European Journal of Law and Technology, Vol 11 No.3 (2020)

EJLT European Journal of Law and Technology

Intellectual Property and the Protection of Apps in the European Union

Helen Gubby, Jos Klaus, Kees van Noortwijk[1]

Abstract

Applications ('apps') for mobile platforms – such as mobile phones or tablets – are a product of the digital age, usually with specific characteristics that set them apart from other types of computer software, for instance from applications designed for desktop computers. Producing these apps can be a lucrative business and for some companies the use of apps has become vital. As apps have become embedded in both the personal and commercial contexts, it is not surprising that competition in apps markets has become intense and that apps may be vulnerable to unauthorised copying. The question then arises: what forms of protection can intellectual property (IP) rights offer app developers? Determining the best and most cost-effective forms of IP protection can be difficult for app developers.

In this article the various components of apps are examined individually and an analysis is made of which forms of IP would be the most suitable. These components are (i) the source code, (ii) the Graphical User Interface (GUI), (iii) movements and transitions, and (iv) logos, icons and fonts used in an app. The focus is on copyright, design and trademark rights rather than patents, as patents are generally less relevant to the average app developer. What the analysis reveals is that the IP protection of digital components is still in the main 'unexplored territory'. Although the law is relatively clear with regard to some aspects, it still contains grey areas as well due to a lack of case law.

Keywords: App Protection, Intellectual Property Law, Copyright, Design right, Trademark

[1] Dr Helen Gubby is a Senior Lecturer at the Rotterdam School of Management, Erasmus University, Rotterdam, the Netherlands; Jos Klaus LLM is an Attorney-at-Law at the Intellectual Property and Technology department of the law firm BarentsKrans N.V. in The Hague, the Netherlands; Dr Kees van Noortwijk is an Associate Professor of Law and Technology at the Erasmus School of Law, Erasmus University, Rotterdam, the Netherlands.

1 Introduction

Apps, computer programs written especially for mobile apparatus such as smartphones and tablet computers, are now deeply embedded in daily life.[2] We keep in contact with each other through apps like WhatsApp, order our taxi via Uber or even find our true love via dating apps like Tinder. For companies, an app can be a key asset, often supporting (or sometimes even: being) the company's goods or services. Given the significance of apps both in the personal and commercial spheres (with an estimated annual revenue of the app market of circa USD 189 million in 2020)[3], it is remarkable that there has been relatively little attention paid to the relationship between intellectual property (IP) protection and apps either in doctrine or case law.

Producing apps can be a lucrative business, but there can be intense competition. Yet even where rivals vie for market share, legal practice shows that IP rights are often not taken into account by app developers. Sometimes this can be ascribed to the naivety of app developers because of their ignorance concerning IP in general. However, some developers have the impression that it is simply not possible to prevent others copying their apps, in whole or in part. Others believe that obtaining IP protection is too costly. But is that true?

An app is basically a piece of computer software. It can still be difficult to predict with certainty how courts will determine cases of alleged IP infringement with respect to software. From the early days of commercial desktop software, lawyers have struggled to fit software into the traditional IP framework, a framework largely developed in a pre-digital age. However, because of the special characteristics of the 'platform' – the mobile device – it is designed to run on, an app differs fundamentally from other forms of software, for instance applications for desktop computers. This is expressed by, for example, eye-catching and interactive graphical user interfaces and the incorporation of gestures, such as swiping. This makes apps into a particular category of software, a category that in itself mandates specific attention, yet this specific attention has been largely lacking in legal discourse. In this article, we examine the differences between apps and desktop software and the legal ramifications of these differences.

Given the relative paucity of app case law and legislation directed specifically at apps, exactly what aspects of apps can be protected by which forms of IP is not always clear cut. Nonetheless, it is helpful for app developers to become more familiar with the possibilities IP law offers them and to learn which forms of IP would provide them with the most

[2] According to the World Intellectual Property Organization (WIPO), around three million apps are currently available, see WIPO, *Intellectual Property and Mobile Applicatons*, p. 8. Accessed 30 October 2020.

[3] Ibid.

efficient and cost-effective protection. This article analyses the relevance of different forms of IP with respect to apps within the European legal context. From experience gained in legal practice, the approach adopted here is to examine the IP protection possibilities for apps by focusing on the four most distinctive aspects of apps: (i) source code; (ii) graphical user interface (GUI); (iii) movements and transitions; and (iv) logos, fonts and icons.

2 Mobile versus desktop applications

Mobile devices such as smartphones and tablets are usually unable to run software designed and built for desktop computers. Although these devices nowadays are equipped with remarkably powerful processors, their hardware and 'operating systems' (such as IOS or Android) are too different from desktops in order for desktop software to run on them without alterations. In this paragraph we will give a brief overview of the most important differences and of the consequences for software development and use.

