

Chapter 2

THEORETICAL FRAMEWORK

2.1 Introduction

Creating an android game is one way of exercising the innovation of technology. It illustrates the effectiveness and sense of information technology in the world of ICT. Since game development is an essential growing market to mobile technology, games are limitless. Ideas and visions are converted into real games, not only story line or plot wise but also with the help of extra ordinary graphics. A game can be played anytime and anywhere. However, it is uneasy to develop an android game because it requires disciplined and skilled people regarding the development and designs. Importantly, developers must be able to know the theories that are best suited to make game imaginations come true. It will not only inform them about the game development strategies but will also guide them upon the reasonable practice and actions for the benefit of the project.

The proponents adapted the IT and Non-IT Concepts that would be helpful for the development of the Android game, HULA-WHO? (Inventions and Discoveries from Past to Present).

2.2 Non-IT Concept/Theory

Game Theory

Game theory is a theory of social interaction, which attempts to explain the interaction people have with one another.

Game theory was originally an economic and mathematical theory that predicted that human interaction had the characteristics of a game, including strategies, winners and losers, rewards and punishment, and profits and cost. It was initially developed to understand a large variety of economic behaviors, including behavior of firms, markets, and consumers. The use of the game theory has since expanded in the social sciences and has been applied to political, sociological, and psychological behaviors as well.

Game theory was first used to describe and model how human populations behave. Some scholars believe that they can actually predict how actual human populations will behave when confronted with situations analogous to the game being studied. This particular view of game theory has been criticized because the assumptions made by the game theorists are often violated. For example, they assume that players always act in a way to directly maximize their wins, when in reality this is not always true. Altruistic and philanthropic behavior would not fit this model.

Examples of Game Theory

We can use the interaction of asking someone out for a date as a simple example of game theory and how there are game-like aspects involved. If you are asking someone out on a date, you will probably have some kind of strategy to “win” (having the other person agree to go out with you) and “get rewarded” (have a good time) at a minimal “cost” to you (you don’t want to spend a large amount of money on the date or do not want to have an unpleasant interaction on the date).

Elements of a Game

There are three main elements of a game:

- The players.
- The strategies of each player.
- The consequences (payoffs) for each player for every possible profile of strategy choices of all players.

Types of Games

There are several different kinds of games that are studied using game theory:

- ***Zero-sum game:*** The players' interests are in direct conflict with one another. For example, in football, one team wins and the other team loses. If a win equals +1 and a loss equals -1, the sum is zero.
- ***Non-zero sum game:*** The players' interests are not always in direct conflict, so that there are opportunities for both to gain. For example, when both players choose "don't confess" in Prisoner's Dilemma (see below).
- ***Simultaneous move games:*** Players choose actions simultaneously. For example, in the Prisoner's Dilemma (see below), each player must anticipate what their opponent is doing at that moment, recognizing that the opponent is doing the same.
- ***Sequential move games:*** Players choose their actions in a particular sequence. For example, in chess or in bargaining/negotiating situations, the player must look ahead in order to know what action to choose now.
- ***One-shot games:*** The play of the game occurs only once. Here, the players are likely to not know much about each other. For example, tipping a waiter on your vacation.
- ***Repeated games:*** The play of the game is repeated with the same players.

Prisoner's Dilemma

The prisoner's dilemma is one of the most popular games studied in game theory that has been portrayed in countless movies and crime television shows. The prisoner's

dilemma shows why two individuals might not agree, even if it appears that it is best to agree. In this scenario, two partners in crime are separated into separate rooms at the police station and given a similar deal. If one testifies against his partner and the partner stays quiet, the betrayer goes free and the partner receives the full sentence (ex: ten years). If both remain silent, both are sentences for a short time in jail (ex: one year) or for a minor charge. If each testifies against the other, each receives a moderate sentence (ex: three years). Each prisoner must choose to either betray or remain silent, and the decision of each is kept from the other.

