Cultures of independent game production: Examining the relationship between community and labour.

Orlando Guevara-Villalobos
University of Edinburgh
George Square
Chrystal Macmillan Building
+44 07551814401
s0897626@sms.ed.ac.uk

ABSTRACT
In this paper I aim to show the process in which independent game development is being shaped and leveraged by communities of developers. Despite digital distribution channels and the emergent markets configured around a new generation of mobile and online platforms, *indie* developers still struggle to develop creatively controlled games mainly by the means of more or less compromising sources of funding. Within this context, I argue that experimentation, user testing and feedback, exploration of ideas, skill acquirement, collaboration and moral support within *indie* communities are crucial elements of the process of game development. These features constitute the very nature of the events organised by communities, providing a series of emotional, cognitive and practical tools to deal with changing markets and work conditions. In sum, I suggest we might be witnessing the configuration of *communities of production* as a means by which developers seek to regain creative control over of their own labour.

Keywords
Production of Culture, Independent Game Development, Indie Culture, Communities of Practice, Digital Games Industry

INTRODUCTION
Since the arrival of digital distribution channels and the creation of virtual marketplaces, video games developed by independently-funded game studios have experienced a major growth during the last decade. Independent studios comprise a varied range of creative workers, from one man companies burning their bank savings accounts to small studios using contract work models to fund their projects. In order to fulfil their goal as independent developers, studios need to play with *neo-fordist* strategies of global capitalism, leaning on flexible labour, constant re-skilling, and work fragmentation as a means to keep competitiveness and to some extent, creative control. Nevertheless, their relatively isolated endeavour entails a series of obstacles implied in the knowledge and design intensive nature of game work, from the basic means to develop a game to the motivational aspects of working individually or in very small teams. The strategies to

Proceedings of DiGRA 2011 Conference: Think Design Play.

face social and technical challenges have led to the intensification of community networks and in some cases, the creation of spaces of development within them.

As it has been witnessed within a whole range of creative industries (Sennett, 1999; McRobbie, 2000; Kennedy, 2010), independent game networks and communities have become a key factor in tackling the social dilemmas of the organisational fragmentation, labour flexibility and adaptability to new markets. These communities are organised around the practice of game development and the passion unleashed by the nature of the tasks entailed in it. As any other social group, communities of developers share a series of interests, beliefs, experiences, emotional load and common practices that are self-referenced. Even more, they have become deeply entwined in every single stage of game development. As we will see, places of development converge with communitarian practices, and communitarian events have become spaces that boost creativity, learning and organise actively work time.

Working on those lines, the following paper aims to describe briefly the sociotechnical features of game development as carried out by independent developers, the ways they harness virtual and local community events to leverage game production, and finally how community interaction provides an *ethos* within which developers create shared meaning, emotional support, but also a pool of skills and knowledge key to develop successfully in the more recent platform markets.

METHODS

The following study shows some findings of a PhD research fieldwork, carried out between January-December 2010. Relying on a series of ethnographic methods (semi-structured interviews, diary research, and participant observation), I interviewed 23 independent game developers based mostly in the UK, succeeded by follow-up questions through e-mails.[*] Their ages range from 24 to 36, with one exception (48). For confidential issues, interviewees' names mentioned here have been changed. Moreover, interviews were complemented with an analysis of developers' weblogs, promotional interviews, and visits to a series of activities organised by the communities of developers they are involved in, featuring weekly meetings, networking events/conferences and game jams. Last but no least, this study was enriched by trade publications, and online press articles, which serve useful to identify and contrast general perspectives on independent game development.

RESEARCH BACKGROUND: COMMUNITY AND WORK.

Since the early 1990's, neo-marxist scholars within the cultural/creative industries field of studies, have addressed the existence of new forms of labour carried by professionals of the *new economy*. This kind of immaterial labour is the one that produces immaterial products such "as knowledge, information, communication, a relationship or an

[*] From the 23 interviewees, 2 live outside the UK. Nevertheless, they were included in the research as the connection between both them and other interviewees was quite strong.

emotional response" (Hardt & Negri, 2004: 108). As Lazzarato (1996: 133) has pointed out, the concept refers to two specific dimensions of work,

> "[O]n the one hand, as regards the 'informational content' of the commodity, it refers directly to the changes taking place in workers' labor processes . . . where the skills involved in direct labor are increasingly skills involving cybernetics and computer control (and horizontal and vertical communication). On the other hand, as regards the activity that produces the "cultural content" of the commodity, immaterial labor involves a series of activities that are not normally recognized as 'work'—in other words, the kinds of activities involved in defining and fixing cultural and artistic standards, fashions, tastes, consumer norms, and, more strategically, public opinion."

Immaterial labour takes places in a post-fordist context where innovation driven and deregulated economies lead towards the adoption of flexible and outsourced work (Kline et al, 2003; Watson, 2008). Here, theorists have identified a process of casualization of work, where production is moved from the traditional workplace to other social spaces (Terranova: 2000), leading to an economisation of daily life (Pongratz, 2001), as creative workers and consumers within the games industry become freelancers and creators of content. Isolated and dis-unionised, scholars point out creative workers' precarious dependency from global flows of capital. Yet the present study suggests this relation is not without its nuances. There is still room to look at the strategies creative workers (in our case indie developers) deploy in order to find alternative ways for game production, dealing with both financial and motivational problems. It is in this thin and blurry line within contesting agencies where this paper takes place.

