Artificial and Computational Intelligence for Games on Mobile Platforms

Clare Bates Congdon[1], Philip Hingston[2], and Graham Kendall[3]

1 Department of Computer Science, The University of Southern Maine, USA
congdon@usm.maine.edu
2 School of Computer and Security Science, Edith Cowan University, Australia
p.hingston@ecu.edu.au
3 School of Computer Science, University of Nottingham, UK and Malaysia
graham.kendall@nottingham.ac.uk

Abstract

In this chapter, we consider the possibilities of creating new and innovative games that are targeted for mobile devices, such as smart phones and tablets, and that showcase AI (Artificial Intelligence) and CI (Computational Intelligence) approaches. Such games might take advantage of the sensors and facilities that are not available on other platforms, or might simply rely on the "app culture" to facilitate getting the games into users' hands. While these games might be profitable in themselves, our focus is on the benefits and challenges of developing AI and CI games for mobile devices.

1998 ACM Subject Classification I.2.m Artificial Intelligence, miscellaneous

Keywords and phrases Games, mobile, artificial intelligence, computational intelligence

Digital Object Identifier 10.4230/DFU.Vol6.12191.101

1 Introduction

Games are an appealing application to showcase AI (Artificial Intelligence) and CI (Computational Intelligence) approaches because they are popular and ubiquitous, attracting a diverse range of users.

Mobile games are easier to bring to market than commercial (large scale) video games. This makes them a practical choice for development and study in an academic environment, using relatively small teams of academics and students, who are able to work on relatively low budgets. For example, the small screen size and lack of powerful graphics hardware typical of mobile devices means that simple graphics, often only 2 or 2.5 inches, are expected, so that large teams of highly skilled artists and 3D modellers are not required.

Mobile devices usually provide a wider variety of input data (touch, location, images, video, sound, acceleration, orientation, personal data, data from/about other users etc.) than is normally available on a desktop or laptop computer and offer a full range of output options (images, video, animation, sound, vibration, wireless, bluetooth, infrared) as well. In addition, the popularity of mobile devices allows developers to recruit large numbers of casual users, whose interactions provide another potentially large data source for game data mining, using techniques such as those described in [6]. Novel game mechanics and interaction methods might be made possible by processing these input data using AI and CI methodologies.

Computational power and battery life present two potential obstacles to intensive AI/CI-based games, and some potential designs will require offloading some of the computation

Artificial and Computational Intelligence in Games. *Dagstuhl Follow-Ups*, Volume 6, ISBN 978-3-939897-62-0.
Editors: Simon M. Lucas, Michael Mateas, Mike Preuss, Pieter Spronck, and Julian Togelius; pp. 101–108
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany

to servers. It might also be difficult to implement large-scale, complex game worlds due to the limited resources that are available. There are also significant challenges in developing AI/CI libraries that can work with low memory, limited battery power etc., adapting or developing AI/CI methods to work effectively in games that are played in short bursts, using unreliable communications, and providing real-time responses. However, these constraints provide significant research opportunities.

Mobile devices are still "young" enough to provide opportunities for developers to implement innovative products without having to employ large specialist teams (e.g. graphic designers, musicians etc.), although some specialists are still required of course. However, devices are becoming more capable – for example, the original iPhone had a screen resolution of 480x320 pixels, a single 2 Megapixel still camera, and a storage capacity of 4–8 GB, while the iPhone 5 is 1136x640 pixels, has two 8 Megapixel cameras and can record 1080p HD video at 30 fps, has a storage capacity of 16–64 GB, and has in-built voice recognition. Applications are also becoming more sophisticated – for example, technologies like Web3D and game engines like Unity3D are bringing 3D graphics to mobile platforms [5]. Inevitably, game players will come to expect more and more from mobile games, so the opportunity for small players and enthusiasts will not last long (perhaps several years, in our estimation). Those who are interested in this area might want to explore, and capitalize, on those opportunities now. Moreover, CI/AI both provide significant opportunities both in terms of research challenges and also to make the games more interesting and more fun to play. We would like to see the research community take up the challenge to showcase what can be done with the limited resources available on mobile devices, but also utilizing the larger number of sensors (e.g. movement detection) and other options (e.g. location awareness) which are not available on traditional "living room" game consoles.

The aim of this chapter is to outline the limitations of mobile computing, with respect to utilizing AI/CI, but also draw out some of the potential advantages that could be exploited now (or certainly in the years to come) as the convergence of technology continues and offers greater opportunities than are available at present.

