

# The Experience of Developing the Game Hortalândia

Filipe Barbosa Nunes<sup>1</sup> Cleverson de Lima Puche<sup>1</sup>  
Autran Macêdo, Ph.D<sup>2</sup> Pedro Frosi Rosa, Ph.D<sup>2</sup>

<sup>1</sup>Computer Science Undergraduate Course, Federal University of Uberlândia, MG – Brazil

<sup>2</sup>School of Computer Science, Federal University of Uberlândia, MG – Brazil

## Abstract

This paper presents the development of an educational game for kids and also the experience shared by researchers in the areas of knowledge, such as computing, nutrition, music and pedagogy. Based on the game development process, we planned to use GDD (Game Design Document) and GOD (Game Overview Document) as the project artifacts, however some of the stakeholders had huge difficulty in handle them. The game development involving people of diverse background material led both to the creation of new artifacts and to the introduction of a changing in the management process. The combination of some good software engineering practices, such as eXtream Programming and Structured Analysis, and the association with new artifacts, hereafter form, allows us to estimate the time required to build the game and thus meet the schedule. The goal of this paper is introducing a newly developed game named Hortalândia and also describing a software engineering practice streamlined for teams composed by stakeholders of diverse areas of knowledge.

**Keywords:** Hortalândia, GDD, GOD, XP, game development process, metric.

### Author's contact:

filipe@comp.ufu.br  
cleverson@comp.ufu.br  
autran@facom.ufu.br  
frosi@facom.ufu.br

## 1. Introduction

Hortalândia is an educational game for kids regarding healthy nutritional habits which targets children aged between seven and eight years. The main purpose of the game is to teach children about healthy eating habits by stimulating the consumption of fruits and vegetables in opposition to the consumption of foods with high level of sugar, fat and salt. The demand for this game came to us as part of a nutritional research of the Medical School from our university. By the time we started to develop it, around one year ago, our team was composed by two nutritionists, two system engineer, two programmers and two designers. We took the nutritionist mates as our stakeholders since they were who actually have the game plot in mind.

However, they could neither stay side by side with us all week days during the game development nor we could be allocated full time to this project.

Therefore, we started our work by meeting the whole team once a week. Our nutritionist mates explained us the main idea of the game, essentially: five “bad guys” should be defeated by a game hero in five different phases. The bad guys represent the most prevalent bad eating habit that most of us have these days, especially the kids. The five bad guys are: Gordurosa, a fried potato pack; Puro Sal, a pack of salty snack; Gasoso, a bottle of soft drink; Açucarada, a cookie; and Adiposo, a cup of ice cream.

An important issue in that time was how to estimate the necessary time to finish the game. Would we be able to finish the game on Jun. 16 (our first due date)? There were many risks by then: (1) the nutritionists did not have detailed game plot in their minds; (2) we had no previous experience in game development; (3) we had no idea which software we should use in the game development so that it was free (GPL like) and had enough playability built in features. After a research about how to develop a game in the literature we realized that we should use Game Overview – GO and Game Design – GD documents to describe the game and since we were a small team with some experience using XP [Beck & Andres, 2004], we adopt this Agile Software Development [Beck & et.al, 2001] representative to guide us during the game development. Therefore, this work describes our difficult experience using GO and GD documents, what led us to create our own game document, and how this document could be used as User Stories so that we could estimate the necessary time to finish the game. This description is organized as follows. We start describing our difficult using GDD and GOD in Section 2; next we explain how we use XP to guide the game development in Section 3; following we briefly describe, in Section 4, Processing [Reas & Fry, 2010], the programming language of our choice to implement the game graphical functions; and finally we present some considerations about our experience under the light of related works in Section 5.

## 2. Hortalândia Game Documentation Form

In the game industry, the use of documents is necessary for better controlling of what the development team is meant to do. Traditionally these documents are filled in at different moments along game development. In the pre-production phase, the GOD (Game Overview Document) is used for general specifications of the main elements of the game. In the production phase, the GDD (Game Design Document) is used by designers to add all details present in the game. Of course, these two documents should be constantly updated during the development of the game as soon as new elements show up. Finally, after the game development completion, the Game Bible Document should be used to describe all elements present in the game. Maybe this whole process fits well in the game industry, although we get in doubts whether this kind of industry has enough time to be so disciplined. By our turn, we have experienced difficulty using the aforementioned documents, and then we decided to create a simpler one. In the next two subsections we describe GO, GD and Game Bible documents (Section 2.1), and our simple form (Section 2.2).

### 2.1. GOD, GDD and Game Bible

The GOD (Game Overview Document) is used by game designers to describe the concept of the game in a general way. It's used to make connection among the different sections of the development department. It is a living document where we integrate the project works. It has four sections which contents are explained below:

- Summary: describes the type of the game, the attractive elements that will make people want to buy (or use) it;
- Game elements: describes the gameplay of the game, how the player interacts with it, and the target audience;
- Story elements: synopsis, description of the story and who is the player;
- Interface: describes the perspective of the game, how the interface interacts with the player and the controls used to manipulate the character.