Vithani & Kumar (2013)[4] describe these differences and argue these are so fundamental that the two types of software (for mobile versus for desktop platforms) usually show completely different 'development life cycles'. Mobile applications usually have functionalities that can be complex, but are more limited in number than is the case for desktop applications. Also, they can be built using existing 'development frameworks', which shields off complicated operating system issues and allow the developer to concentrate on the essence of the application. As these development frameworks are based on well-known programming languages such as Java or C# (C-Sharp), the number of developers capable of crafting these applications is large. At the same time, competition between mobile application vendors is intense. These two factors lead to a shorter life span for most mobile applications, compared to their desktop counterparts.

Focusing on functionalities, mobile applications can be complex because they might interact with telephony, a built-in camera, GPS, etc. On the other hand, traditional methods for data input and output (screen, keyboard, pointing device) are more restricted in mobile devices. In other words, physical interfaces are rather different from those in desktop computers.

Another aspect is the use of scarce resources, such as working memory and (battery) power. Optimizing these resources is a major necessity for mobile application developers,

[4] Tejas Vithani & Anand Kumar, 'Modeling the Mobile Application Development Lifecycle', in: *Proceedings of the International Multiconference of Engineers and Computer Scientists 2014*, Vol. I, Hong Kong: IMECS 2014.

which can heavily influence – and hence complicate – application design. Another factor that causes complications is that mobile operating systems differ significantly from each other. A 'native' app developed for Android usually cannot be converted easily to a 'native' IOS app, unless a specific 'cross platform' development framework is used. Targeting multiple platforms can also complicate the testing and 'debugging' of the applications significantly.[5] All of this requires extra effort from software developers.[6]

Finally, mobile applications show differences with respect to the delivery of updates. As most mobile apps are downloaded from internet-based 'app stores', their users are automatically registered with the software vendor and updates can be brought to them more easily than with desktop software (which in many cases can be used anonymously). Users have become accustomed to that means of acquiring updates, and hence expect frequent and automatic updates for the apps they have installed. This in turn puts more pressure on mobile app developers and vendors. All of these factors influence the software development process for mobile applications, and are the reason that these applications show fundamental differences compared to traditional desktop applications.

These differences can be legally relevant. The shorter software life cycle combined with specific, extra efforts to produce it are the cause of higher investments to be made, which puts extra pressure on the protection of the finished product and therefore of these investments. The almost inevitable use of software development frameworks can complicate license conditions for the apps built using these frameworks. And most importantly, the user interfaces, including the ways to interact with mobile apps, are not only different from those in desktop applications, but are often also very specific and refined. Copying elements from a well-designed and/or well-known user interface could therefore be lucrative, which puts extra pressure on their protection.

3 Source code

3.1 Options for protection

Like desktop software, the source code of an app, the code as written by the programmer, can be copyright protected. In the authors' view, the established legal framework should therefore be applicable. Obtaining copyright protection in an EU member state does, in

[5] See for examples of this Davi Bernardo Silva, Marcelo Medeiros Eler, Vinicius H.S. Durelli & Andre Tekeshi Endo, 'Characterizing mobile apps from a source and test code viewpoint', in: *Information and Software Technology* 101 (2018), p. 32-50.

[6] The particular issues related to cross-platform development of (mobile) applications are analyzed in-depth in W.S. El-Kassas, B.A. Abdullah, A.H. Yousef and A.M. Wahba, 'Taxonomy of Cross-Platform Mobile Applications Development Approaches', in: *Ain Shams Engineering Journal* (2017) 8, p. 163-190.

principle, not require registration and it is obtained free of charge.[7] There are limits to copyright protection: it extends only to the authorial aspects of the work. It does not extend to functionality or underlying ideas; a specific trend or style is not protected as such, separate from its expression in the author's work. This means that copyright will not protect an idea for an app, such as a dating app or car auction app in general. Anyone would be free to use that idea.

Technical functionality requires patent protection. Yet Article 52(2) of the European Patent Convention 1973 (EPC) specifically excluded computer programs 'as such' from patentability. This may seem strange, given that the source code would appear to have a technical character and European patent law requires a patentable invention to be a technical solution to a technical problem. The rationale behind this decision was that source and object code were seen simply as a set of instructions, and therefore not as an invention. If it is not an invention, it is not patent eligible.