The above leads me towards the analysis of independent game work as a communitarian practice. A growing literature in learning, business management, economic sociology and cultural geography has underlined the relevance of informal forms of organisation at work, theorizing about relevance of trust in networks (Coleman, 1986; Ettlinger, 2003) or the creation of informal organisational cultures based on shared occupational interests (Lave & Wegner, 1991) and their impact on innovation and growth within companies (Brown & Duguit, 1991; Wenger, 1998; Kimble & Hildreth, 2004).

These *communities of practice* would come to describe the relationships within a group of individuals defined by their joint enterprise, mutual engagement and shared repertoire of communal resources (Wenger & Snyder, 1998). As stated by Foray, communities are growing to turn into an "organizational system allowing the exploitation of virtuous properties of knowledge" (2004:182), being the centres of production and reproduction of it.

As Calhoun observes, communities nowadays resemble more "dense, multiplex, relatively autonomous networks of social relationship." (1998: 391) In fact, communities of independent developers are inter-embedded networks, each one providing a space (virtual and/or physical) where they construct shared meaning and trust, but they also negotiate the construction, distribution and transference of hard and soft assets (skills, knowledge, code, work practices), led by their passion for game design[1]. As a result of the histories of precariousness in the game industry and the creative passion for games as a medium, I suggest we are witnessing a reconfiguration of some knowledge and practice communities into communities of production, an autonomous project worth paying attention.

Communitarian practices are deeply embedded into the process of game development, providing an organisational framework witnessed in the same process of indie game production. In addition, communities provide their members of emotional attachment and meaning that result in strong bonds of moral support and trust, providing motivation to otherwise isolated and futile efforts of game development, but also they become an informal mechanism to cope with the organisational needs derived from work fragmentation by enabling networking practices, knowledge transference, and skill learning. Communities of developers "can help with the creative sparks, the play testing of games, and how to handle the business aspects of selling the games. Really all sorts of things at all levels." (David, email interview, August 16, 2010)

CULTURES OF GAME PRODUCTION.

In order to understand where communities of indie developers stand in the process of game production, it is essential to at least describe roughly developers' socio-technical approach to their work. Although the terminology can vary, it is very common to describe the process of development in a series of well-defined stages featuring for instance Design and Prototype, Pre-production, Production and Testing (Kerr, 2006). Nevertheless, independent developers carry out a different approach in which activities like experimentation, prototyping and testing are fundamental activities throughout the process of development.

I suggest looking at the process of independent development as a two phased model. A first stage describes the creative process of defining a game project, by the means of a constant experimentation and testing of ideas, while a second stage of development addresses an iterative process of prototyping, where developers start working with basic game features and artwork, adding more complexity and redefining game features through a series of prototypes.

Experimentation and pre-prototyping

As most developers have asserted, game ideas are inspired from anything happening in their daily lives, from reading a book to personal experiences. Sometimes these ideas are sketched out in the design document where developers break down a game in its basic mechanics and objects as well as the software design. Nevertheless, behind a game design document there is a whole process of creative experimentation. The notion of "messing about" with technology is very important as usually experimenting with programming languages, 3D software or game making applications and reflecting on their use converge with the process of coming up with and sketching game ideas[2]. As stated by Soul, toying with technology and game ideas can feed off each other:

> "…either I have an idea, or I'm messing about with some tech and from there it will mull in my head for an hour or so, and usually at that point I'll have an 'maybe if I try this' type idea, at which point I try it, and then that usually points the way for the next thing to try. Usually after a day or so I'll have a prototype or base control for a game. Sometimes it's fun already, sometimes it isn't and I drop it." (Soul, e-mail interview, September 16, 2010)

When developers come with a feasible game concept, and of how that idea could be carry out according to their knowledge, they might start a more specific planning with a design document where they specify in more detail the different tasks needed to reach the completion of their game. Still this practice is more common amongst micro studios, being more personalised for self-employed developers. As stated by Rob

> "I really don't. I have no sense of planning at all. The most I get is a couple of pages of A4 with some drawings, but there is never… most of it is in my head. I can't set deadlines for myself, that's just doesn't work, I crumble under the idea of being organised." (Rob, personal interview, May 18, 2010)

Iterative development (prototyping)

Most common among independent developers is the work towards a playable prototype for their game, from which a succession of iterations will derive around the already developed features and the addition of new ones. This cyclical process is borrowed from a software development process called spiral model (Boehn, 1986; Rollings & Morris, 2004; Iuppa & Borst, 2010). It stresses on rapid prototyping in order to assess the functionality of a product. Nonetheless, this model tends to be more complex and heavily managed for the relative small projects developed by indies and their working habits. The phase of understanding requirements and system design is normally flexible and open to changes. Keeping the spiral principle, but adapted to personal needs and styles, indie development matches with what Rick has defined Evolutionary Delivery,