The GDD (Game Design Document) is used for describing, in details, what the game is supposed to look like, the gameplay, story and more detailed concepts. It is used primarily in pre-production containing the general ideas of the game. After its approval, the designer expands it to a level such that it can guide the development team to create the game. The main pitches of GDD are: story, interactivity and appendices for further information. GOD and GDD

must be updated during whole development of the game to assure the documentation consistency. Finally, the Game Bible is created after the conclusion of the game, in general. In this document, all game elements are described, in details, such as technical information, controls, interaction of the characters, conceptual art, hours spent on the game, and further detailed information.

By our turn, we have experienced difficulty using the aforementioned documents. Our stakeholders, who actually have to write the game plot, felt uncomfortable filling those documents due to the high amount of required details. As a result, their descriptions about the game usually lacked details, and then we spent much time trying to understand their text or in virtual meetings to clarify what had been written. Then we decided to create a less detailed form for game description so that was easier to fill in and could be used as User Story, one of the XP's milestones.

### 2.2 Hortalândia Game Form

Our stakeholder complained to us that they felt GO and GD document required many details that were unknown. And we felt a simpler document than the aforementioned ones so that we could use it as User Story. Then we decided to create a form especially for our needs. This document was inspired in our perception about how our game could be organized. We realized that our game could be divided in phases and each phase could be divided in scenes. A scene was the fine grain of our game. Then we created a document with two parts: Phase Document and Scene Document.

- Phase Document (PD) – shall be filled in with the number and the name of the phase, followed by the description of the phase objectives;
- Scene Document (SD) – shall be filled in with the scene number and the description of scene;

We explained to our stakeholders how to fill in the documents. Our expectation was that they could feel more comfortable with this document. We use PD to have whole picture of the phase (this document is important when our stakeholders are unreachable). We use SD to understand what should happen in the correspondent scene. Related to this document we were very specific in explaining that one scene should not exceed one sheet. It is a metric that we rescued from Structured Analysis (SA) [Gane & Sarson, 1977], [Demarco & Plauger, 1979] – a system development methodology from 70's. A good practice in this methodology was that a simple process should be able to be described in one sheet. A process in SA corresponds to an action that can change data in the system being modeled. A process can be compound by other processes. A simple process is the one that is primitive, in other words it is indivisible. Ultimately a simple process corresponds to an algorithm. If you

were not able to write a process in a single sheet then it was not a simple process. Similarly, if one scene could not be described in one A4 sheet, this scene was too big and should be split into two. And so the stakeholders did. And, at least for the developers of the game, it was better to distribute the SDs among the pair-programmers and estimate the demanded time for a scene, as we present next.

### 3. Hortalândia Game Creation Process

In the very beginning of the project we had weekly meetings with the whole team to understand what the game was about, and to capture the main project requirements. These requirements were supposed to be written by the nutritionists using GO and GD documents in the first division of the work. At the same time, the programmers were trying Processing to mitigate the development risks, and to get used the Processing architecture design. We needed to adopt a software development method to guide the game development, and then we chose XP [Beck & Andres, 2004] due to some previous experience with it by some of the authors. We adapted the XP to our needs basing on what could improve the development, so that the features we have been using are: iterations, frequent deploy of small releases of the software, pair programming, user stories, refactoring, and close stakeholder collaboration. The original XP life cycle is presented in Figure 1 [Wells, 2009].

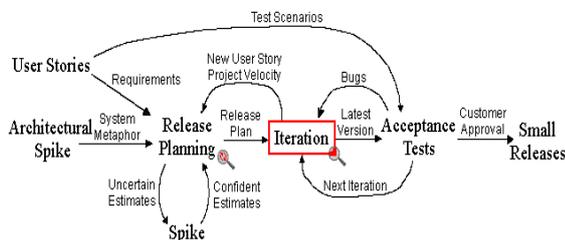


Figure 1: XP life cycle

In this adaption of the XP, the pair programming supported on the faster development of the scenes, turning possible the small releases of the game. Those releases would be shown to the stakeholders and then verify if there was something to alter. Therefore, the suggestions and tests there were done could give us a vision of the whole structure that Hortalândia would get.

After few months the programmers had already accumulated enough experience with Processing programming language. Programming errors were scarce and when a bug showed up they were able to fix it quickly. On the other hand, the nutritionist got used to describe the game using PD and SD, and they were used to deliver to us two or three scenes every month approximately. This scenario led us to realize the development pace of one scene: 10 days per pair-programmers working 20 hour per week. It is

important to mention that: (a) programmers and designers are undergraduate student of computer science working under scientific initiation (IC) project – that is why their dedication was partial time; (b) each pair of programmer was compounded by one senior programmer and one designer with programming skills. This scenario led us to hold two kinds of meetings: intellectual/acceptance and technical. The intellectual/acceptance meetings were held at each 15 days by the programmers, the system engineer A and the nutritionists (stakeholders). The purposes of this kind of meeting were twofold: to present a small release of the game and/or to discuss the content of the next iteration. One iteration usually consisted of the development of two new scenes and the eventual corrections in the previous scenes. The technical meeting was held by the programmers, the designers and system engineer B. The purposes of this kind of meeting were to present the ongoing development during an iteration product (small release), to discuss technical issues, such as how to correct some program mistakes, and plan how to attack the next iterations.

