Understanding the benefits of game jams

Exploring the potential for engaging young learners in STEM

Allan Fowler
Kennesaw State University
Marietta, GA
USA
afowle56@kennesaw.edu.edu

Johanna Pirker
Graz University of Technology
Graz
Austria
jpirker@iicm.edu

Ian Pollock
California State University
East Bay
Hayward, CA, USA
ian.pollock@csueastbay.edu

Bruno Campagnola de Paula
Pontifical Catholic University of Parana
Brazil
bruno.paula@pucpr.br

Maria Emilia Echeveste
Universidad Nacional de Córdoba
Córdoba
Argentina
meecheveste@gmail.com

Marcos J. Gómez
Universidad Nacional de Córdoba
Córdoba
Argentina
mgomez4@famaf.unc.edu.ar

ABSTRACT

There is a wide range of implementations of game jams throughout the world . Game jams have been organized in a number of different formats, themes, and timeframes [43]. What they all have in common is the opportunity for participants to make a game within a specified constraint such as time, location, technology, or theme. Additionally, game jams as social experience support active and collaborative learning formats. In this paper, we discuss the potential of game jams for young learners, describe successful jam events in this context, and provide a list of tools useful for organizing game jams for this target group.

CCS Concepts

•Social and professional topics → Computer science education;

Keywords

Game Jams, Game Design, Game Development, Programming.

1. INTRODUCTION

Game Jams to introduce young learners to computer science concepts in a fun and engaging way have the potential to introduce computer science principles. Introducing these concepts at an early age in an entertaining way, it is possible to influence or improve perceptions of computer science as a career.

Working with an easy to use game development tool, Fowler and Cusack [39] report that participants found the experience fun and engaging. Using tools such as Kodu[1], Scratch[2], GameMaker[3], and Greenfoot[4] young learners discover computer programming concepts using visual programming environments [40].

According to Fowler, Fristoe, and MacLaurin [40], Kodu is entirely event driven. Programming involves the placement of tiles in a meaningful sequence to form a condition and action based on each rule. As a result, if the user does not enter the correct code, the system will still run, but the actions will not be performed, thus reducing some of the frustrations commonly experienced by novice learners. Fristoe et al. [46] and Jones [58] also found that Kodu and similar tools are effective tools to introduce young students to some foundation programming concepts.

The lack of diversity and underrepresented minorities in STEM careers has been discussed and debated in the literature [19, 53, 74, 100]. While the representation of underrepresented minorities (URM) in STEM subjects is improving, there is still some progress to be made in the computer sciences. Through introducing CS principles through game jams, the authors intend to address the imbalance of URM through introducing a game jam and development program in both formal and informal learning environments.

2. GAME JAMS

In the literature several definitions for game jams can be found [63]. Most game jam events share similar characteristics and a process flow. In the next section those elements are listed and briefly described.

2.1 Characteristics of Game Jams

Different characteristics to describe game jams [41, 42]. The most common ones are listed below.

- *Social:* The participants are encouraged to work in small teams (2-5) to brainstorm ideas and to develop their games.

[1]http://www.kodugamelab.com/
[2]https://scratch.mit.edu/
[3]http://www.yoyogames.com/studio
[4]http://greenfoot.org

ITiCSE '16 Arequipa, Peru

DOI: http://dx.doi.org/10.1145/3024906.3024913

- *Multidisciplinary:* The The participants are encouraged to work with peers with other skill sets.
- *Time Constraint:* The projects are developed within a limited time-frame. Most common are time-frames between 12 and 48 hours.
- *Theme Constraint:* Projects developed during a game jam should follow a specific common theme, which is most often a topic in the form of a keyword, an image, or a set of development-specific rules
- *Location Constraint:* Most jams are set at a specific location, however, also online and remote forms of jams exist.
- *Game Context:* While at hackathons the context of the development is software projects, the context for game jams is narrowed down to developing games (digital or analog)
- *Jam Environment:* The focus of a jam is not to create a final product, but rather a playable prototype to illustrate an idea. Thus, it is more about the process and not about the final product.

While game jams can be organized to align with several of these elements, it is also common to focus on only one or some of these. This often depends also on the purpose of the game jam. For game jams with mainly educational purpose, often additional characteristics and constraints can be identified. This includes, for instance, specific development environments, programming tools, or technologies, or themes with an educational purpose.

2.2 Process Flow of Game Jams

Typically a game jam consists of five main phases: (1) introduction to the game jam and to the theme, (2) the brain storming process, (3) the group forming, (4) the main development phase, and (5) the final submission and/or presentation of the outcomes. Before the jam additional courses and workshops can be provided to educate participants about tools or the development process. After the jam additional efforts can be provided to encourage participants to develop further or event publish their games [81]. Often industry involvement helps to engage participants to finish working on their project and publish the outcomes.