The exclusion of computer programs from patentability will not apply, however, if the software is a component in the operation of a technical system or the control of a technical process. What is required in order for software to become patentable is that the software contributes to a further technical effect. However, the requirements for patentability under the EPC for all inventions come into play; the invention must be new, involve an inventive step and be industrially applicable. For example, the Board of Appeal of the European Patent Office held that improving the speed of a computer program was not in itself sufficient to establish an inventive step. Only if the alleged speed-up had affected an established technical effect could the claimants have argued that the faster program contributed to a technical effect and thus to an inventive step.[8]

In the case of apps, many technical functions have long become standard and would not fulfil the requirements of 'new and inventive step'. Furthermore, it can easily take four or five years before the patent is actually granted if it survives the examination process; in a fast-moving sector like apps by that time even if the patent is granted, it could already be for outdated technology. The cost of obtaining patent protection is also relatively high and, if it is granted, increases if the patent is maintained for several (European) countries over a longer period of time. A patent is therefore generally less relevant to many apps developers.

Looking at the EU framework, the Software Directive does require member states to 'protect computer programs, by copyright, as literary works within the meaning of the Berne Convention for the Protection of Literary and Artistic Works'. A computer program will be protected 'if it is original in the sense that it is the author's own intellectual creation.

[7] The Berne Convention as revised in 1908, and reflected in Article 5(2) of the Paris text1971, according to which the enjoyment and the exercise of copyright shall not be subject to any formality. However, several Berne Convention signatories have established voluntary national registration systems for copyright. This voluntary system of registration is generally aimed at identifying the work and serving as evidence in court in litigation disputes (for example, in order to prove more easily the date of creation of the work).

[8] EPO *Microsoft/On-demand property system*, (T130/11), 11 March 2016

No other criteria shall be applied to determine its eligibility for protection.'(Art.1 (1) and (3).[9] An 'own intellectual creation' is therefore mandated.

What is required in order to fulfil the requirement of originality, as being the author's 'own intellectual creation'? Although copyright laws as such have not yet been subject to EU-wide harmonisation, the bar may not be high. Important in this respect is the decision of the Court of Justice of the European Union (CJEU) in the *Infopaq* case in 2009.[10] That case concerned publishing extracts from articles in a daily newspaper, which consisted of a search word plus five preceding and five subsequent words. It was held that, although words in isolation would not be an 'own intellectual creation', words in combination, such as in parts of a sentence, could be – in this specific case – an 'own intellectual creation'. It would seem that 'own intellectual creation' requires only that the author is in a position to make a choice or selection and arrangement; making choices dictated simply by creativity rather than technique justifies an 'own intellectual creation'. Publishing these extracts was consequently copyright infringement. In software development, the number of relevant options a programmer has at his/her disposition is usually high, even taking into account that parts of the software already belong to the *Umfeld* and therefore do not enjoy protection. This in itself would seem to be an indication that a computer program could easily pass the originality test. There can be complicating factors, however, for instance when a programmer re-uses the work of others in the creation of a new program.

3.2 Who is the copyright owner of the source code?

Even if the requirements for copyright protection are fulfilled there may be another problem: it is not always clear who the copyright owner is. If an app developer is an employee and the app is developed in the course of employment, within Europe the copyright owner will generally be the employer. This however does not apply if the app developer is working on a freelance basis and has not made arrangements to transfer his/her IP rights to the client.

Complications in ownership can also arise when source code builds upon existing programs. Versions could be modified and updated by different parties. Furthermore, much computer software builds on existing software or makes use of specific 'building blocks' or 'development frameworks'. Consequently, part of the source code could be the intellectual property of a third party. If use is made of the services of multiple individual developers, each being responsible for a part of the software, the copyright to the completed work may be owned by several natural persons or legal entities. As already noted, in the case of apps users expect frequent and speedy updates for their apps, which requires an ongoing development process. When multiple right holders are involved,

[9] Software Directive 1991 replaced by Directive 2009/24/EC
[10] *Infopaq International v Danske Dagblades* [2009] ECDR 16 (C-5/08)

matters are complicated even further as such a continuous process requires the careful management of each developer's IP rights in order to ensure the legality of every new software version.

Consequently, strict version management is important if there is a dispute as to who was the first to create the code. There are various means to enable the app developer to prove his/her right to the copyright for the code. App developers can store the code with a legal professional, such as a civil-law notary. Another means of proving authorship is to deposit the code at the Benelux Office for Intellectual Property (BOIP): the source code of the software can be stored in an i-DEPOT. An i-DEPOT is a sort of 'digital vault' by which one can prove that the author deposited that particular code at the stated date. The i-DEPOT is not restricted to Benelux countries; it is not subject to any territorial limitation. The costs of an i-DEPOT are very low which contributes to the useful character hereof. European courts, particularly Benelux courts, tend to accept an i-DEPOT as evidence. Although outside Europe it is up to the court in the country in question to decide whether to admit it to the proceedings, it is difficult to see why an i-DEPOT should not be admissible as evidence anywhere in the world.

