have progression periods, in order to gradually increase the level of difficulty to capture the cow.

In order to control the cow behaviors, a timer is used to calculate the moment and position to draw this object. When the riding set is near, the cow will adopt an escape behavior defined by AI script. If collides during the escape, haystacks are displaced on the road in different directions, without causing cow damages. During the pursuit, the cow runs at a fixed speed, while the riding speed increases slowly.

During the encircle tries of the cow, AI states are alternated considering the riding set position and the distance between it and the cow. Behaviors like zigzag walking and running fast can be applied. If the rider hits the cow using the boleadoras, the ragdoll cow is triggered and it falls on the ground. Figure 13 shows the capture moment of the cow.

Figura 12: Cow model

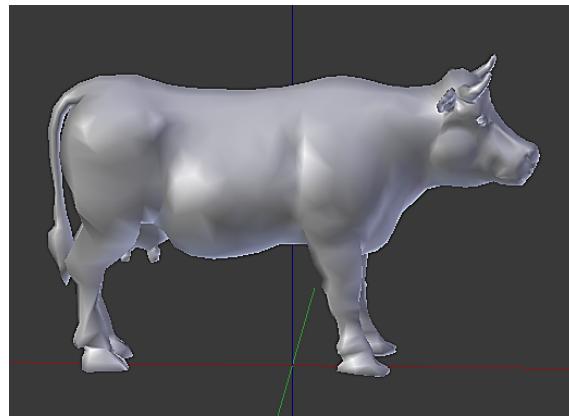


Figura 13: Capturing the cow by the riding set



4.5 Game User Interface

The graphical user interface consists of menus (Figure 14) and user controls created with Unity resources (text mesh and GUI class). User controls touch screen and accelerometer support.

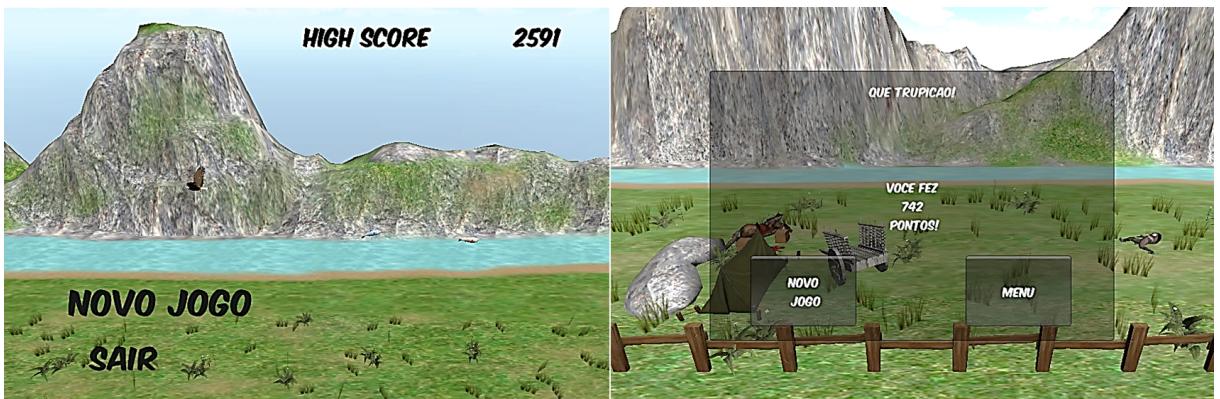
The main menu allows starting new game, exiting game and consulting the help. Its background presents a game terrain and the highest score HUD (in the upper right corner).

The final menu only appears when the main character dies. It shows buttons to start new game or exit, the obtained score and a custom phrase depending on the score. In order to support different Android devices, the menus are positioned and sized according to the screen resolution.

The game controls are detected by menu and horse scripts. The horse of the riding set has two types of controls, touch and accelerometer. The touch controls divide the device screen in half: when a touch is detected on the left half, the horse begins its jump animation; touch on the right half only is useful to control the release of the boleadoras.

