

# GAMIFICATION AND GAMING TECHNOLOGIES

VANKA SAI SUMANTH & AKSHAY YADAV

24 march, 2017

## Abstract

This paper discuss about the gamification, importance of artificial intelligence in gaming technology and about programming languages for artificial intelligence, an overview of the background of gamification and its applications, the relevant key game concepts, various game design elements, gamification and traditional game methods and programing languages constructs used in non gaming context.

## 1 Introduction

Gamification is the use of game design elements like game thinking and game mechanics in non game contexts to solve problems. Gamification applications helps people in all sectors to engage in areas such as marketing, human resources and education, for personal development or innovation purposes. One of the essential game elements that can be used to support and stimulate game like characteristics is game mechanics. Like when we are using Hackerrank, codechef etc, we get credits such as points, badges and leaderboards. To understand and implement Gamification in multiple fields and to develop present gaming technology, we first need to look into gaming constructs, game programming and available technologies.

## 2 Game Programming

Game programming is a part of game development. Game programming is the software development of video games. It needs high skill and specialization in one or more of the following areas, which helps people to create a game: *computer graphics, simulation, **artificial intelligence**, network programming, database programming, physics and audio programming*. Among these, Artificial Intelligence(A.I.) is the heart of game design and Gamification. Here, we would see which programming language constructs are useful for A.I.

### 2.1 Artificial intelligence

When we play a game, we need to have enemies to combat. We want these enemies to seem intelligent and challenging to keep the game fun and engaging. When making a game, we can do this through different type of techniques collectively known as artificial intelligence (AI) programming.

**Artificial intelligence** is the science and engineering of making intelligent machines,

especially intelligent computer programs. It is similar to using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable.

## 2.2 Programming languages for A.I.

What is the best programming language for AI?

The one you are comfortable with! There are a few exceptions though, e.g. if you want to artificially evolve a program, it will make your task easier to write it in an interpreted language.

There is no exact answer for this question, as it really depends on what languages you like programming in and comfortable with. Programs for Artificial intelligence is written in most of the languages till now. The most languages are Lisp, Prolog, C/C++, recently Java, and Python.

### LISP:

One of the reasons why AI has favored high-level languages such as Lisp is its fast prototyping was favored over fast execution. The language features which are good for AI programming include: *functions as data, dynamic typing, interactive environment, uniform syntax, garbage collection and extensibility.*

### PROLOG:

Because of this language programmers started realizing that a set of logical statements and a general theorem prover could make programs. This language has an unifier which has traditional advantages of lisp which is useful of artificial intelligence programming. Prolog is good for problems in which clear logic is involved, or whose solutions have a brief and clear logical characterization. Prolog is hard to learn which is a drawback.

### C/C++:

C/C++ is mostly used when the program is simple, and execution speed is important. The common examples are Statistical AI techniques such as neural networks etc . Back-propagation is only a couple of pages of C/C++ code, and needs most of speed that the programmer can muster.

### Java:

Java uses many ideas from Lisp, of which garbage collection is a notable one. Java is known for its portability which is used for many A.I. applications. It has a good set of built-in-types. Java is not high-level language like Lisp and Prolog. It is less faster when compared to C or C++. But the best part of Java comes with its feature of portability.

### Python:

Python has better libraries when compared to others. But this language is not accepted or used widely till now. But research says it might pass over java very soon.

## 3 Current Tools and Techniques

A major evolution in game development technologies has occurred since its early days. Starting from assembly language, many tools and techniques evolved. This section describes the most used game development technologies and explains why they do not yet completely fulfill industrialization needs.

### 3.1 Multimedia APIs

Multimedia APIs (Application Program Interfaces), are programming libraries that can be used to directly access the machine hardware. APIs such as Microsoft DirectX and OpenGL, are not only useful for providing means to create games with good performance, but also for enabling the portability of computer games among devices manufactured by different vendors. Therefore, if we use Multimedia API, then there is no need to worry about low-level peculiarities of each possible target device.