3.3 Using open source software licences

For many software development companies, keeping their source code secret is absolutely vital. What is provided to customers is only the object code, which is the machine code necessary for the program to function. The software package, therefore, does not provide users with the option to alter and redistribute the source code. Open source software (OSS), on the other hand, does provide recipients with the source code, enabling users to inspect the code, modify it and improve it. Using OSS can be an attractive proposition both to businesses in general, as it could reduce cost and dependence on software companies, and to app developers.

A word of caution is necessary, however. There is a tendency for more naïve firms to presume that OSS means that no IP rights are involved. This misapprehension can lead them to presume that they have proprietary rights over the OSS that they have modified or improved, whereas in fact they might have no such right. For example, licences like the GNU General Public License (GPL) do not allow their software to become proprietary software: any additions or improvements made by anyone using software under this licence must make their entire code freely available to the public. OSS is subject to licence and it is precisely because there is copyright protection for software that software authors can issue licences determining the conditions of use of their software creations. Even large corporations cannot always escape from this provision. Tesla has used open source software, such as Linux Kernel, Buildroot, Busybox and QT to build its operating system and features. It consistently failed to comply with the terms of the licences: no source code was given or offered to the user. In May 2018, under pressure to comply with the licensing conditions, Tesla finally started releasing some of its source code on GitHub.

4 Graphical User Interface (GUI)

A GUI acts as a conduit, an interface, between the user and the computer – enabling the user to communicate with a hardware device through various symbols, visual metaphors and pointing devices.[11] The EU Software Directive defines an interface as 'the parts of the program which provide for such interconnection and interaction between elements of software and hardware.'[12] In this communication channel between the user and the computer, text, graphical images and movements can play a role. The interface is an essential part of an app and indeed may make the app instantly recognisable to users. The GUI is usually a significant part of every computer program, because it constitutes the main output medium for any information directed at the user. However, for mobile apps the GUI is also the most important medium for user *input.* In practically every mobile app, a touch-sensitive screen acts as the main input device, which can be used for typing, pointing, scrolling and sometimes even for app-specific activities such as zooming in or out, moving forward or backword or switching to a different mode, activities for which on a desktop computer the keyboard or a separate pointing device such as a mouse is used. This extended input functionality is an important element of a mobile app's GUI and makes it an even more significant part of the software than is the case in desktop software. Therefore, the urgency of protecting the GUI well for app developers is probably even higher.

4.1 Copyright

Does a GUI qualify for copyright protection as a computer program under the Software Directive? The answer to that is now clear: no, it does not. The CJEU ruled in the BSA case that the GUI is not an expression of a computer program, as a GUI simply enables a user to interact with the computer program by clicking on the screen.[13] A GUI therefore falls outside the copyright protection provided by the Software Directive.

However, that does not mean that a GUI cannot be protected by copyright at all. In that same decision, the CJEU ruled that a GUI can be protected by copyright under the national copyright laws of the European Union member states by virtue of Directive 2001/29.[14] The CJEU stated that it is for the national court to ascertain whether the GUI meets the national criteria for copyright protection. In order to help national member state courts decide the issue, the CJEU provided certain guidelines:

[11] Brittanica online edition, https://www.britannica.com/technology/graphical-user-interface. Accessed 30 October 2020.

[12] Software Directive 2009/24/EC, Recital (10)

[13] BSA case: *Bezpečnostnĭ softwarová asociace - Svaz softwarové ochrany v Ministerstvo kultury* [2010] (Case C-393/09)

[14] The Directive referred to here is the Information Society Directive (Directive 2001/29)

> *When making that assessment, the national court must take account, inter alia, of the specific arrangement or configuration of all the components which form part of the graphic user interface in order to determine which meet the criterion of originality. In that regard, that criterion cannot be met by components of the graphic user interface which are differentiated only by their technical function*.[15]

This ruling by the CJEU means that copyright protection is dependent upon a court's interpretation of national copyright legislation. For example, with respect to the Netherlands, the *Endstra* ruling by the Dutch Supreme Court will therefore apply: this requires that a work must possess 'its own original character and bear the personal stamp of its author'. The case has given rise to unprecedented discussion and debate among Dutch scholars and practitioners.[16] Also, a selection of elements that, individually, do not enjoy copyright protection (e.g. due to technical functionality or a lack of own original character) may still enjoy copyright protection, as long as the selection is, so to say, original.[17] Therefore, although there is a lack of case law in the Netherlands, a GUI that consists of elements that are already known to the public, may still enjoy copyright protection if the combination and/or arrangement of these elements stands out from the crowd. Yet, the scope of protection is likely to be very limited. Further, an analogy with 'classic' website design can be made – as such designs share (at least visual) similarities with GUIs. Website design case law learns that a layout of several screens, tables and fields, play a role in the establishment of copyrights. The same applies to the use of colours, fonts and alignment, the locations of text fields and the overall layout of the screen.[18] It is opportune to conclude these factors apply to GUIs of apps as well.

