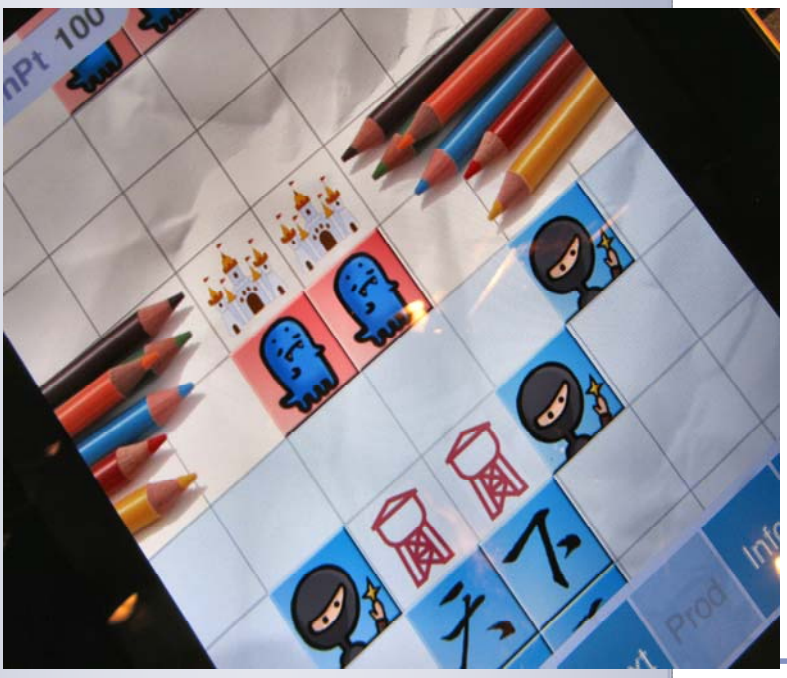


# Challenges and Opportunities of Mobile Game Development

By Simon S.H. LUI

In the 1990s, most computer games were designed for family desktop computers. Nowadays, you will probably see people playing the games A*ngry Birds* or *Where's My Water* on the bus with their iPhones or Android phones – and this shift has only been happening for a few short years. The world of gaming is changing tremendously. People are playing games on the go with smart-phones. For game developers, it is not as simple as "writing games for smaller consoles" --- many challenges as well as opportunities are surfacing.

Let's talk about challenges first. Mobile games need to run on devices with limited resources, including CPU, memory space, storage space, etc. In order to have a better understanding about these resources, imagine you are writing a proposal in your office. You are the CPU, performing thinking and writing tasks. With a higher CPU processing speed, you can finish a single task in shorter time. Your proposal booklet refers to the storage space, which is for saving data. Your work desk is the memory space. With a larger work desk, you have more space to process data and hence to finish your work more efficiently.

Now for a real example. The latest iPhone 4S only has 0.5GB memory and a 0.8GHz 2-core CPU. A standard desktop computer usually has 4GB memory and 4-core 2.0GHz CPU. There is a huge difference between the two devices. However, the game *Street Fighter* is available on both the iPhone and the Windows computer platform. Users will fully expect similar graphics quality and performance on both versions. In this case, programmers have to handle resources very carefully -- especially for the iPhone version. To deal with high-resolution graphics, the system needs to allocate memory space effectively. Programmers need to keep the memory as clean as possible. For example, only necessary graphic data are allocated with memory space. Once they are not needed, they will be released from memory immediately. In order to make the iPhone version with less computation power as responsive as the Windows computer version, programmers need to reduce the burden on the CPU. This can be achieved by using computational effective algorithms such as *divide and conquer* and the *Dijkstra algorithm* for mathematical calculation.


Screenshot of the game soundmitate. Player imitates a given sound and the program measures the accuracy of the user's imitation attempt.

A divide and conquer algorithm works by recursively breaking down a problem into two or more sub-problems of the same (or related) type, until these become simple enough to be solved directly. The solutions to the sub-problems are then combined to give a solution to the original problem.

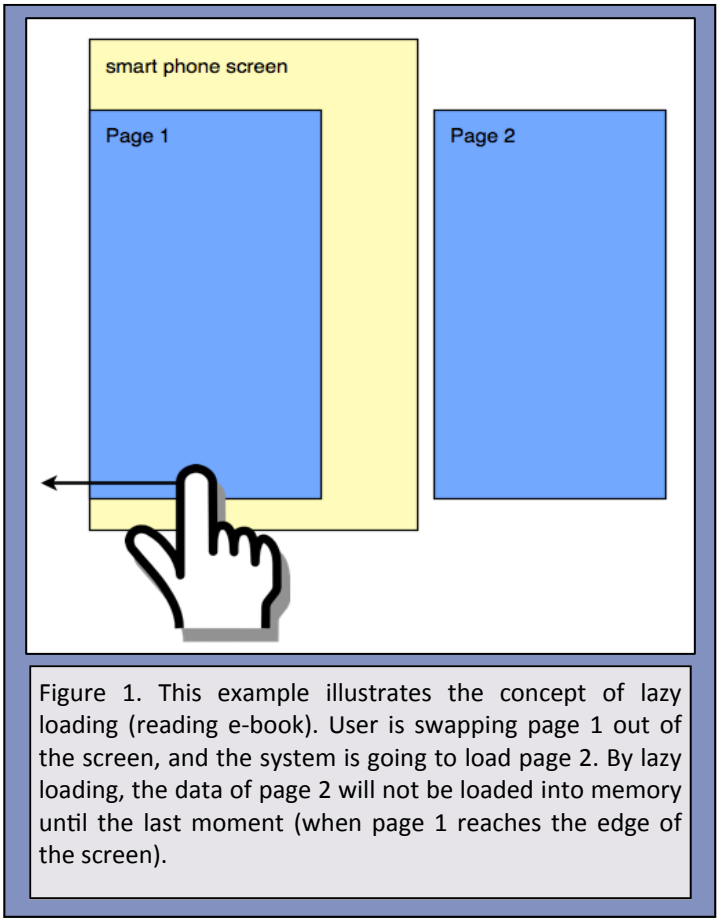



Figure 1. This example illustrates the concept of lazy loading (reading e-book). User is swapping page 1 out of the screen, and the system is going to load page 2. By lazy loading, the data of page 2 will not be loaded into memory until the last moment (when page 1 reaches the edge of the screen).