3. BENEFITS OF GAME JAMS

In game jams, students have the possibility to work together on small game projects and are part of an entire development cycle. They not only learn the specific skills to develop a game (e.g. programming, art skills, audio), but get insights in the entire collaborative development process. Game jams enable participants (also known as jammers) to collaborate on game projects informally and in a cooperative and active learning environment. The game jams have potential to teach various skills. In this section, we describe different kinds of skills that game jams improve.

3.1 Social Capacity Building

> *Social capital-understood roughly as the goodwill that is engendered by the fabric of social relations and that can be mobilized to facilitate action"* [2].

Learning events such as game jams and hackathons are unique and separate from other types of education by virtue of their intensity and duration [38]. As participants work in teams, the social dimension of game jams is one of the important aspects of such events and the development of social capacity and social capital is one of the key benefits. Individual participation in teams can be understood in terms of social capital and social capacity, whereas a team's ability to perform the broad set of tasks needed to create a game experience can be understood in terms of community capacity.

Writers like Illich [54] and Freire [45] wrote about the importance of education and community collaboration. Social learning, was a cornerstone of the critique of schooling [54], while "problem-posing" education was seen as a tool for change and linked to community practice and a mutual process [45] and the role of the social environment as a source of development and not its setting [105].

From the discussion above it is possible to derive four dimensions of social capacity: (1) Assets, (2) Network, (3) Leadership, and (4) Environment. It may be helpful to do so in order to understand how these areas might be enhanced through education and how education might be enhanced through these four dimensions.

Whereas Eichler defines social capital as "the sum total of commitment, resources, and skills that a community can mobilize and deploy that address community problems and strengthen community assets" [35]. In contrast, emphasizes appropriability and convertibility of maintaining many relationships which can be mixed and matched to maximize the ability to help others. Goodman et al. [48] also de-emphasize assets and favor multiplicity of connections and extend the idea of a network definition of social capacity, while also stressing the need for an environment that allows for the collective engagement of community members to plan and reach decisions for action.

Bielaczyc and Collins [12] define a learning community as a culture of learning in which everyone is involved in collective learning. If the community encounters a problem, the entire learning community brings its collective knowledge to bear to solve the problem. Lee [67], expands on this by showing that it is no longer necessary for a member of the community to understand everything the community knows as long as the member is able to identify who within the community has the expertise to solve the problem. An alternative elaboration on these ideas is offered by Johnson et al. [57], who describe communities of learners as a form of distributed memory, connected through an electronic network.

3.1.1 Assets

While equipment seems self-evident in terms of representing facilities such as a meeting room or equipment such as a computer or a car, human assets are more complex. The individual capacity or capital that someone represents is really an amalgam of several factors such as that person's skills, experiences, certifications, commitment, and connections. To fully grasp the capacity or capital a given person represents, systems have to be in place to communicate this. At times this may be achieved by self-reporting or by strong ties to another person, yet at other times a listing in a directory or identification with a certification may be enough.

Assets might represent individuals or the skills an individual holds. It could also be the collection of skills that is held collectively by a group of people. A team of people might represent a valuable asset to the operation of the entire company. Assets might also be considered items such as equipment, held by individuals or organizations that are appropriable and convertible in relation to connections. Appropriability of assets might be understood in the sense that an individual uses their group of friends to gather information, using their friendship for another purpose. Convertibility means that one type of assets can be converted into other forms of assets or capital.

3.1.2 Network

An asset based view of social capacity is to look at nodes and availability of nodes, whereas the value of a networked definition privileges the number and character of connections between nodes not the actual nodes or assets themselves. Social capital defined by connections looks at the quantity of connection not necessarily at the quality of each asset or node. In this model, an individual might be valuable to an organization not because of the skills they hold, but by the connections they have. Some senators are very powerful by virtue of the connections they have to other politicians for example. This aspect of the network is also notable in that connections are not located in the individual or organizations, but between individuals or organizations. It could in this sense be considered non-rivalrous, meaning that one person's use of it does not diminish its availability for others.

Networks in this case are the varying levels of connection and goodwill that connect assets with each other. One dimension of this network is the direct network of people as represented by co-workers, collaborators, fellow members, and friends and acquaintances. Another dimension of the network are the people whom we share a connection with which has not yet been fully recognized. While the first group of connections might in many cases represent strong ties on our network, the unknown or unexplored connections might represent weak ties.