The prisoner's dilemma can be applied to many other social situations, too, from political science to law to psychology to advertising. Take, for example, the issue of women wearing make-up. Each day across America, several million woman-hours are devoted to an activity with questionable benefit for society. Foregoing makeup would free up fifteen to thirty minutes for each woman every morning. However, if no one wore makeup, there would be great temptation for any one woman to gain an advantage over others by breaking the norm and using mascara, blush, and concealer to hide imperfections and enhance her natural beauty. Once a critical mass wears makeup, the average facade of female beauty is artificially made greater. Not wearing makeup means foregoing the artificial enhancement to beauty. Your beauty relative to what is perceived as average would decrease. Most women therefore wear makeup and what we end up with is a situation that is not ideal for the whole or for the individuals, but is based on rational choices by each individual.

Assumptions Game Theories Make

- The payoffs are known and fixed.
- All players behave rationally.
- The rules of the game are common knowledge.

[UNKN2013]

2.3 IT Concepts/Theories

Theory 1

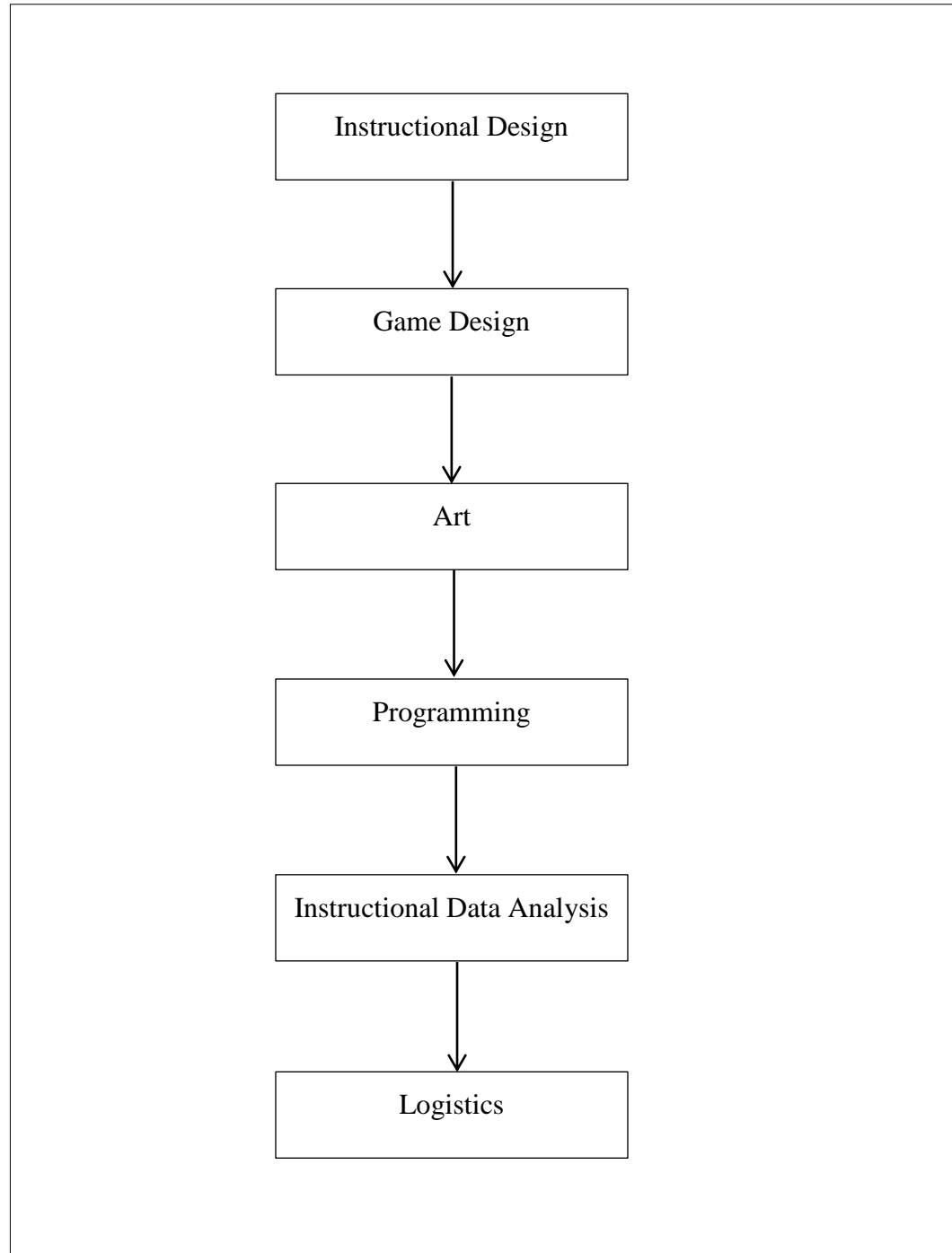


Figure 1: Game Design Process

Game Design process for an educational game by Purdue University which includes instructional design, game design, art, programming, data-analysis, and logistics sections. This design process emerges out of the work done by the subject matter expert, instructional designer, and the game designer for the team. The lead programmer and lead artists also contribute during the design phase by using their expertise to chart out how the team should tackle the technical portions of the game.

Instructional Design

- **Audience Analysis:** An analysis of the learner's current skills and how those skills map to the instructional content
- **Entry Behaviors:** Identification of the learning objectives the player must have mastered prior to playing the game in order to be successful. Any skills or knowledge identified as an entry behavior will not be covered in the game. It is a pre-requisite for the game.
- **Instructional Goals:** Broad educational goals for the game.
- **Instructional Objectives:** Performance objectives for the game. It is very critical that instructional objectives are granular enough to allow for the diagnosing of instructional problems. At a minimum objective should address (1) behavior to be measured, (2) conditions under which the behavior will be measured, and (3) a minimum level of achievement needed to master the objectives.
- **Assessment Items:** Assessment items for each of the learning objectives should be created. It is important that each assessment item only tests single instructional objectives. In the case of an educational game the assessment items are often constructed as in-game activities that map to specific learning objectives.