At the top right of screen, it is also a pause button. If pressed by the first time, interrupts the game; if pressed again, the interaction continues.

Figura 14: Game menus



Lateral horse movements are controlled by the balance accelerometer. During the level start, the application calculates the accelerometer position relative to the mobile position, and defines it as the horse's center of balance. According to the device tilt, the horse moves sideways, up to a maximum value, controlled by the collision with the river or the fences present in the scene.

5 Conclusion

This paper presented “Rota do Tropeiro”, a mobile game development process on the Android platform, integrating different resources. Our study indicated well-known tools can be used by beginners or young entrepreneurs in order to develop high quality games for mobile devices. Unity, for example, proved to be easy to learn through its rich documentation.

During the initiation phase of this project, we noted the exploitation of cultural traditions in games is rare. This detail influenced our choice in the element games, as the models for horse and rider, in order to represent equipment and clothes used by gauchos. The final result was a distinct entertainment game, absent on the market.

For future work, we intend to create new obstacles and scenes, simulate day and night periods, offer a rich graphical user interface and implement a multi-player mode. In addition, we intend to evaluate usability and gameplay with subjects groups.

Referências

- [1] ALLEN, R. America's army and the military recruitment and management of 'talent': An interview with colonel casey wardynski. *Journal of Gaming & Virtual Worlds*, Intellect, v. 6, n. 2, p. 179–191, 2014.
- [2] SIMÕES, J.; REDONDO, R. D.; VILAS, A. F. A social gamification framework for a k-6 learning platform. *Computers in Human Behavior*, Elsevier, v. 29, n. 2, p. 345–353, 2013.
- [3] ESA, T. *Essential Facts about the Computer and Video Game Industry*. Washington, USA, 2014. ESA_EF_2014. Disponível em: <http://www.theesa.com/wp-content/uploads/2014/10/ESA_EF_2014.pdf>. Acesso em: 15 jun. 2015.
- [4] MARSH, D. *Nine Paths To Indie Game Greatness*. 2008. Disponível em: <http://www.gamasutra.com/view/feature/131952/nine_paths_to_indie_game_greatness.php>. Acesso em: 15 jun. 2015.
- [5] KIM, Y.-J.; KANG, S.-W.; SONG, T.-S. A design and development of korean traditional board game on android environment. In: *Future Information Communication Technology and Applications*. [S.l.]: Springer, 2013. p. 379–387.

- [6] BANGUN, D. A. N.; HASWANTO, N. Indonesian gameplay “tari lima serangkai” as leads the local culture content. In: JSSD. *Proceedings of the 5th International Congress of the International Association of Societies of Design Research*. [S.I.], 2013.
- [7] ARKENSON, C. et al. Tag and seek: a location-based game in tainan city. In: ACM. *Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play*. [S.I.], 2014. p. 315–318.
- [8] LAB, D. *Sok and Sao's Adventure*. 2015. Disponível em: <<http://direxplay.com>>. Acesso em: 15 jun. 2015.
- [9] POPCAP. *Plants vs. Zombies 2*. 2015. Disponível em: <<http://www.popcap.com/plants-vs-zombies-2>>. Acesso em: 15 jun. 2015.
- [10] ASSEMBLY, T. C. *Total War: SHOGUN 2*. 2015. Disponível em: <<http://www.creative-assembly.com/game/shogun-2>>. Acesso em: 15 jun. 2015.
- [11] GOASDUFF, L.; RIVERA, J. *Gartner Says Smartphone Sales Surpassed One Billion Units in 2014*. 2015. Disponível em: <<http://www.gartner.com/newsroom/id/2996817>>. Acesso em: 15 jun. 2015.
- [12] SORIANO, M. *Skeletal animation*. 2009. Disponível em: <http://alumni.cs.ucr.edu/~sorianom/cs134_09win/lab5.htm>. Acesso em: 15 jun. 2015.