> "we are trying to get a complete working something very early on, we had a prototype of this game going within a week, and it was kind playable, showed it to people it and make them see what the game was like, and from then on it was just a matter of improving and refining until you run out of time or until you decide it is enough" (Rick, personal interview, August 24, 2010)

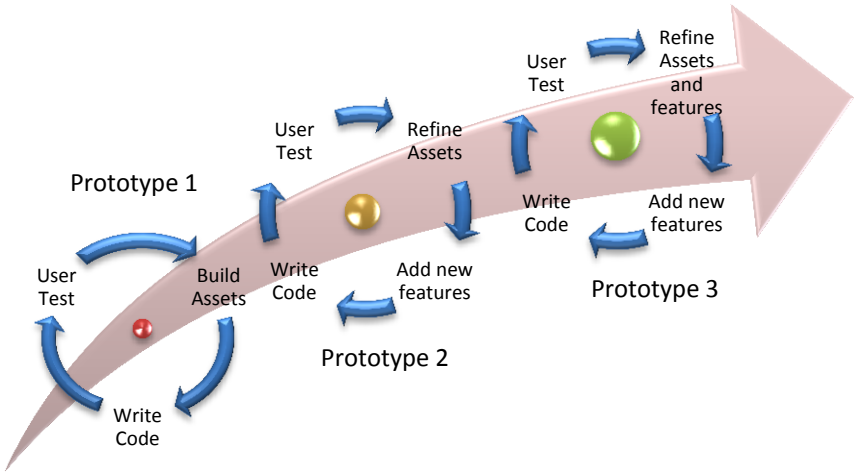

Figure 1: Inside the iterative process of game development

As we can see in figure 1, the model contains in itself the mechanics of coding in game development in which developers edit, compile and run code. In a similar fashion, the whole process of development of each prototype would undergo a three-phased process: working on a prototype (edit), building a playable version (compile) and subjecting it to player testing (run).

As stated by developers this scheme of work keeps game development as a reachable enterprise, allowing them to work more organically, making any planning easier and allowing the developer to make "bold decisions" as they reflect upon the feedback received by testers (Scott, interview, August 31, 2010).

COMMUNITIES WITHIN GAME DEVELOPMENT

Networks of collaboration and knowledge can be traced back to the consolidation of Silicon Valley's horizontal organisation (Saxenian, 1996), and the collaborative projects featured by free software organisations during the 1980's (Sennett, 2008). These networks have informally existed as part of software and game developers identity, becoming each time more decisive for the success of game companies, and a strong driver of game production.

Hitherto, communities of game development had comprised mostly hobbyists and professional developers who worked on their projects as a free time activity. Nonetheless, since independent development became possible and profitable as low-budget projects, indie communities turned into a basic infrastructure to substitute the highly specialised and expensive processes of AAA games.

Currently, these networking practices describe the relationships found by Wittel (2001) and Kennedy in new media workers (2010), where precarious and unstable conditions of cultural industries lead to "a kind of informal, voluntary professional collectivity" (Kennedy, 2010: 198). Nonetheless, meeting indie colleagues, sharing information, knowledge and code as well as leveraging game development are practices that help to the construction of stronger communitarian bonds.

Informational Networking

Indie communities as networks are well known for their ability to connect professionals involved in game development, as their aspiration for game publishing has led them to find collaboration in areas where their expertise is limited, or the work too encompassing. Either online (through community fora) or in physically located events, developers keep constantly engaging into conversations, expressing their professional interests, field of work and exchanging information about the industry. By doing this, they get to know freelancers, collaborators, and potential team partners.[3] For instance Kris tells us how this works: "we got freelance audio guy, we got him doing some audio for Subversion right know, and we actually met him at that Cambridge indies event, because he was working with one of the other indies." (Kris, skype interview, August 27, 2010).

Furthermore, developers' constant engagement through networking or community events, enable them to exchange information and knowledge about new technologies, business opportunities, marketing strategies, as well as advice about outsource work, and how to

deal with contractors. Evidently, these knowledge and information seeks to find cost-effective and profitable methods to develop and publish games:

> "Because of camaraderie, people are very happy to share experiences and knowledge, and that knowledge is very valuable. You know, when we came to do self-published. We had no experience in publishing or whatsoever, ever. And we got a fantastic exposure and we sold 100.500 copies at £4… £5 in some cases. And the only way I could do that is by knowing from other people what their experiences were." (Karl, personal interview, July 15, 2010)

Seemingly, developers also leverage their daily work by being constantly in touch with their peers. By the means of web communication and physical interaction, communitarian bonds enable knowledge transaction when technical obstacles emerge in the process of game making. Even more, in cases like the Cambridge Friendship Club, this interaction is constant during their weekly events:

> "We asked [Scott] how he was using the Unity user's interface and he was doing it different from us… then we switched to use the interface in the same way, because we had the same difficulties in it. It was useful to know, it was working quite nicely and that helped us to make decisions to make a switch, so that's a specific technical issue. Other times we got a problem that we had found a blank frame in the game and we said "have you come across that" and he said "no", but if he had and he solved it, that would work out, that sort of things." (Rick, personal interview, August 24, 2010)

Sharing work

Within indie communities and networks, code sharing is a defining feature of game work. It fulfils different purposes, as it is both the product of the cultural *ethos* of the Web and a learning practice.