Multimedia APIs set a new stage in game development, by empowering programmers with more abstraction to experience an easier game development process. These APIs are heavily used today and certainly will last for a very long time, being used either directly or indirectly. These libraries handle almost all the desired low-level functions, but still game itself has to be programmed. It provide features that are generic for computer games development and do not offer the abstraction level desired by game programmers. The semantic gap between game designers and the final code remains too high if multimedia APIs are the only abstraction mechanism used. Interaction with these APIs can only be done programmatically, not visually. This approach may prevent automation and productivity in the execution of some tasks.

### 3.2 Visual Game Creation Tools

Visual game creation tools were created with the intention to simplify game development and make it more accessible to a broader range of communities, and soon became very popular. They aim was to complete games by just clicking with the mouse not with programming at all. The end user is aided with graphical and easy-to-use interfaces for creating defining entity behavior, game animations, the flow of the entire game and to add menus, sound, text screens and other resources to the game.

These tools try to address such a problem by offering to users script languages, targeted at allowing more complex behaviors to be specified. This require users to have some programming skills and to learn a new language. This is not the original purpose of such tools. Instead of working with error-prone scripting languages inside an environment which was not originally conceived for codification users might prefer to have the benefits of true object-oriented programming languages, with the support of robust integrated development environments with full editor and debugging support.

### 3.3 Game Engine

Game engines were result of applying Software Engineering concepts to computer games development. An engine can be seen as a reusable API, which gathers common game development foundations and provides to developers a programmatic interface through which game behavior can be specified. The basic game functionalities provided by game engines are built on top of multimedia APIs.

Game engines are the tools in computer games development. By providing more knowledge encapsulation, abstraction, and a reusable game development foundation, they allowed the game industry to reach an unparalleled productivity level. But still it has some drawbacks. Main among them are complexity and costs.

## 4 GAMEPLAY

### 4.1 What is "Gameplay ?"

All designers and every game player talk about gameplay which is evidence of the importance of such a concept. There is probably no universally definition of gameplay. However the players focus on gameplay as a key element to determine the quality of a digital game. So what is gameplay according to players ?

A player-centered approach can lead to define gameplay as the set of activities that can be performed by the player during the ludic experience, and by other entities belonging to the virtual world.

### 4.2 Game Mechanics

“Interactivity” and “activity” are two concepts underlying the definition of gameplay. Players have to interact with abstract objects and their manipulation requires a level of proficiency achieved through a learning process. The use of the abstract objects and their relationships are regulated by rules, which organize a set of ludic activities and turns it into a complete and coherent game. Some of these abstract objects are called “game mechanics” in the game design.

Players do talk about game mechanics. In fact, right after commenting the overall quality of the gameplay, players tend to focus on the elements they have to deal with in order for things to happen in the virtual world. A player-centered perspective can lead to defining game mechanics as a proper tools for gameplay, atomic rule-based is capable of receiving an input and reacting by producing an output. Such output translates into a state change of the mechanics itself or into the triggering of new interactions with other game mechanics. A given game mechanics might be capable of receiving different inputs and reacting consequently.

## 5 How Gamification Platform helps user to learn Programming ?

Gamification platform provides a reach stuff to learn programming and they also provides a interface to start programming. They start from basic and take you to deep inside the programming.

### 5.1 Codecademy

This is the most popular gamification learning platform on the internet. It's totally free of cost and it covers HTML/CSS, Python, Ruby, Javascript and API's. It consists 100 of lessons, each one of them contains 5 to 20 lessons. It teaches both complex and

complicated stuff, such as web and game design. It has a system of badges and achievements – the more lessons you complete, the more badges and points you get.

## 5.2 Codeschool

The main thought is that “the best way to learn is by doing”. It covers Ruby, Javascript, HTML/CSS, iOS and Git. The main difference from Codecademy is that the fact that it combines video content, coding in the browser. Similarly to Codecademy it rewards with badges, points and achievements.