The British concept of originality for copyright purposes has traditionally focused upon whether the work is the product of the author's own sufficient skill, labour and effort, expense and judgment. Artistic originality or ingenuity has, in general, been irrelevant, as the emphasis is on effort and investment. With respect to non-literal copying, which is usually at stake when copyright infringement in or by a GUI is claimed, the Navitaire v Easyjet case[19] still forms the basis. In this case, new computer software had been created, but that software purposely resembled existing software, which was currently in use and made by a different company. That company claimed this newly created GUI infringed its copyrights, although it was clear that the programmers of the new software had not had access to the original software's source code. It was decided that this claim failed because

[15] BSA case, para. 48

[16] P. Bernt Hugenholtz, 'Chronicle of The Netherlands, Dutch copyright law, 1995-2000', RIDA January 2001, 187, pp. 111-175. Accessed 30 October 2020 at: https://www.ivir.nl/publicaties/download/RIDA_2010.pdf

[17] Supreme Court of the Netherlands 12 april [2013] case 11/04114, *Firma Hauck GmbH & Co. KG v. Stokke A/S*

[18] District Court of North Holland (the Netherlands) 5 november [2014], HA ZA 12-335, *Complan vQsinc;* and District Court of Utrecht (the Netherlands) 8 februari [2012]*,* KG ZA 12-4, *E2MA v Malicor.*

[19] *Navitaire Inc. v Easyjet Airline Co. & Anor*, [2004] EWHC 1725 (Ch)

it was grounded on the copying of ideas and functionalities, both of which are excluded from copyright protection by the Copyright Directive. This was confirmed and even reinforced in *Nova Productions v Mazooma Games* in 2007.[20] From this it can be concluded that the sole fact that two applications look like each other, for instance because their GUIs are alike, in itself is insufficient to assume a breach of copyright.

How far the UK approach really differs from the European notion for copyright protection is again a matter of legal discussion by UK lawyers.[21] However, arguably, by delegating adjudication to national copyright law, the CJEU has opened up the possibility of potential differences in interpretation. The danger is that there will be more uncertainty at the pan-European level regarding GUI copyright protection. As noted above, because apps require a more extensive input functionality than a desktop computer, this may be a matter of even higher concern for app developers.

4.2 Design right

Software as such cannot be design protected, as it is excluded from the Community Design Regulation EC/2002/6 (CDR) (see article 3 (b)). Apps in themselves are therefore excluded from design protection, as design rights protect only the appearance of an item. However, how a GUI looks can be very important and a design right could protect the visual aspects of a GUI.

Within Europe, an unregistered Community design (article 11 (1) CDR) is given protection free of charge and automatically for a period of three years from the date on which the design was first made available to the public within the territory of the European Union. After three years, the protection will lapse and cannot be extended. If the design has been disclosed without registration, formal design registration is still possible if the registration takes place within one year of disclosure by invoking the 'grace period' exception. Design registration can be an important means to obtain protection for the appearance of the GUI.

Protection would require that the GUI fulfils two requirements. Firstly, the design must be new, meaning that no identical design has been made available to the public (art. 5 (1) CDR). Secondly, it must have individual character; a so-called 'informed user' must consider that the overall impression of the design is different from the overall impression of designs that are already known (art. 6 (1)). Often the requirement that the design is new does not pose a barrier, as the chance that there is an identical design on the market is quite limited.

[20] *Nova Productions Ltd v Mazooma Games Ltd,* [2007] EWCA Civ 219 [12]

[21] Andreas Rahmatian, 'Originality in UK Copyright Law: The Old "Skill and Labour" Doctrine Under Pressure', IIC (2013) 44: 4. Accessed 30 October 2020 at: https://doi.org/10.1007/s40319-012-0003-4

More problematic with respect to apps is the individual character requirement: many apps are similar in style to existing apps. They have a standardized interface, based on what is already out there, e.g. a standard lay out for a list with results ('pop up menu' or 'drop down list'). The CJEU has ruled that for a design to have individual character, the overall impression it produces must be different from that produced by one or more specific designs taken individually, not by a combination of features taken in isolation and drawn from a number of earlier designs: a 'mosaic approach' is not permitted.[22] At present, there is very little available case law that would bring greater clarity regarding how the courts would interpret new and individual character specifically with respect to app design infringement.