Throughout the internet, independent developers benefit from the shared works done and published for free by hobbyists and other indies. These works include game assets (art and audio libraries, applications, engines) but also games' source code. Open Source initiative Pygame is good example of the above, where art, music assets and game engines created by community members can be downloaded without cost, while hyperlinking to other websites where more assets can be found. Seemingly, Steward is a developer who always has a shared link to his games' source code in his webpage. More radically, Rob allows and enthusiastically invites others to use his works as the base for their own projects (Rob, n.d.). The networking practices described above basically help developers to catch up with these online sources, creating a complex network of asset's supply.

Furthermore, as Kennedy (2010: 199) suggests, sharing code is also a learning technique that keeps developers up to date and improving their craft. Rick, while commenting the pros of the indie community at Cambridge, he states: "Terrence sent me his Flash framework because we got interested in how it works and we would be willing to share code with other people" (Rick, personal interview, August 24, 2010). Code sharing is not

just exchanging work, but it is also a practice of 'opening up'. For instance, as part of their activities during the weekly meetings at the Cambridge Friendship Club, developers show their code to others to either explain how they tackle a problem, or to work it out in group.

Nonetheless, sharing code is not without controversy. Assets and code are a product itself of game work, and constitutes companies own commodities. As stated by Karl "code has a tangible cost. So it is right that if you give some code to somebody, he should then pay something for it." (Karl, personal interview, july 15, 2010). In addition, some independent companies address the indie turn as the re-appropriation of their own IP as a tool of empowerment within the industry relations of production. Nevertheless, these assets are normally outmatched by the mainstream industry hi-fi neurosis, enabling companies to compete in a small scale with affordable prices for independent studios. The difference with other developers is definitely the scale of their business and the quality of their work. Whereas companies with high skills in programming tend to have a better structured code written, other developers do not consider theirs good enough to be monetised, but a shared base that can be taken by other developers to build upon it.

Experimentation, testing and feedback.

As stated in the previous section, developers such as Kris, Soul, Scott, Terrence and Andy exposed their strong inclination towards a creative first stage strongly based on experimentation in game mechanics, and in other cases interactive mechanics around storytelling.

These practices far from being individual efforts, they are generally fostered by the interaction of developers in a special activity organised as a community: the game jam. Game Jams are small competitions organised by different networks or communities of developers where the participants have to rapidly prototype a game in a short but variable period of time. Global Game Jam and Ludum Dare are 48 hours competition, whereas the Indie Game Jam invites to write a game in 4 days. In addition, organisers establish a theme or restriction as a problem to be solved. For instance, developers might have to create a game with the theme "total darkness", or to design a game that would only use the keys "1, 2, 3, 4".

There is consent about what game jams are for within the process of development. Presumably two of the first game jams initiatives since 2002 (Indie Game Jam and Ludum Dare) aim both to "encourage experimentation and innovation in the game industry" (Kris Hecker, n.d.) and to create a space where people can "make the time to create a game prototype for [themselves]."(Ludum Dare, n.d.). [4]

Summarised, game jams are about opening creative spaces where developers can explore their ideas. Interestingly, their dynamics match perfectly with the way independent developers reflect upon their production. Developers use these activities as the perfect lab where to test and give form to their game ideas. Thus, they enhance and dynamise the pre-development process by trying out and throwing away game mechanics: "I mean one lesson about the jam is that it forces people into thinking very fast and quickly realizing their idea, and sometimes those ideas work or they don't, but you often take away those ideas and make new ones based on those ones." (Scott, personal interview, August 31, 2010) Furthermore, game jams enable developers to try software and design techniques.

As strong the game idea can be its development and playability also depends on the technology and the techniques used to create the game experience. This leads developers to a constant experimentation of techniques and technologies that eventually will be used to convey the artistic vision of the developer.

Another important characteristic is the way the community supports the iterative mechanics of game development, present in game jams, weekly meetings and/or simply through developers' websites. For instance, Ludum Dare games are always assessed by the participants, who provide feedback and decide by voting the best games of the event. Moreover, Terrence comments how he uses his webpage "…to post playable builds of games way before I finish them because I want to get a little bit of feedback […] and very often they can point out very basic things they can see and I missed" (Terrence, personal interview, August 24, 2010). Likewise, developers make the most of their weekly meetings by showing their prototypes and discussing them with their peers, in words of David, these spaces are a "…fun time to demo your game or take a look at some of the game assets work in progress, […] or demo a game that is almost ready for release to the group at large" (David, email interview, August 16, 2010)[5]. As for the Cambridge Friendship Club, Rick comments that "showing [your game] to game developers, that's gonna give you specific feedback of 'that's gonna sort trip you up' or 'in my game, I did that and I got those problems'. That is a useful kind of feedback you get from people." (Rick, personal interview, August 24, 2010)[6]

As we can see, by interacting with the community, indie developers energize the iterative cycle of development. Technical and user's feedback, testing technologies, ideas and prototypes, or simply 'playing around' set up the creative and technical conditions to succeeding in developing a game.

Community and self-management.