## 5.3 Treehouse

Apart from badges, achievements and video, Treehouse includes quizzes. It includes Android, C#, CSS, Databases, Design, Development, Digital Literacy, Game Development, HTML, iOS, JavaScript, PHP, Flask, Swift, Python, Ruby, WordPress, Java, Web Design, PHP, WordPress, workshops, courses. You can earn points and badges by watching videos and solving quizzes.

## 5.4 CodinGame

CodinGame uses actual games to help in learning process. It covers C, C++, Java, C#, PHP, Python, Ruby, Javascript, Objective-C, Go, Haskell, Scala, Perl, Dart, Bash, Pascal, Groovy, Clojure, VB.NET, Lua, F#, OCaml, Swift, Rust. Everything happens in real time, in a browser where a person codes and sees the game in the corner of the screen.

## 5.5 CodeCombat

Similarly to CodinGame, Codewars also uses actual games to make people learn coding. The screen is divided between a code editor and a labyrinth with a player’s avatar that can be controlled by a set of commands that have to be typed in correctly, because each error in code decreases hit points. Each level consists of different quests, such as collecting gems, defeating monsters and finding the way to the next level.

There are more gamification platforms which have their own features but they all are working almost the same. Idea of gamification can be considered as a novelty not only in programming, but also in other areas of life. Fortunately, there are more and more people who appreciate the value of gamification, also in IT. It’s a great way to boost motivation and productivity, that’s why new gamification platforms pop up like mushrooms!

# 6 How is Gamification Changing our Lives ?

If you walk through public area then you’ll find that people are using their phones, playing mobile games etc. This is the reason for game industry is among the least appreciated business phenomena in the world. Gamification has two primary mechanisms. The first is to take design structures from games, such as achievements, points, levels and leaderboards. The second is to design techniques that designers have developed over many years. Games pull you in and carry you through a journey that remains engaging and using an evolving balance of challenges. Many enterprises use tools built on top of customer relationship management platform to motivate employees through competitions,

points and leaderboards. Gamifications works because our responses to games are wired into our psychology. It activate our innate designs to recognize patterns, master challenges, communication with others and recognize patterns. They can create a safe space for learning and experimentation. Gamification motivates through rewards: Earn some points, win a prize, top the leaderboard and repeat. These motivations helps us to learn and this motivation has changed our lives.

## 7 Applying the Framework to Serious Games

Serious games have been used before as motivators for learning and for healthy, sportive activity. The sensor technology and the psychophysiological methods provide great possibilities in such application scenarios, both (1) to enhance adaptation and personalization of existing games and systems during play and (2) to evaluate the effectiveness of serious games in general.

The individual player experience is important in serious gaming, since this frame is where an effect of the serious game on behavioral change in the player can be witnessed. It depends on the kind of serious game that should be played,

## 8 Future of Gamification

Serious gaming simulations and game mechanics such as leaderboards, skill-based learning , performance, achievements, and engagement. Gamification mechanics is being employed nowadays in education, training, marketing, and wellness initiatives. Game elements are interspersed throughout the platforms that have made social networks popular. Businesses, and other organizations have come to depend upon the competitive metrics. Gaming functionality will continue to grow and be used in more and more facets of our lives. People will receive training on development programs, be exposed through education and jobs, have the ability to learn about areas that are important to them using this technology . It will allow people to understand complex topics easier, faster, and make the learning process more anticipated and less to be feared or avoided and it will be the most important social and commercial development of the next fifty years and will touch all aspects of the organization as it is applied to management, sales, production, and other areas of commercial practice. Gamification will enable people to gamify many aspects of life and work, in digital, physical, and blended environments.

## References

Read Paul Graham's essay, "Beating the Averages" <http://www.paulgraham.com/avg.html>  
<https://en.wikipedia.org/wiki/Gamification>  
<https://www.cs.auckland.ac.nz/courses/compsci747s2c/lectures/paul/definition-deterding.pdf>  
[http://hubscher.org/roland/courses/hf765/readings/Groh\\_2012.pdf](http://hubscher.org/roland/courses/hf765/readings/Groh_2012.pdf)  
<http://www.paulgraham.com/avg.html>