Obtaining an optimal scope of protection can be complex. On the one hand, submitting a too detailed GUI design could easily pass the criteria for constituting a valid design, but at the same time the ambit of the protection would be reduced, as another design that deviated from it only in details could lead to the interpretation that the informed user would consider its overall impression to be different. On the other hand, a more general design for registration could fail the individual character requirement or even possibly the newness requirement, which would make registration difficult from the outset. Obtaining a successful design registration can be, therefore, quite challenging.

4.3 Trademark protection

Many GUIs are now so familiar that users can immediately link them to their respective business enterprises. A good example is the Apple GUI, instantly recognisable for many end-users as a product of the American tech corporation. The same can be said for the famous Microsoft Windows Start Menu, known from Windows 95/98/2000/XP operating systems. Therefore, a GUI – or parts thereof – can function as a designation of origin.

This raises the question of whether a GUI can be protected as a trademark. A trademark has to be capable of distinguishing goods or services from those of others and therefore has to be distinctive. [23] EU trademarks can consist of any signs, in particular words (including personal names), or designs, letters, numerals, colours, sounds, the shape of goods, or of the packaging of goods, as well as any other means of identifying the commercial origin.

[22] *Karen Millen Fashions Ltd v Dunnes Stores* [2014] (Case C-345/13)

[23] Directive 2015/2436/EU, article 4.1 (c) and, with respect to European Union trade marks, Regulation (EU) 2017/1001, article 7.1 (c).

In Europe, changes to trademark law have been made to keep pace with technology. The EU Trademark Reform Package, comprising the 2015 Trade Mark Directive[24] and the 2017 Union Trade Mark Regulation[25] has been implemented in some member states in 2017 already and should be implemented in all member states by 2019. In the UK, the Trade Mark Regulations 2018[26] implementing the new Trade Mark Directive were approved in 2018 and took effect on 14 January 2019. The new law means that the former requirement that a trademark must be represented graphically in order to be registered no longer applies. A sign may be represented in any appropriate form using generally available technology, as long as the representation is 'clear, precise, self-contained, easily accessible, intelligible, durable and objective'. A trademark must, however, be capable of being represented in the trademark register 'in a manner which enables the competent authorities and the public to determine the clear and precise subject matter of the protection afforded to its proprietor'.[27]

This development in the law opens up more possibilities for GUIs as trademarks. For example, a multi-media animation used by the GUI could be registered, in which not only visual aspects play a role. While trademark protection is possible, it may be problematic. A GUI consists partly of functional and technical aspects. The CJEU has, however, with respect to shape marks (such as the shape of an electric razor or toys) determined that trademark laws are not meant for obtaining a technical monopoly. Signs which consist exclusively of the shape which results from the nature of the goods themselves, or the shape of the goods, which is necessary to obtain a technical result are not eligible for trade mark protection.[28] By analogy, that could mean that a GUI set up in a *merely* didactic way (for example the GUI of an educational app) or in an ergonomic way (such as an app for the visually impaired) may fall foul of this ruling. However, the borders of this doctrine are vague, since GUIs are likely to bear non-technical aspects as well. Furthermore, it is possible that an entire GUI will be seen as either too complex to function as a trademark, or that a specific part of it lacks the required distinctiveness being too simple or banal. Another complication is that the GUIs of apps regularly change; every few months updates are implemented which could affect some aspects of how it looks – which may make a trademark registration redundant. There is at present a lack of case law on the GUIs of apps to bring clarity. Nonetheless, it may be worthwhile for the apps developer to explore the possibility of trademark protection. Trademarks, unlike some other forms of IP, can provide a long-term monopoly; in principle they can last indefinitely if used normally in the course of trade. This could be very useful as protection for certain specific core elements of a GUI,

[24] Directive (EU) 2015/2436 of the European Parliament and of the Council of 16 December 2015 to approximate the laws of the Member States relating to trade marks

[25] Regulation (EU) 2017/1001 of the European Parlament and of the council of 14 June 2017 on the European Union trade mark

[26] S.I. 2018/825

[27] Regulation (EU) 2017/1001 on the European Union trade mark, 14 June 2017, Recital 10 and Article 4 (b)

[28] CJEU *Philips v Remington* 18 June [2002] (C-299/99), *Lego Juris A/S v OHIM and Mega Brands, Inc.,* 14 September [2010] (C 48/09)

which the developer expects to keep using for many years. However, if a developer in practice uses a GUI that differs from the registration, vulnerability to a *non-usus* cancellation may become a risk to take into account.[29]

5 Incorporation of movement

Movement plays an important role in apps. This concerns not just the user carrying out certain actions, for example the lock screen on a mobile phone that asks the user to 'slide to unlock', but also the app itself showing certain animations and transitions, for instance when moving from one screen to another. A transition could be the opening up of a menu choice (such as the classic Start menu in Windows), but also the way in which a new screen appears, for example a transition to a screen showing a list of potentially interesting cars for the user after the user has entered search criteria for the kind of car he or she wants.