Autonomy is a highly cherished condition by developers who in turn carry the responsibility to manage their own work. Managing time can become an important issue for some developers in terms of getting work done. Interestingly, community events and networked blogs help to structure developers' approach to their work.

Although some developers like to keep a more tangible separation between both, others prefer to harness game jams to get work done. As stated by interviewees "if you have an application with a purpose for already, and you are having a deadline, then that [game jam] can force you to design very quickly and to think on new feats and kind of meet big pot systems" (Scott, personal interview, August 31, 2010). While some developers use these spaces for experimentation, other ones use the time to work on their projects and make as much progress as they can.

Furthermore, an interesting topic often addressed by developers in their blogs is related to work management. Some of them address their concern about how they organise their work, as well at its flow across time (Nyveldt, 2011). In addition, it has become common discuss ways to balance work and life (Goss, 2010), or work taking care of your children (Goss, 2010). At the end, although they like to work in their own ways, indies feel the need to express their concerns about their profession, and to find with their peers common solutions to those problems.

***

In perspective, Independent games as a final product are highly shaped by the communitarian practices mentioned throughout the section. Celebrated games like World of Goo came up by 2DBoy developers experimenting in game jams. As for my interviewees, most of Soul and Steward's games published in their website are product of game jams. Seemingly, Terrence's most famous games have emerged from his constant participation in community events.

GAME DEVELOPMENT WITHIN COMMUNITIES

The efforts to understand the process of developing independent games have shown us that far from being an individual endeavour, it is the collective result of the complex interaction among developers, other industry actors within the chain of value, and communities (both players and developers). Although the enterprise has proven profitable for some developers, many of them struggle to finance their games within precarious economic conditions, triggering a quest for assistance. In the past section, I talked about how independent developers have used communitarian interaction to leverage game production. Yet there are other aspects of indie communities worth noting, as they emerge from developers' activities within a community and play their part in facilitating game production. These ones are the affective nature of indie communities, and the process of learning a set of skills and know-how that becomes useful when developing a game.

Motivation, trust and support.

As many scholars have pointed out in the past, either online (Weisband, et al, 1995; Wellman et al, 1996) or through physical interaction (Brown & Duguid, 1991; Wenger 1998), the motivational driver to the formation of communities or networks can be found in shared interests. These communities of interests express a domain that orchestrates members concerns and passions (Wenger, 1998). It is clear to me that independent developers' passion for games as well as shared isolation, trigger a need for contact in an (new) enterprise where complex sets of specialised knowledge and skills are hard to acquire by oneself or a small self-contained team of developers.

The motives driving community interaction within indie developers are from the most varied nature. Although there seems to be tangible goals regulating this interaction (information, knowledge, team building), there is also an emotional need and an identification as "indie" (which in itself is a culturally and politically effervescent identity) triggering community interaction. These *multiple rationalities* (Ettlinger, 2003) are socially intertwined and reinforced through collaborative work, moral support and trust building within these communities.[7]

At the Cambridge Friendship Club, Rick defines the *ethos* of the community as follows:

> "…people aren't too precious about things they have… they are not "that's mine!", we would be more like "look, we're all exercising our ability to make things". By jamming, by sharing everyone's maximizing their creativity. So, everyone is confident about doing stuff. So why be worried about how something

> we made yesterday was boring… that's not the attitude, it's more like "hey I can do another game". So everyone is kind and free with their ideas and not too precious about it, and that's really nice. (Rick, personal interview, August 24, 2010)

Reciprocal relationships and shared experiences as a community of practice and production create the conditions to develop trustful bonds, 'reinforce common values, goals and practices' (Pargman, 2005: 106). By interacting as a community, indie developers acknowledge their capabilities, allocating synergies of performance among them. For instance, Terrence is considered to be very creative and prolific, while Scott's expertise in Unity 3D and creative ideas have helped developers to experiment and improve their skills. Interestingly, the community relationships do not seem to develop any hierarchy based on capabilities, but in turn developers highlight the fact that everyone deal together with the process of problem solving and problem finding: "Kris over there is dealing with the same problem I am dealing right now. We work on it kind of separately but we also discuss and try to figure things out together. It's been quite helpful in that way." (Terrence, personal interview, August 24, 2010)

Furthermore, emotional attachment is also developed as indies are constantly "looking after one another" (Scott, personal interview, August 31, 2010). This feature of communities of indie developers was perhaps the most present in every interview, showing the emotional need to communicate and make contact beyond virtual means. David comments how often weekly meetings turn into a "talk shop" about the trials and tribulations of the business, while Scott asserts that "in terms of the local community a lot of it is actually like moral support […] also is kind of good to have people to whom to lean on for advice" (Scott, ibidem). Physical interaction, as we will see in the next section becomes highly appreciated in the process of game making, but here it is important to highlight the fact that it is portrayed as a facilitator to convey feelings and support as it opens new dimensions to strengthen developers bonds.