- **Presentation Strategy:** With the audience analysis, learning objectives, and in-game learning assessments prepared, we can then concentrate on how to present the necessary instructional materials to the learner in a manner that will prepare them to successfully complete the assessments.

In the case of most educational games, the presentation strategy will drive many of the gameplay decisions about the game. The game-mechanics and game narrative will reflect the presentation strategy.

Game Design

Game Concept:

- **Game Description:** This is a very broad description of the game.
- **Genre:** What kind of game is it?
- **Platform:** Will this run on a PC, console, PDA, phone, etc.?

Game Mechanics:

- **Core Gameplay:** What actions will be available to the player consistently and how will those actions influence the world?
- **Mode of play:** How many different modes of play will be available in the game?
- **Game Flow:** How will the player progress through the game be organized. Is the game broken into levels? What triggers the end of a level? How will in-game assessments be integrated into the flow of the game?
- **Types of Characters:** How many different types of characters are there? How do they behave differently in the game?
- **Gameplay Elements:** What environmental elements exist in the game that add to the game play? Will there be items that act as a power-up? etc. Are there different types of weapons?

User Interface Functionality: What are the user interface items and what functionality is needed for each of the items. This includes screens and menus.

Narrative: The back-story for the game and characters

Art

User Interface: What do the screen elements and menus actually look like? This should include the color scheme, resolution, fonts, etc.

Gameplay Elements: What do the game elements look like? This includes sketches of the characters and the setting for the game.

Sound and Music: Identifies any needed music and sound effects

Programming

Special Technical Requirements: Details any technical requirements that are beyond the norm for a game. If the game will rely heavily on networking or use a special type of rendering technique, it should be noted here.

Game Engine: Will a particular game engine be used?

Rendering: How will rendering and light be handled?

Artificial Intelligence: How will AI be handled?

Physics: How will physics be handled?

Gameplay code: How will gameplay specific code be handled?

Instructional data tracking and analysis: Are there any data tracking requirements for the instructional aspects of the game? How will that be done? Do we need to interface the game to a database? Do we need to send performance data via the network in real-time or can we batch it after the gameplay session?

Instructional Data Analysis

Data analysis model: How will the data generated during the gameplay session be analyzed?

Logistics

Human Resources: How many people will contribute to the game and in what capacity?