As explained in the previous section, in mobile apps there is often a very direct interaction between the user and the GUI, as the touch sensitive screen on which the user interface is projected usually also acts as the main input medium. This is also of importance in relation to movements and transitions. In many cases these are related to the user's input actions. For instance, text is scrolled at the same pace as that of the user's finger movement. The amount of time the scrolling continues is directly related to the speed of the 'sweep'. As the emphasis lies on this way of interaction, the role of movement in mobile apps can be more significant than it is in desktop applications.

Although these movements and transitions are closely connected to the GUI, it is nonetheless useful to deal with these separately. The reason is that, as mentioned above, a GUI will very often be updated by the app developer, whereas certain transitions may well have a longer economic existence. It should be noted that with respect to these aspects, the law finds itself in a rather grey and as yet underdeveloped area.

5.1 Copyright

An animation or transition could, in principle, be seen as an audio-visual work and as such covered by copyright. However, for copyright protection it would have to satisfy the requirements outlined above. It is doubtful whether a simple transition would be sufficient to trigger copyright protection, although possibly a complex transition may do so.

[29] See in this context: CJEU *Specsavers v Asda* 18 July [2013] (C-252/12)

Originality ("some form of artistic embellishment"[30]) is probably an essential requirement for copyright protection. However, given a paucity of case law and developed doctrine with respect to these forms of movement in computer applications in general and in particular concerning apps, it is not possible to provide clear guidelines as to when copyright would succeed.

5.2 Design rights

Protection of movement by design rights may be difficult, but is possible. Apple, for example, does have what is known in the USA as a design patent on the ornamental design of the lock screen that asks the user to 'slide to unlock'.

The protection of movement by design rights in Europe is possible, but there is at the present time not a developed body of law on this subject. Design rights would seem to be the most suitable option to protect the external appearance of a certain movement or transition. Transitions between screens can be submitted to the register if a 'slideshow' is uploaded. In order to be accepted, the European Trademark and Design Network: Convergence on graphic representations of designs states:

The sequence of snapshots needs to be visually related (must have features in common) and it is the responsibility of the applicant to number the views in such a way so as to give a clear perception of the movement/progression.[31]

5.3 Trademark protection

In the EU, the Trademark Reform Package has increased the possibility to register unconventional marks. It is already possible to register movement marks. Based on the EU Intellectual Property Office (EUIPO) guidelines for examiners, applicants are recommended to include a slideshow or digital animation as part of their trademark application. A description should also be added.

Just as is the case with the GUI, it is essential that a mark is capable of distinguishing goods or services from those of others. It is questionable whether a simple transition from one

[30] Noam Shemtov, *Intellectual Property and Mobile Applications*, WIPO 2019, p. 42. Accessed 28 October 2020 at https://www.wipo.int/export/sites/www/ip-development/en/agenda/pdf/ip_and_mobile_applications_study.pdf

[31] European Trademark and Design Network: Convergence on graphic representations of designs - Common Communication, 15 May 2018, p. 7

screen to another would fulfil this requirement. It may, therefore, lack distinctiveness *ab initio*. While it could be argued that the public has become increasingly familiar with these sorts of characteristics of apps and would link them to a specific enterprise (for example, double clicking in Windows to open a program or folder), it is debatable whether the relevant public would perceive a certain transition as a trademark. Once again, it is unlikely that a mundane transition would have the necessary distinguishing capacity, whilst complex transitions would suffer from being too complex to serve as a trademark. Furthermore, a complex transition might have a certain functionality as its essential characteristic, which could hinder trademark registration.

6 Logos, fonts and icons

Logos, pictograms (also in the form of icons), and fonts are not new. They have been known for many years already in a pre-digital, offline era. In the digital context, they are widespread and certainly not a unique element of apps. Their protection is relatively straightforward.