At this point we can ask why this is important in terms of game development. As most of the interviewees conveyed, community interaction, trustful relationships and the support developers receive from their peers work as a powerful source of motivation: "Sometimes frankly you really need people around you working, like here Terrence working in his stuff right now. That's motivating, that's actually encouraging." (Scott, ibidem)

During the weekly sessions, developers at a local café keep encouraging each other by discussing topics about game design, technologies available, as well as art and programming techniques. Stories and news about games, companies and other shared likes are also part of their basic interaction. As a community, they show a special empathy when a developer has a problem with a project or a more personal matter.[8] These bonds make activities like game jams a special moment that reinforces the idea of making games, and the wonderment of finding through coding, assembling and testing a representation of their ideas. Motivation, trust and support within indie communities emerge in this process in which developers acquire the group's subjective viewpoint, learn to speak its language, and set the foundations to learn from each other.

Skill acquirement

The effects of communitarian activities among independent developers go beyond shared meaning. Different paths into the games industry, the constant development of new tools, and a still dubious formulation of formal training for game development entail a serious dilemma for developers. However, trustful and localised interpersonal networks or communities create a learning space for the transferring and developing of [tacit] knowledge, common practices of production, and innovation. For instance, game jams seem to tackle aforementioned obstacles by helping developers to refine their craft, and learn a series of skills that smooth the process of development.

As Sennett (2008) states, a skill is simply a trained practice. It can comprise the experience, qualifications and ability used in making something, or the experience and ability formally required by the nature of the job (Cockburn, 1983). Moreover, a skill can only be developed by a practice embedded in routine. It is by organised repetition that a person can train, develop and eventually master a series of skills, from which he/she can start expanding even more.

Game jamming as an activity follows this process in detail. For the same reason why a game jam can be used to leverage development, it also helps us to understand its relation with skill acquirement; in a game jam people are learning and exercising the process of game development itself, notion known by Arrow (1962) as learning by doing, where learning is a by-product of production: "[N]othing has helped me understand what goes into making a game better than the time I've put into rapid development events" (Soul, e-mail interview, September 16, 2010). Even more, the idea of figuring out the game ideas and how to represent them digitally in a fun product is conceived by developers as fun in itself.

As for the skills mentioned by developers, they stand for every single aspect of the process of development. Rapid prototyping, help developers "to exercise some strong design principles and some strong decision making that you would otherwise not necessarily do" (Scott, personal interview, August 31, 2010). Kris also stresses on the effect of game jam in game design: "it is actually a very good practice, because sometimes you can get too attached to these very big epic ideas and going and making a game in three hours actually forces you to zoom in on important gameplay things." (skype interview, August 27, 2010)

Nevertheless, ideas are still important for independent developers, and there seems to be a strong consciousness of how game jams fosters creativity: "…one lesson about the jam is that it forces people into thinking very fast and quickly realizing their idea, and sometimes those ideas work or the don't, but you often take away those ideas and make new ones based on those ones" (Rick, personal interview, 2010). In game development, the ability to come up fast with game ideas and test them opens different opportunities for developers, and helps them to assess a game in terms of their possible impact on players, without getting too attached to them.

As I mentioned in an early section, experimentation is inherent to game jams. When the subject of experimentation is a piece of software, the know-how of the community and the training grounds are joined in the act of jamming, also known as learning by using (Rosenberg, 1982). Hence, developers' expertise in the use of certain technologies is enhanced as result of its constant use. This process goes beyond domesticating

technology as developers figure out ways to "cut the corners" as well as effective procedures to maximise the use of certain tool.

Skills like the ones mentioned by Kris and Terrence have a serious impact in game development. Industry-wise, AAA games set the dominant conventions in game making. Realistic simulations and cutting-edge technologies support the generation of tenths of hours in game content and highly specialised art design. Nevertheless, new platforms (Iphone, Flash games, hand-helds) entail new conventions in game making, given their limited performance, storage and variable interface. In sum, game jams are a learning space whose rules and dynamics lead to the development of the necessary skills to take full advantage of the conditions of the new platform markets.

Production and Transference of Knowledge

As it happens when learning skills, relevant knowledge for developers is usually "rooted in the flow of practice within communities." (Duguid, 2005) In independent game development, communitarian organisation is turning into a primal structure that provides the means to access and produce knowledge. I have already outlined how networking is a common practice among communities, channelling information and knowledge across developers through events, weblogs and fora.

Furthermore, communities are not a structure that simply allows the flow of information. Rather, within them developers produce, share and give meaning to that knowledge (Wenger, 1998; Brown & Duguid, 1991). As a result of community interaction, developers construct what I could call a pool of knowledge, a deep understanding of the process of production, the mastering of tools and the different ways to frame a problem in order to handle it (for instance, the transmutation of a game idea into software architecture, or the way to program an AI in order to run realistically a race car in any environment).

Within this pool of knowledge, indie developers create a common understanding of the games industry. Reflecting upon working practices and predominant –artistic, design, technological- visions of game production, developers create meaningful ways to deal with the production pipeline. As stated by Rick, Charlie and other developers, interacting with other indies informs them about new technologies, potential market opportunities and common problems in game production. As David mentioned, most of the time, they share and discuss game *post-mortems,* giving an understanding of the problems arisen in the project from the process of development to the marketing and distribution stages, enabling them to discuss and learn from it.