Schedule: How much time has been allotted to each task? [PURD2007]

2.4 Theory 2

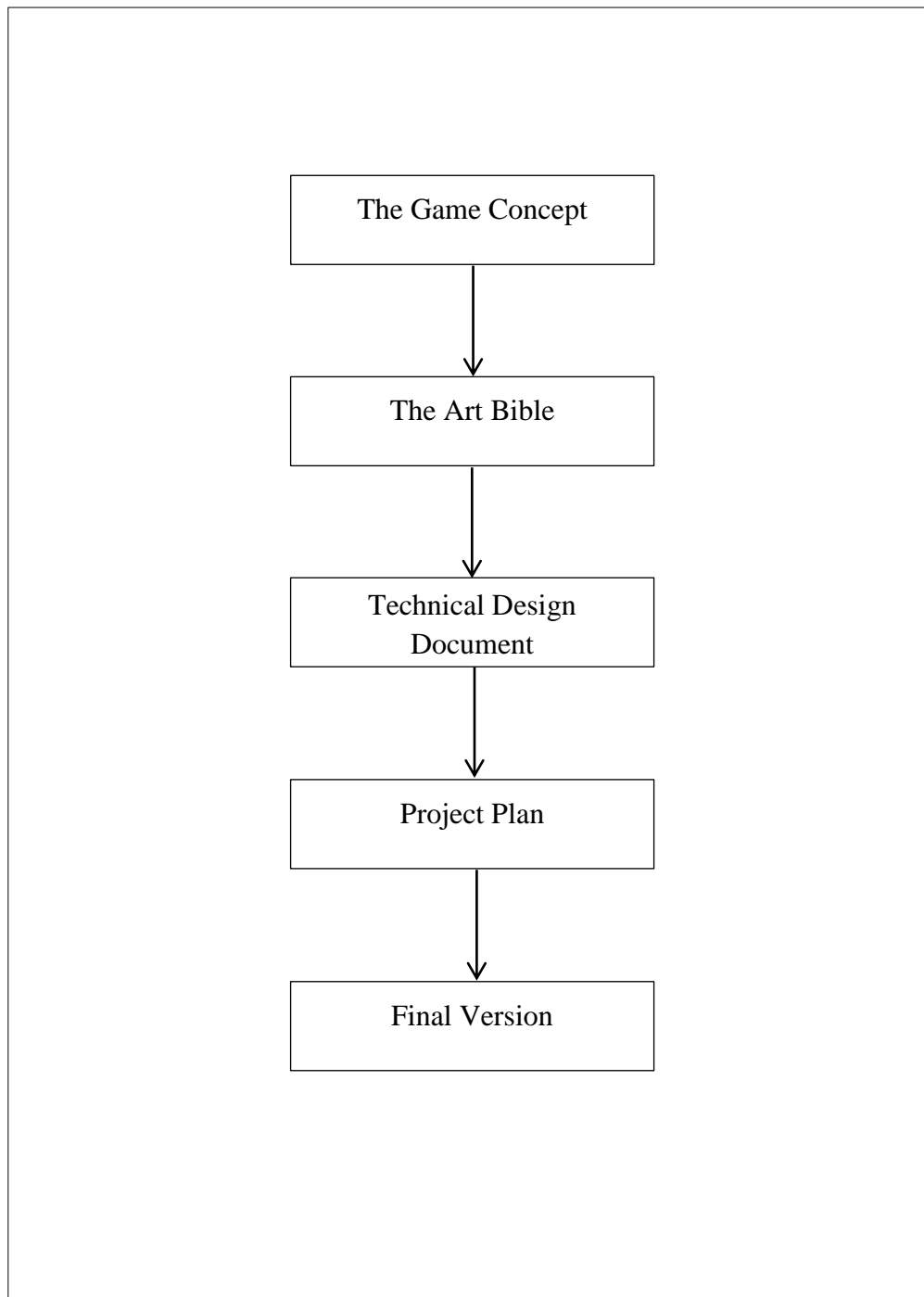


Figure 2: Game Development Process

The Game Development Process

The game development process usually begins with a game concept. Among hobbyists and small teams, the concept may be formed in the head or written on a napkin, but in the more serious studios it is written down carefully. This document guides the prototyping phase, low-fidelity paper versions of the game, or very simple digital versions of the game are developed, tested, and explored. If the concept looks promising, more documents are created, such as the art bible the production plan, and eventually lead to the creation of the game design document as well as the technical design document.