The process was very appropriate to our needs. Besides our metric time for developing a scene was fair enough. It allowed us to realize that considering our team and the remaining time to the due date we would not finish the originally five phases of our game. More important, it gave us tangible arguments to convince our stakeholders that we should prune some phases and scenes of our game, so that we could finish the game on time.

### 4. Processing Language

Processing [Reas & Fry, 2010] is an open source programming language and environment for people who want to create images, animations, and interactions. Initially developed to serve as a software sketchbook and to teach fundamentals of computer programming within a visual context, Processing also has evolved into a tool for generating finished professional work. Today, there are tens of thousands of students, artists, designers, researchers, and hobbyists who use Processing for learning, prototyping, and production.

In order to simplify user tasks, Processing provides user-friendly, intuitive and multi-platform, through which you can create and edit source code, download helper libraries and develop applications for both PC and mobile devices. Natively, Processing uses infinite loop, this concept is characteristic in games and animations. Two functions, or methods, are available for implementation by the user, they are: “setup” and “draw”.

The “setup” function implements the initial settings of the application, for example, defining a background, screen size, etc. It is the first function to be executed and is invoked only once. The “draw”

function is implemented throughout the application until it is closed, thus representing an infinite loop. The Figure 2 illustrates the flow of execution of an application according to Processing.

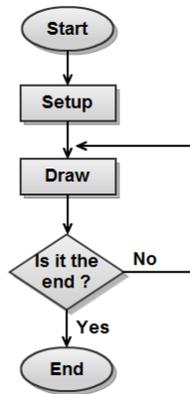


Figure 2: Execution flow according to Processing

We decided to use the prototyping capability that Processing offers to create the first animated draft of Hortalândia. The tools, methods and documentation provided by the language, easy integration with Java [Walsh & Fronckowiak, 1998] and, in consequence, the high portability, made us choose to use Processing to create the Hortalândia game.

## 5. Related Works

We have used Extreme Programming (XP) as development process for creating Hortalândia. However, other works present similar experience using Scrum [Gomes, et al., 2011] [Peres & et.alli, 2011]. Our choice for using XP was just due to previous experience of some of authors.

Considering the Hortalândia's form, it was the mean that our stakeholders used to express their needs relating to the game. We used the content of the forms as User Stories, similarly to [Schetinger, et al., 2011].

## 6. Final Considerations

We finish the game on time. It was not the initial game that our stakeholder had planned for. Initially they had in mind a game with five phases and we delivered a game with three phases. We could show them that with the quantity of people we had involved in the project we could not finish the game in the due date (Jun. 16). We could convince them because we had in hands a sound metric rescued from Structured Analysis. This metric signaled us that one pair of programmers of our team spent 10 days to deliver one scene of the game. Since we knew how much scenes left to be developed we could show to our stakeholders that there were too few calendar day to accommodate a five phase game.

## Acknowledgements

Our thanks to: Ana Elisa Madalena Rinaldi and Priscila Nunes Carvalho, our nutritionist mates, for the game plot and their supervising task; Danilo Bizinotto Borges for his collaboration in some drawings; Danilo Silva Aguiar for the sound track and sound effects; and Raphael de Souza Pimenta for his contribution in part of the game code. Cleverson de Lima Puche also thanks CAPES / MEC for the scholarship granted by the Tutorial Education Program - PET.

## References

- BECK, K. & ANDRES, C., 2004. *Extreme Programming Explained*. 2ª ed. Massachusetts: Addison-Wesley.
- BECK, K. & et.alli, 2001. *Manifesto for Agile Software Development*. [Online] Available at: [agilemanifesto.org](http://agilemanifesto.org) [Accessed 12 Jul. 2012].
- REAS, C. & FRY, B., 2010. *Getting Started with Processing*. 1st ed. Sebastopol: O'Reilly.
- GANE, C. & SARSON, T., 1977. *Structured Systems Analysis: Tools and Techniques*. 1 ed. s.l.:McDonnell Douglas Infomation.
- DEMARCO, T. & PLAUGER, P. J., 1979. *Structured Analysis and System Specification*. 1 ed. s.l.:Prentice Hall.
- WELLS, D., 2009. *Extreme Programming Project*. [Online] Available at: <http://www.extremeprogramming.org/map/project.html> [Accessed 16 Jul. 2012].
- WALSH, A. E. & FRONCKWIAK, J., 1998. *Java Bible*. s.l.:John Wiley & Sons.
- GOMES, J. E. et al., 2011. *Adoção de Metodologias Ágeis para Produção de Jogos Sociais com Times Distribuídos*. In: SBGames 2011. Salvador, s.n.
- PERES, A. & et.alli, 2011. *Methods and Processes Definitions for Multiplatform Social Network Games Development with Distributed Teams*. In: Anais SBGames 2011. Salvador, s.n.
- SCHETINGER, V., SOUZA, C., FONTOURA, L. M. & POZZER, C. T., 2011. *User stories as actives for game development*. In: Anais SBGames 2011. Salvador, s.n.