Seemingly, developers discuss and/or try technical procedures and reinforce their knowledge of certain technologies, which in turn enable them to develop skills further on. Particularly during local game jams, developers constantly evaluate, try and change their approach to a game idea or challenge in a constant loop of *trial and error*. As a common action to perform, developers visit their neighbours during a break and inform themselves about others' projects and their working progress. There, they discuss the methods and tools employed to achieve an idea within a game, keeping a mutual feedback throughout the event. Thus, developers collaborate with each other and create in the process the *know-how* that can be used to solve future challenges.

CONCLUSION

This paper focused on the role of communities in providing an alternative infrastructure to tackle the challenges of independently-funded game development, but also a meaningful space in which developers learn and construct a valuable skillset to perform successfully within virtual markets. Furthermore, especially through the example of the Cambridge Friendship Club, we could see how community interaction within indies is providing developers with the emotional and motivational strength to fulfil their work. By doing this, developers counteract the experiences of isolation, anxiety and lack of motivation commonly found within the creative industries. All these suggest that albeit new platform markets and digital distribution set up the base from where independent games started to thrive, changes in other spheres of game work help us to comprehend how indie game production is developing its own sustainability, such as new forms of work organisation and the sources of emotional attachment and knowledge.

ACKNOWLEDGMENTS

I would like to thank Nick Prior and Kate Orthon-Johnson for their comments on this work. I am also very grateful to all the game developers that collaborated with their professional experience to this research. Their contributions are at the heart of this work.

ENDNOTES

1 Interestingly, these practices were not necessarily originated by instrumental needs subdued to production but simply as a part of the *ethos* of game developers, strongly based on collaboration and community strengthening (Bowen & Deuze: 2009). It was not until myriads of game developers in precarious conditions, hobbyists, amateur developers, art designers and students met digital distribution and new platform markets, when communities became an important infrastructure that now is redefining the experience of work fragmentation and flexible work.

2 This constant "learning by doing" (Arrow, 1962; Foray, 2006) is a key habit and need for indie developers. The availability of countless feasible technologies for game development, the search for technologies –as Rob claims- that best match their personal mindset, conceptual art style and their need for constant update keep developers trying out new libraries, techniques, applications for games and new software in general. This habit becomes the personal dimension to create a pool of tacit knowledge (Polanyi, 2009) that boosts the process of coming up with game ideas and experimentation.

3 These forms of collaboration are usually based on skillsets, creative interests. Developers can simply agree for a graphic designer (or musician) to provide some assets for a project, according to the formers creative vision. Nevertheless, when both them acknowledge each other work and share a common vision, possibilities to team up in a partnership with a more even creative input are likely to happen. As Terrence told me, for one of his games he just needed a quirky retro music, and he knew someone who could compose it just right. Conversely, by the time of the interview he had just started a project with John, whose works he considers have dealt a big impact on his vision. Although they split up their work according to their skills, they both contribute evenly with game design and subject the artistic vision of the project to discussion.

4 These game jams work within the boundaries established by the community of developers who attend them regularly. Nevertheless, more than identify them as communities by themselves (very common among developers) they tend to be community clusters chained by a broader community of game developers who constantly participate (either in group or individually) in each of these activities. For example, Terrence normally takes part annually of more than dozen of those events, organised by different groups: Ludum Dare, Global Game Jam, World of Love, Tech of the Month and Indie Kombat.

5 These face to face interactions present very interesting implications about the dialectic of communities as a set of virtual and physically located networks. Although developers like Terrence, Steward, Scott and Kris stressed on the importance of being part of a virtual community of indie developers when events such as conferences take place, the need to transcend CMC and establish physical contact with developers is key to establish a stronger bond and trust in order to share, give support and more importantly, to show your projects and to be subject of criticism.

6 Presenting and discussing game projects is an ever-present subject in indie game meetings. I verified this during my visits to the CB2 Café where the Cambridge Friendship Club meetings take place. In addition, they have organised a formal "show and tell" meeting on the first Tuesday of every month where developers are encouraged to present their projects in order to start an open discussion about them.

7 The construction of the indie developer's identity is a very complex process, tailoring members' biographies, narratives of the "mainstream" industry, and the construction of political, social and aesthetic meaning of indie games through practice. Space constrains deter me from treating this process in depth, as I my aim is to link the role of communities in game development as a "workshop" that provides affective and organisational assets as well as means of production.

8 Between conversations, some indies were discussing how good was for everyone to see Terrence moving to Cambridge. Seemingly, some chats revolved around the idea of convincing both Steward and Soul to move there and be part of the local community.

BIBLIOGRAPHY

Arrow, K. "The economic implications of learning by doing", in Review of Economic Studies vol. 29 (1962), pp.155-173.

Bowen, M. and Deuze M. "The independent production of culture: a digital games case study," in Games and Culture vol.4, (2009), pp. 276-295.

Brown, J. and Duguid P. "Organizational learning and communities of practice: Toward a unified view of working, learning, and innovation," in Organization Science vol. 2, no. 1 (1991), pp.40-57.

Calhoun, C. "Community without propinquity revisited: Communications technology and the transformation of the urban public sphere," in Sociological Inquiry vol. 68, no. 3 (1998), pp. 373-397.

Cockburn, C. Brothers: male dominance and technological change. London: Pluto, 1983.