At this point, production is begun, a stage that can last from several weeks to a few years. An alpha version of the game is finally produced at which point the game is playable from beginning to end. This is tested and de-bugged until the beta version of the game is developed. The beta version is similar to the alpha version, but should include all final art assets, and be stable enough that a selection of customers can play the game and report problems. The final stage is sometimes called the gold stage, where the game has gone through complete testing and has passed a final review.

The Game Concept

What is the goal or purpose of the game? What is the premise? Is it a tetris-like game where the player arranges geometric forms while they fall? A two-person shooter that takes place under water? A learning-game where the player "becomes" each animal they are studying? The concept should include a short treatment on player motivation, as well as some discussion as to what makes the game unique. This unique factor is sometimes called the Unique Selling Proposition. What are the unique features to the game that would make someone want to buy it? Also key in the game concept document is the genre - how does this game fit within the existing game ranks? Is it an action game, an adventure game, a sports game, a simulation? Some thought should also go into the rating you hope the game will achieve, and what platform the game will be played on (Mac, PC, console, iPhone, etc.). Another very important consideration is the competitive analysis; several current, successful titles should be reviewed and compared to the proposed game, including why the proposed game will compete favorably.

Finally, what are the goals of the game? Are you hoping to make the player feel tension, happiness, creative freedom, intellectual achievement?

The Art Bible

This is a collection of sources, sketches, storyboards, and digital resources to be used in the creation of the game. The art bible keeps the art team focused on how the characters, terrain, environment, pickups, interface, lighting, etc, should look, so that the game has a consistency between artists, scenes, and levels.

Technical Design Document

This document focuses on the game engine, what features it will have, how it will be leveraged to move from concept to finished game. If the game engine is not to be developed from scratch, this document can include an overview of the features to be used in the engine, and any custom features that may need to be added in order to support the game's goal.

Project Plan

This document plans the stages of the development of the game in the form of a schedule, taking into account all of the necessary game design, prototyping, creation of documents and other planning, and goes through actual production of alpha, beta, and gold versions of the game. Here the man-hours to create the game are calculated, as well as all other costs including costs for the software and any other expenses with operating the company.

Step 1: Decide what kind of game you want to make

Do you want to make a 2D game or a 3D game? Adventure game, sports game, or simulation? Should the game be playable on a PC, Mac, Linux, console, iPhone, or cell phone? Is it a single-player game or multiplayer? Will you utilize First Person perspective, Third Person perspective, "god" perspective? See the Game Genres page for a breakdown of the different types of games you could develop.

Step 2: Consider your skills, tools, budget, and resources

Now that you have an idea of what kind of game you want to create, take some time and do a reality check; how well do you and your resources measure up to the challenge you have set for yourself? Are you skilled with creating art assets such as a game environment, pickups, skyboxes, characters? Can you animate? Can you program or script a game engine? Do you know anyone who can help? Do you have the budget to hire help? Do you own or can you afford the necessary hardware and software to build the game? Have you decided on a game engine? Get an idea of the different options for Art Asset creation by clicking the [Game Dev Art Assets](#) page. Get an overview of the different Game Engines out there by clicking the [Game Dev Game Engines](#) page.

Step 3: Revise Plan based upon Reality Check

With your skills and resources clearly in mind, and having read through the sections on Art Asset creation, game engine selection, programming, and game genres, revise your plan so that it and reality coincide. The other aspect to this step is to simplify your plan so that it is doable in a few weeks or months. It is very common in the world of game development to have young, inexperienced artists and programmers with huge, impressive ideas but no real means of getting the game accomplished. Start small and simple, finish something, and build from there. Create a basic game design document that establishes some framework for how the game should play, what the rules are, etc.

Step 4: Make a Paper or Low-Tech Prototype

Take your game idea and test it out as if it were a board game. Work through the game-play with your development team or some friends. If it's not interesting as a board game, it probably won't be interesting as a video game. This is not always possible, of course; a game like asteroids is not easy to simulate as a board game, so in that case you might need a digital prototype that is very simple but generates the "feel" of the gameplay. The bottom line here is to test your game idea out before investing too much time and money into it.