The rest of the chapter is presented as follows. In the next section, we present the (limited) work that has been carried out on AI/CI for mobile devices. In Section 3 we lay out what we believe are the defining characteristics of mobile environments. In Section 4 we outline the challenges faced when using mobile devices. Section 5 presents the opportunities that arise when using a mobile device, rather than a desktop, console, or other *stationary* computer. In Section 6 we provide some insight as to what AI/CI can offer mobile computation. We also outline some possible projects that would be feasible at this time, as well as some thoughts as to what might be possible in the next 5–10 years. Section 7 concludes the chapter.

2 Prior Work

We were able to find only a limited amount of work that considers AI/CI in mobile games and there seems to be limited scientific literature about using AI/CI on mobile devices at all. In this section, we summarize the few papers we did find, on AI/CI for games as well as for non-games on mobile devices.

In one gaming example, Aiolli and Palazi [1] adapted an existing machine learning algorithm to enable it to work within the reduced computational resources of a mobile phone. Their target was the game "Die guten und die bösen Geister", which is a board game requiring the player to identify which game pieces (Ghosts) are "good" and which are "bad". Therefore, an AI opponent for the game would need to be able to perform a simple classification task. The more usual classification algorithms were rejected on the basis of requiring too much memory or too much computation. Instead the authors opted for a very simple system based on two prototype feature vectors, one for good and one for bad ghosts. Unfortunately, they did not report any comparison of performance of this simple scheme over more complex classifiers, but the point is that for such applications, there is a trade-off to evaluate between accuracy and computational resource requirements. There was also no evaluation of the different schemes in terms of player satisfaction.

In a more recent example by Jordan et al. [10], the authors report on a research prototytpe *BeatTheBeat*, in which game levels are matched to background music tracks based on features extracted from the audio signal, and these are allocated to cells on a game board using a self-organising map.

In a paper discussing the potential uses of AI methods in serious mobile game, Xin [13] suggests that, while AI methods could add value to such games, computational requirements might require a client-server solution, offloading the AI to a server.

Although not focusing on games, Kruger and Malaka [11] argue that AI has a role in solving many of the challenges of mobile applications, including

- Location awareness
- Context awareness
- Interaction metaphors and interaction devices for mobile systems
- Smart user interfaces for mobile systems
- Situation adapted user interfaces

This paper introduces a special issue of the journal Applied Artificial Intelligence containing articles describing the state of the art as it was in 2004. Many of these same challenges may provide opportunities for novel mobile game concepts based on AI/CI.

In [2], Baltes et al. describe their experience with implementing high-level real-time AI tasks such as vision, planning and learning for small robots, using smart phones to provide sensing, communication and computation. Although their aims are different from ours, many of the research challenges in terms of implementing AI solutions with limited computational resources are similar. Their robots' agent architectures are based on behaviour trees described using a XML schema, and translated off-line into efficient C code. Vision is based on fast approximate region detection. A standard algorithm was found to be too slow and was modified to take advantage of specific domain knowledge (e.g. expected object colors). Another high-level task that they tackled was multi-agent simultaneous location and mapping (SLAM). Once again, the task was simplified by taking advantage of the structured environment (robot soccer). BlueTooth was used to share information between agents. A particle filter method was used to maintain estimates of the robots' poses, with a limited particle population size dictated by the available memory. We see that the researchers used a variety of strategies to cope with the limitations of the computing platform: offline pre-processing, modification and simplification of algorithms for specific tasks and environments, and sharing of information between mobile devices. We expect that some of the same strategies and even some of the same algorithms will be applicable in both the robotics and games domains.

3 Characteristics of a Mobile Environment

Our working definition of a mobile device for game playing is a device that is networked, and is small enough to be mobile, yet still provides a platform for general computation. In the future, one might imagine that many kinds of mobile devices might be used in games.

For example, a car's GPS system might be used in a futuristic version of a scavenger hunt car rally (scavenger hunt games for mobile phones already exist – e.g. SCVNGR, textClues). However, at the present time, we are chiefly thinking of smart phones and tablets.

While computational resources (CPU, memory, persistent storage) are available on these devices, they are all limited in comparison to standard platforms, and limited battery power is an additional (and major) consideration.

On the plus side, these devices usually have a number of other features that are often not available, and especially not all together, on "standard" gaming platforms:

- **location services** – whether by GPS, WiFi or cell tower triangulation;
- **personal ownership** – generally one person is more or less the sole user of a particular device.
- **Internet access** – to data, services and other users;
- **multiple modes of connectivity** – WiFi, Bluetooth, 3G/4G may be provided, and it is expected that connectivity will not be continuously available.
- **a range of non-standard sensors** – touch screen, camera (for image capture and subsequent processing), microphone will probably be provided, and others may be, such as a gyroscope, accelerometer and video camera;
- **non-standard outputs** – a small screen, some sound, possibly vibration.
- **other app data** – apps may be able to share data, especially with social media platforms.