6.1 Copyright protection

Copyright protection is applicable to logos and icons under the Berne Convention as 'works of drawing, painting'[32] although they must also fulfil the 'own intellectual creation requirement'. National copyright legislation is a source of protection. For example, icons are protectable under the UK's copyright regime and in the Netherlands as an 'artistic work'. Simple logos and icons, such as the depiction of an abstract house as the 'Home' icon, would probably not fulfil copyright requirements. The IP protection of fonts varies between jurisdictions. Fonts may be explicitly protected by copyright, for example in the Netherlands, and also in Germany where they are protected for fixed periods.[33]

6.2 Design rights

Under the CDR, any industrial and handicraft item is considered to be a suitable product subject for a design. Graphical symbols and typographic typefaces are expressly listed in

[32] Berne Convention 1886, Art. 2 (1)

[33] The German 'Schriftzeichengesetz' provides an initial protection for typefaces under copyright for 10 years. This period may be extended (for a fee) once by another 15 years. After the expiration of this period (10 or 25 years), a typeface can still be protected when digitized into a computer font, as a computer program. In The Netherlands, typefaces could be protected under 'normal' copyright if they fulfil the requirements for that (the typeface must be an 'own intellectual creation').

the statutory definition as examples for such products.[34] Icons are covered by the broad notion of graphical symbol.

6.3 Trademark protection

The rules that apply to the analogue world are equally applicable to obtaining a trademark in digital form. Just as in the analogue world, if a logo or icon is too simple to guarantee distinctiveness, it will fail the requirements for trademark protection.

7 Concluding Remarks

Developing software and implementing that software on a mobile device, like a smartphone, is not identical to developing and implementing software on a desktop. While certain aspects of software development and implementation for an app overlap with those of desktop software, there are differences. It is useful for the app developer to be aware not only of the more general legal approach to IP protection for software, but also particular points of concern for protecting software run on a mobile device.

As patent protection is less relevant to most app developers, given the existing standardisation underlying the technical functionality of many apps and the fast-moving nature of the sector, the focus in this article is on the other forms of formal IP protection: copyright, design and trademark. By separating out the different components of apps, an analysis has been made of the relevant IP possibilities for each of those components. The important question is: which form or forms of IP protection offer the app developer the best and the most cost-effective form of protection?

With respect to the IP protection of source code, copyright will be the sole option. However, app developers have to be aware of the possibility that updates and other modifications to the source code might be carried out by third parties and that certain software components used to build the app might have been developed by third parties, who in turn may have IP rights on those changes or building blocks. This can be particularly critical for apps developers, given that users expect swift and frequent updates. It is therefore important that app developers keep an eye on which versions of the source code they use and which parts of it represent their own intellectual property. Depositing the code either with a legal professional or the BOIP can be a useful means to prove who was

[34] Council Regulation on Community Designs No 6/2002, Art. 3 (b)

the author of the code, and therefore the copyright owner, as the code was formulated at the date of the deposit.

Various forms of IP are relevant with regard to the GUI. Copyright and (unregistered) design rights are practical options. Possibilities for trademark registrations have widened with the implementation of the Trademark Reform Package. Although registering a trademark in this context is certainly not without its difficulties, it may well be worthwhile. A trademark, if renewed, does not have a limited duration, unlike other forms of formal IP. This in itself could be very advantageous.

Movements and transitions that form vital components of an app are probably best protected by design rights. When registering these aspects for a design right, care must be taken to make the movements or transitions clear, for example by including animations in the application. As is the case with GUIs, trademark protection will in certain cases offer an alternative form of protection.

Logos, icons and fonts, also like GUIs, can be protected by several forms of IP: copyright, design and trademark protection are all possibilities. However, if the feature is too simple to qualify as a ‘work’ it will not have access to copyright protection, nor will it enjoy trademark protection if the feature is not sufficiently distinctive. On the other hand, in the EU design rights explicitly protect typographic typefaces and graphical symbols. Icons are covered by the broad notion of graphical symbols and so will usually enjoy protection.

There are still some grey areas of law as far as software protection in Europe is concerned. To date, there is relatively little case law specifically on apps. Yet if an app developer knows how to use the various forms of IP as they relate to the components of an app, it is possible to achieve a significant level of protection. While it is likely that the GUI as a whole cannot be protected, IP protection of separate components of the GUI may be sufficient to ward off infringements by competitors. From legal practice, it can be observed that simply registering IP, regardless of the scope of the protection, can in itself act as a persuasive deterrent to would-be infringers.

If the most is made of the IP protection that is available, then even in the digital environment businesses have sufficient options to protect the identities they have built up and their digital presence. Apps are relatively new, at least in the realm of legal doctrine and case law. Time will tell which IP choices will be the most reliable and most effective for the app developer.