Step 5: Acquire the Hardware and Software

When you feel sure of your objective and have a good idea of the hardware and software options, it is time to buy some equipment. If you don't have much money, you will at minimum need a computer; almost all of the software you will need to create a game can be had for free, but the tradeoff is that you may not be using software that is the best for the job, and it may be hard to find team members who are knowledgeable about that software. Also, if you are hoping to leverage your game development experience into a job, you may want to use, as much as possible, the same software the game studios are using. Please Note: If you are not developing a commercial game, you may be able to use educational versions of software to develop your game. See our [Developing Game Art](#) and [Choosing a Game Engine](#) pages for more information.

Step 6: Prototype High-Risk Features

Now that you've developed a realistic plan, are convinced that the gameplay is compelling, have acquired the hardware and software you will need, and have acquired the necessary team, you are ready to see if you can actually develop the features you will need for the game.

In any game development, there are going to be those features that are high-risk; perhaps you need to be able to incorporate a special kind of collision detection, or you want your character to be able to fly around in the game environment. What happens if you spend three months developing your game and then discover in-game flight is impossible in the game engine you have selected, or you find that you don't have the coding talent to pull it off properly? Well then you may have just wasted three months of work. Without in-game flight, your levels will need to be redesigned for some kind of land travel, and many of your game mechanics will have to change as well. With that in mind, it makes sense to create rough prototypes for every high-risk feature in your game before you begin full-blown production. It is important to know where you stand with the high-risk features, and to have the ability to accurately estimate how long each phase of development will take.

Step 7: Establish the Game Art Pipeline

Getting the art into the game can be straightforward or mindboggling, depending on the game engine you decide to use, and the 3D modeling and animation software you are developing it in. It is critical at this stage to truly understand exactly how you will get all of the game art, including terrains, skyboxes, pickups, and especially rigged characters, into the game. Develop a system that works, and document it so that everyone necessary knows how it is done, to avoid confusion and time loss down the road.

Step 8: Develop or Acquire the Game Art

The game art can be a massive part of any game development effort. While we go into the process of developing game art at GameDev: Creating Game Art, there is also the option of acquiring the game art. Sites like Turbosquid regularly sell entire 3D cities, fully rigged 3D models, fully animated models, perhaps everything you need, potentially saving you an enormous amount of time. However, depending on how the game art was rigged and animated, you may still need to make changes in order for that art to come successfully into your game engine. Before investing in too much third-party game art, test to see how it is going to work out in the real world.

Step 9: Develop the Alpha Version

The next reasonable phase would be to create a working version of the game, where everything of importance works, though there may still be some rough edges and bugs. At this phase the game art can still be simplistic or borrowed from stock characters, pickups, and landscapes. This purpose of the Alpha version is to get the game out there in a playable format for testing. This will be the first time the game as a whole has been playable, from start to finish. This is a time not only for debugging, but to correct any last minute issues with design, playability, and interface.

Step 10: Develop the Beta Version

The beta version is the Alpha version, debugged, with finished game art. This version should be good enough for select members of the general public to play. There

will still be bugs in this version, but they should not be so numerous that it is a frustration to test the game out.

Final Version

This version of the game is ready for distribution; it has been tested by numerous end-users, on all intended platforms, in as many situations as reasonably possible. Any significant bug fixes beyond this point can be distributed in either bug patches, or incorporated in subsequent releases.

[UNKN2013]

2.5 Summary

In the development of the game entitled HULA-WHO? (Inventions and Discoveries from Past to Present) the proponents will adapt the Game Theory for Non-IT Concept.

Furthermore, for IT Concepts, the Theory for Game Design Process by Purdue University and Game Developments Process will also be used in the development of the game. Game Design Process involves six steps which starts upon instructional designing and ends up upon the logistics. While Game Development Process is composed of five steps starting from game concept up to the final version.

This Non-IT and IT Concepts/Theories is greatly important upon developing of the entire game. By the help of these theories, the proponents will be able to systematically organize all the activity to finish the work on time.