Also, usage patterns for these devices are often different from those on standard game platforms such as PCs – games are often played in short bursts (waiting for a meeting, on a bus/train etc.), and gameplay may be interruptible.

In designing and implementing games for mobile devices, these differences combine both to provide challenges, which AI and CI have the potential to solve, and to provide opportunities for novel game concepts based on or supported by AI and CI methods.

4 Challenges When Using AI/CI on Mobile Devices

Mobile devices introduce a number of constraints to game design:

- Limited CPU and memory in some ways harken back to the days of early video games.
- Small screen size limits graphical complexity.
- The reality that these devices are typically used when running on a battery further encourages limiting CPU and memory usage beyond what is physically available on these devices.
- However, real-time responses are often called for with mobile devices.
- Connectivity issues must be kept in mind, as devices may lose signal either while out of range of a cell tower or due to a user opting not to pay for wi-fi access at a given location.

It is our thinking that these challenges provide interesting constraints when designing AI/CI-based games, as will be discussed in the next section.

5 Opportunities When Using AI/CI on Mobile Devices

There are some limitations to using mobile devices for gaming (such as small screen size, limited battery life, less powerful processors etc.) but there are also many opportunities for utilizing mobile devices, which are not present on static devices. We briefly mentioned some of these in the introduction, but in this section we discuss these opportunities in a little more detail.

5.1 Small Screen

Having a smaller screen could be seen as a limitation but it could also be viewed as an opportunity. Having limited graphic capabilities means that the programmers may not have to focus as much on this aspect of the system as would be the case if you were designing a system that had a large screen, high resolution and a powerful graphics processor to assist with the processing required in rendering the screen (although screen resolutions are improving and mobile phone GPUs are becoming more powerful). If a programmer's (or researcher's) skills are in AI/CI, then having a platform which is relatively easy to program could be an advantage as you are able to focus on the AI/CI, without having to be so concerned about the graphics. This may also reduce the need for artists on the project team. Of course, as technology continues to develop, the advantages that we outline here will gradually diminish, and the quality of graphics and art will become a higher priority.

5.2 Location Awareness

A static computer, by its nature, is stationary, and this could be seen as one of its major limitations. A gaming device that is able to be in different geographical locations at different times, opens up a range of possibilities that were not available even a few years ago. It is obvious that having devices that can be moved around offers many opportunities but the focus of this chapter is to look at those opportunities from an AI/CI point of view. AI/CI could be utilized in a variety of ways. As the player roams around the game (both physically and within the game world) the AI/CI agent could tailor the game playing experience to meet the expectations of the players.

5.3 Interaction with Other Players

Having a capability such as Bluetooth provides opportunities to meet with other players that are in a similar location, but you were not aware that they were there. This would be useful in locations such as a city center but imagine how many people are potentially with a few feet of you at a sporting event or a concert. Once the application had identified potential game 'buddies' the AI/CI could be used to validate the other person's skill level, whether they are actually a match for you to play with etc. A lot of innovation in gameplay is taking place in the mobile market. A couple of examples are Fingle (a bit like the classic ice-breaking game, Twister, but for hands on a tablet – `http://fingleforipad.com/`) and Swordfight (an example of a Phone-to-Phone Mobile Motion Game [15]).

5.4 Social Media

Mobile platforms already take advantage of the many social platforms that are available. Facebook and Twitter are probably the most well known but there are hundreds, if not thousands, of other platforms that offer users the ability to communicate with one another. Indeed many people, we suspect, use their phone more for texting and updating their status rather than for making phone calls. If a networked, mobile platform is used for game playing, users might want to update their various social networking sites with the games they are playing, their progress, their high scores, who they are playing with etc. This could place a burden on the user who does not have the time to disseminate all this information, but still wishes it to be known. AI/CI could be used to learn when/what the user wishes to update to social networking sites. For example, a user might always tweet a new high score, but not update their facebook page. Another user might keep a certain person regularly updated about their progress through a game via social media messages aimed at just that user. The challenge is to learn what to update and when, and provide the API (Applications Programming Interface) to the various social media feeds, many of which already exist.