Coleman, J. Individual Interests and Collective Action. Cambridge: Cambridge University Press, 1986.

Duguid, P. "The Art of Knowing: Social and tacit knowledge and the limits of community of practice," in The Information Society, vol. 21 (2005), pp. 109-118

Ettlinger, N. "Cultural economic geography and a relational and microspace approach to trusts, rationalities, networks, and change in collaborative workplaces", in Journal of Economic Geography, vol. 3(2003), pp. 145-171.

Fearon, R. (n.d.). *Abuse My IP, Make Games.* Retrieved from http://bagfullofwrong.co.uk/bagfullofwords/abuse-my-ip-make-games/

Foray, D. The Economics of Knowledge. Cambridge: The MIT Press, 2004.

Goss, O. (2010, November, 21). *The Distracted Programmer*. [Web log message]. Retrieved from http://www.streamingcolour.com/blog/2010/11/21/the-distracted-programmer/

Goss, O. (2010, October 10). *I'm Indie, and I'm Proud* [Web log message]. Retrieved from http://www.streamingcolour.com/blog/

Grannovetter, M. "The Strength of Weak Ties", in American Journal of Sociology, vol. 78, no. 6 (1973), pp. 1360-1380.

Hardt, M. and Negri, A. Empire. Cambridge: Harvard University Press, 2000.

Hecker, C. "Indie Game Jam IGJ0." Retrieved from http://www.indiegamejam.com/igj0/index.html.

Hildreth, P. and Kimble, C. Knowledge Networks: Innovation through communities of practice. London: Idea Group Publishing, 2004.

Iuppa N. and Borst T. End-to-End game development: Creating independent serious games and simulations from start to finish. United States of America: Focal Press, 2004.

Kennedy, Helen. "Net Work: Professionalization of Web Design", in Media Culture and Society, vol. 32, no 2 (2010), pp. 187-203.

Kerr, A. The business and culture of digital games: Gamework/Gameplay. London: Sage, 2006.

Kline, S. et al. Digital Play: the interaction of Technology, Culture and Markets. McGill-Queen's University Press: Montreal & Klingston, 2003.

Lave, J. and Wegner, E. Situated learning: Legitimate peripheral participation. New York: Cambridge University Press. 1991.

Lazzarato, M. “Immaterial Labour”, in *Radical Thought in Italy: A potential politics*, ed Michael Hardt and Paolo Virnio, (Miniapolis: University of Minnesota Press, 1996), pp. 137-147.

Ludum Dare. “About Ludum Dare.” Retrieved from http://www.ludumdare.com/compo/about-ludum-dare/

McRobbie, Angela (2000). “Fashion as a culture industry”, in *Fashion Cultures: theories, explorations and analysis*, ed Stella Bruzzi and Pamela Church Gibson (New York Routledge, 2000), pp. 253-263.

Nyveldt, A (2011, July, 26) *The organization of a indie developer* [Web log message]. Retrieved from http://www.nyveldt.com/blog/post/The-organization-of-a-indie-developer.aspx

Pargman, D. "Virtual community management as socialization and learning", in P. van der Besselaar, G. De Michelis, J. Preece and C. Simone (eds.), *Proceedings of the Second international conference on Communities and Technologies*, (Milano, June 2005), Dordrecht: Springer, pp. 95-110.

Polanyi, M. The tacit dimension. Chicago: University of Chicago Press, 2009.

Pongratz, H. and Voß G. “From employee to entreployee: towards a ‘self-entrepreneurial’ work?”, in Concepts and Transformation, vol. 8, no 3 (2003), pp 239-54

Rollings A. and Morris D. Architecture and design: A new edition. Indianapolis: New Riders, 2004.

Rosenberg, N. Inside the black box: Technology and economics. Cambridge: Cambridge University Press, 1982.

Saxenian, A. L. "Inside-Out: Regional Networks and Industrial Adaptation in Silicon Valley and Route 128," in Cityscape: A Journal of Policy Development and Research vol. 2, no. 2 (1996), pp. 41-60.

Sennett, R. The Corrosion of Character: personal consequences of work in the new capitalism. New York: Norton, 1998

Sennett, R. The craftman. New Haven & London: Yale University Press, 2008.

Terranova, Tiziana. “Free Labor: Producing Culture for the Digital Economy”, in Social Text 63 vol. 18, no. 4 (2000), pp. 33-58.

Watson, Tony. Sociology of Work and Industry. London: Roudledge, 2008.

Weisband, S. et al. "Computer-Mediated Communication and Social Information: status silent and status differences," in Academy of Management Journal vol. 38, no. 4 (1995), pp. 1124-1151.

Wellman, B. et al. "Computer networks as social networks: Collaborative work, telework, and virtual community," in Annual Review vol. 22 (1996), pp. 213-38.

Wenger, E. Communities of practice: Learning, meaning and identity. Cambridge: Cambridge University Press, 1998.

Wenger, E. and Snyder W. “Communities of practice: the organizational frontier”, in Harvard Business Review, (January-February 2000), pp. 139-145.

Wittel, A. “Toward a Network Sociality”, in Theory, Culture and Society vol. 18, no. 6 (2001), pp. 51-76.