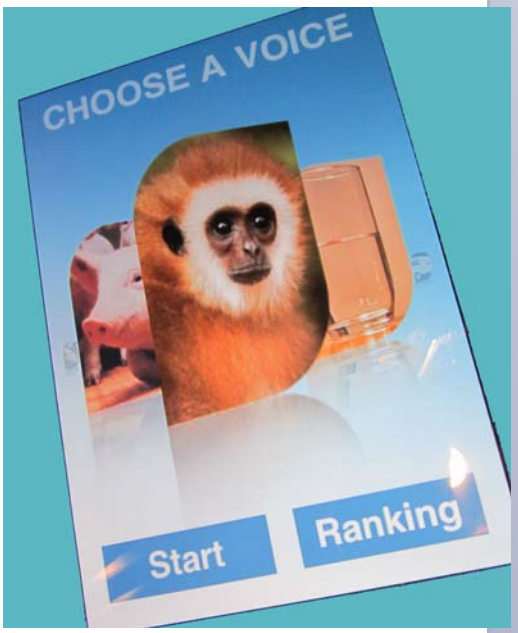
Dijkstra's algorithm, developed by Dutch computer scientist Edsger Dijkstra, is a graph search algorithm that solves the single-source shortest path problem for a graph with nonnegative edge path costs, producing a shortest path tree. This algorithm is often used in routing and as a subroutine in other graph algorithms.

There is often a trade-off among computation time, memory space and accuracy. However, the bottleneck for mobile game design usually hinges on a lack of memory space. A popular programming technique called *lazy loading* is often used to solve the problem. This technique is very popular for mobile game designs but not commonly used on family computer games. The concept is to allocate memory resources at the very last moment when the system really needs it. In this case, memory will be clean for the longest possible time. However, without data preloading, the processing time may become slow.

Another example of lazy loading is illustrated in Figure 1. Let's look at one more example of a trade-off decision. I developed an iPhone game called *soundMitate* in which a player imitates a provided sound and the program measures how accurately the user can imitate the original sound. If I develop this game for the desktop platform, precision will be the most important consideration. However, speed is the greatest concern for mobile applications. I truncate 50% less salient frequency components when performing the audio comparison. In this case, it runs 10 times faster but yet preserves 98% accuracy, which is adequate for a game. Programmers should test their game thoroughly for making trade-off decisions. Usually programmers perform the test not only by themselves, but invite a focus group to try it out on a variety of platforms for more objective and diversified feedback.

For user interface design, all information on the screen needs to be concise because of the small screen size and low resolution. For example, a mobile game should have the least number of buttons available on the screen. The less important buttons should be discarded, or first hidden and triggered to appear by shaking the device or by a drag and drop function. The button cannot be too large or else it will occupy too much space, yet they can't be too small that they can't be tracked by human eye. If it is a touch-screen device, the button needs to be large enough such that it can be controlled by finger or stylus. For example, a 57x57 pixel button under the 72-dpi environment will do the job nicely on most smartphone using a touch screen.

On the other hand, mobile game consoles do have their own specialized features. For example, many mobile game consoles can be triggered by an accelerometer or gyroscope (e.g. the *ecViolin* iPhone app triggers vibrato tone by shaking), or blowing air into the microphone (e.g. the *Ocarina* iPhone app by Smule, which imitates a blowing instrument). Also, mobile game consoles can provide location-based information. Let's take the iPhone game *soundMitate* as an



example again. It has an Internet ranking system. Players can upload their results to the cloud database with location-based information. Location-based information is also used to automatically tune the result of players according to the native language of their location. I expect to make use of the collected location-based data to further develop the system as a pronunciation learning game. Mobile game consoles certainly do initiate creativity and openness.

The world of gaming keeps on changing. We first had family computer games, and now we have games on mobile consoles. So what will be the next popular platform? Some contemporary games already go further and are designed for a daily life consoles such as on a table or a wall. In the changing world of gaming, challenges always come with opportunities, and it is good for developers and gamers to experience change!

About the Author: Simon S.H. LUI is a ….

## IEEE-HKN HISTORY SPOTLIGHT

Eta Kappa Nu (HKN) is the result of the vision of Maurice L. Carr, an electrical engineering student at the University of Illinois. He shared that vision with nine other students who together founded the organization on 28 October 1904. The original perspective regarding qualifications for membership was that scholarship was important but that the selection of students with the character and attitude that would make them probable leaders in the profession was even more important.

Eta Kappa Nu's first publication was a small four-page leaflet titled "The Electrical Field," issued in the spring of 1906. It was devoted almost entirely to the subject of employment. It contained commentary about four companies that employed electrical engineers, and it included the list of names of the members graduating from both the University of Illinois and the Purdue University chapters. This leaflet was published annually until 1909, when it began to be published semi-annually. In 1913, "The Electrical Field" was renamed "The BRIDGE" and was published annually. Today, "The BRIDGE" is published twice a year, and beginning in 2011 was available in an electronic format.


Courtesy of the University of Illinois Archives, Alumni and Faculty Biographical File, Record Series 26/4/1