5.5 AI/CI Libraries for Use in Mobile Games

The limited CPU and memory resources typically available on mobile devices suggest the need for AI and CI libraries specifically designed for mobile applications. Two approaches come to mind. Firstly, for applications that require execution on the device itself, stripped down and simplified implementations of common algorithms would be useful. On the other hand, for applications where a client-server model is appropriate, cloud or web service based implementations would be a good solution.

In the academic literature, we could not find any examples of the first kind of any substance. However, there are many examples of small libraries from the open-source community that could provide a good starting point. Many of these examples are implemented in Lua, a scripting-like language with object-oriented capabilities that is commonly used for games. Some examples are Abalhas, which is a PSO implementation in Lua (by Alexandre Erwin Ittner, available at `http://ittner.github.com/abelhas/`), LuaFuzzy, a fuzzy logic library written in Lua (`http://luaforge.net/projects/luafuzzy/`) and LuaFann, a fast artificial neural net implementation (`http://luaforge.net/projects/luafann`). One could perhaps envisage a collection of small, modular library components, written in Lua, and covering these AI and CI technologies, along with others such as evolutionary algorithms, a Lua version of OpenSteer, an A^* implementation, a lightweight rule-based system library perhaps based on CLIPS, and so on.

Of course, this is only one possible development path. For example, web-based development using JavaScript in conjunction with native code, as discussed by Charland et al [4] is another possibility. There are also existing open-source AI and CI codes, such as JMLR MLOSS (`http://jmlr.csail.mit.edu/mloss/`), implemented in various languages such as C++, Java or Python. While there may be issues such as size and portability to overcome, much of this could also be utilised : we point out the Lua pathway as one that might work particularly well for mobile games.

There are also examples of cloud-based implementations of AI and CI technologies that might be utilised in a client-server approach for mobile games. For example, there is Merelo et al.'s cloud-based evolutionary algorithm [7], Li's cloud-based fuzzy system [12] and Haqquni et al.'s cloud-based neural network system [9]. Apple's SIRI is an example where local processing is combined with higher performance cloud-based processing to solve an AI problem – speech recognition.

6 What Can AI/CI Offer for Games on Mobile Devices

6.1 Procedural Content Generation

Using AI/CI methods for Procedural Content Generation (PCG) in games is an active research area with some notable successes in recent years. Spore is one high-profile example in the commercial arena. We argue that several factors make mobile games well suited for PCG. Firstly, in terms of typical length of play sessions and complexity of typical game environments, mobile games are smaller in scale than many standard video games. This should mean that PCG is achievable with limited computational resources, and could be done locally on the device, without having to offload the task to a server. Second, some of the more interesting AI/CI methods for PCG make use of player preferences, either in real-time or offline. Mobile games with many players would have a ready source for the training data needed to drive these systems.

For example, Interactive Evolutionary Computation (IEC) is a CI technique that could be very well suited for mobile games. Hastings et al. have applied this technique successfully in *Galactic Arms Race* [8]. This game features weapons defined by particle systems controlled by a kind of neural network called a *Compositional Pattern Producing Network*, and these are evolved using *cgNEAT*, a version of *Neuro-Evolution by Augmenting Topologies*, where fitness is determined by popularity of player choices in the game. The authors coined the term *collaborative content evolution* to describe this approach.

The mobile game-playing population could provide an ideal environment for collaborative content evolution, with a large pool of players, playing many short game sessions, providing a very large number of judgements to feed into fitness calculations. Crowd-sourcing used in this way should enable content to evolve rapidly, giving game players a constantly novel, changing game experience, guided by the preferences of the players themselves.

6.2 Personalisation and Customisation

Recently, CI techniques are being used to adapt gameplay to optimise player satisfaction in real time. For example, Yannakakis and Hallam reported success with using neural network based user models adjusted in real time to improve player satisfaction in "playware" games [14]. Using the kind of lightweight libraries proposed in 5.5, this kind of gameplay adaptation and other customisation could be added to mobile games, and neural networks and other machine learning methods have already been proven to be effective for adaptation in other, non-mobile games.

6.3 Ubiquitous Games etc.

The terms *ubiquitous or pervasive* computing have been in use for some time now. As far back as 2002, these terms were also applied to games (see e.g. [3]). There's obviously a considerable overlap between these kinds of games and mobile games – mobile devices provide the means of achieving ubiquity/pervasiveness. A related concept is that of the *augmented reality game*. Here too, modern mobile devices have the camera, audio, and display capabilities to support augmented reality applications. For ubiquitous games, real-time adaptation with CI algorithms running on the device, could be combined with periodic sychronisation with a cloud-based repository, so that the learned personal profile can be shared across locations and devices. For augmented reality games, either a generic light-weight augmented reality library or perhaps some application specific implementation in the style of Baltes et al. [2], could be used.

7 Conclusions

Mobile platforms are already widespread and their use is largely for interacting with social media sites and for tweeting. Some people also use them for what they were originally designed for, making phone calls. Game playing is becoming more widespread on these devices, more so on phones than tablets, with around a third of mobile phone owners reportedly playing mobile games (see, for example `http://www.infosolutionsgroup.com/popcapmobile2012.pdf`). Computational Intelligence and Artificial Intelligence are not often present in these games, or if present, are unsophisticated. However, there is a window of opportunity where we are able to integrate these technologies into these games, with less of the now usual overhead of having to work with graphic designers, musicians, plot design etc. As mobile platforms develop the complex, large teams associated with console based game design are likely to converge such that it may be more difficult, if not impossible, to enter this market.

In this chapter, we have outlined some of the opportunities and challenges in introducing CI/AI onto mobile platforms. We hope that the research community will take up the many research challenges that exist in this exciting, fast moving area.

References

1 F. Aiolli and C. E. Palazzi. Enhancing Artificial Intelligence on a Real Mobile Game. *International Journal of Computer Games Technology*, 2009.

2 Jacky Baltes and John Anderson. Complex AI on Small Embedded Systems: Humanoid Robotics using Mobile Phones. In *AAAI Spring Symposium Series*, 2010.

3 Staffan Björk, Jussi Holopainen, Peter Ljungstrand, and Regan Mandryk. Special issue on ubiquitous games. *Personal and Ubiquitous Computing*, 6:358–361, 2002.

4 Andre Charland and Brian Leroux. Mobile application development: web vs. native. *Commun. ACM*, 54(5):49–53, May 2011.

5 Kevin Curran and Ciaran George. The future of web and mobile game development. *International Journal of Cloud Computing and Services Science (IJ-CLOSER)*, 1(1):25–34, 2012.

6 Anders Drachen, Christian Thurau, Julian Togelius, Georgios N Yannakakis, and Christian Bauckhage. Game data mining. In *Game Analytics*, pages 205–253. Springer London, 2013.

7 Juan Julián Merelo Guervós, Maribel García Arenas, Antonio Miguel Mora, Pedro A. Castillo, Gustavo Romero, and Juan Luís Jiménez Laredo. Cloud-based evolutionary algorithms: An algorithmic study. *CoRR*, abs/1105.6205, 2011.

8 Erin J. Hastings, Ratan K. Guha, and Kenneth O. Stanley. Automatic Content Generation in the Galactic Arms Race Video Game. *IEEE Transactions on Computational Intelligence and AI in Games*, 1(4):245–263, 2009.

9 A.A. Huqqani, Xin Li, P.P. Beran, and E. Schikuta. N2Cloud: Cloud based neural network simulation application. In *2010 International Joint Conference on Neural Networks (IJCNN'10)*, pages 1–5, 2010.

10 A. Jordan, D. Scheftelowitsch, J. Lahni, J. Hartwecker, M. Kuchem, M. Walter-Huber, N. Vortmeier, T. Delbrugger, U. Guler, I. Vatolkin, and M. Preuss. BeatTheBeat music-based procedural content generation in a mobile game. In *2012 IEEE Conference on Computational Intelligence and Games (CIG'12)*, pages 320–327, 2012.

11 A. Kruger and R. Malaka. Artificial Intelligence Goes Mobile. *Applied Artificial Intelligence: An International Journal*, 18(6):469–476, 2004.

12 Zhiyong Li, Yong Wang, Jianping Yu, Youjia Zhang, and Xu Li. A novel cloud-based fuzzy self-adaptive ant colony system. In *2008 Fourth International Conference on Natural Computation (ICNC'08) – Volume 07*, pages 460–465, Washington, DC, USA, 2008. IEEE Computer Society.

13 Chen Xin. Artificial Intelligence Application in Mobile Phone Serious Game. In *First International Workshop on Education Technology and Computer Science (ETCS'09)*, volume 2, pages 1093–1095, 2009.

14 G.N. Yannakakis and J. Hallam. Real-time game adaptation for optimizing player satisfaction. *IEEE Trans. on Computational Intelligence and AI in Games*, 1(2):121–133, 2009.

15 Zengbin Zhang, David Chu, Xiaomeng Chen, and Thomas Moscibroda. Swordfight: enabling a new class of phone-to-phone action games on commodity phones. In *MobiSys*, pages 1–14, 2012.