In a 2010 talk to students at San Francisco State University, Eugene Lee, CEO of socialtext.com [66] proposed that knowledge workers spend an average of 28% of their time searching for information. Based on the question, "how can you ask someone something if you do not know them yet?", Lee suggested that more can be done to leverage the informal and anonymous connections of web2.0 application such as twitter in search for answers from people we have not yet met and do not know to ask. This echoes [3] findings that weak social ties function differently from strong ties. They suggest that "weak ties facilitate the cost-effective search for codifiable information and that strong ties facilitate the cost-effective transfer of complex information and tacit knowledge."

3.1.3 Leadership/Motivation

Leadership is interesting since it is a combination of a given assets skill, yet it is dependent on an assets connections. Leadership often serves to increase the motivation of others to work together on a task and helps in the coalition building needed to achieve a given end. Leadership in social learning environments such as game jams can come from participants themselves but also from the organizers.

The extent to which students can direct their inquiries and take charge of their learning by active participation is often reflected by their motivation. Motivation is connected to the social capacity as well. The importance of interpersonal interactions as social context for shared meaning construction, critical judgment, problem solving and motivation and is emphasized [10].

3.1.4 Environment

The setting of the game jam, in classrooms, universities, industry locations shape the types of interactions that are possible. The environment can also be the theme or expressed purpose of the game jam. Creative challenges and event instructions have been shown to reflect in a game jam.

Strong social cohesion in teams can become a resource but also a constraint to innovation. Strong social cohesion supports members in times of need, but are also often at the heart of conflict and constraint when members of the same communities set up to explore new and unknown territories. Social capital thus has a stabilizing effect on individuals, which can both empower, but also hinder their ability to pursue new ideas.

Environments seeming to promote individual and collective participation, but then fail to deliver on those promises not only make the work irrelevant, but in turn also serve to de-motivate participants to participate in future actions. However the inverse holds true as well. Collective action found to be successful increase the motivation to pursue another goal.

Vygotsky [105] asserts that the cultural development of a child happens twice, first, on the social level, and only later on the individual level, making an environment that is not the context of learning, but the beginning of it. For Vygotsky, learning is intrinsically a social experience and following his thought one might speculate that an enhanced social capacity would create the possibility of an enhanced learning capacity. While for Vygotsky the zone of proximal development (ZPD) supposed a physical presence to other people, a new more distributed ZPD could be defined by mapping the varieties of physical or mediated connections any given learner might have during events such as game jams.

Game jams are often understood as individual learning events, but should also be seen as the way in which participants are connected to each other and the potential for action that these connections create. One of the most interesting aspects of the social capacity building within game jams is their relationship to emerging learning theories of connectivism[94].

Looking at the thin and thick connection in terms of the types of information they can convey can help the way we ask questions of such connections. Understanding the special and reciprocal relationships that are created by game jams can help in developing better resources for social capacity building. If we consider the way in which social capacity increases through the exercise of this capacity we can relate it to learning and even label it a kind of knowledge or skill that must be actively developed. The act of exercising social capital is then both an expression but also a further development of this capacity.

> The planning of new educational institutions ought not to begin with the administrative goals of a principal or president, or with the teaching goals of a professional educator, or with the learning goals of any hypothetical class of people. It must not start with the question, "What should

> someone learn?" but with the question, "What kinds of things and people might learners want to be in contact with in order to learn?" [54].

3.2 Engagement and Play

Game jams enable participants to make games and then play those games. This empowers, engages, and provides deep and rewarding learning experiences for the creators [59]. Moreover, game jams encourage the participants to not only make their own game but to play this game and share this play experience with others. Play has been shown to be a powerful mediator for socialization and learning [59, 111, 93]. Not only is play an exercise for developing physical activity and is a valuable form of assimilation of reality [14, 28, 49, 52]. Play is also important in developing a sense of self-regulation [104]. During play, learners can construct their own level of challenge [87, 88].

3.3 Intellectual Capacity Building

3.3.1 Career Decision Making - Peers, Counselors, Teachers and Parents

In reviewing the STEM pipeline Sass [92] uses longitudinal K-12 data college transcripts, to examine the effect of matched race and gender as well as academic backgrounds of teachers on students as they enter into their first year of a four year university.

In regard to race and socio-economics, Sass surmises that achievement gaps begin early and persist throughout high school and serve as indicators linked to drop out and college enrollment. When looking at the pre-college outcomes the data shows substantial "leakage" in the STEM pipeline for Black and Hispanic and students from low-income families, while data for women show that they are much more likely to attend college and therefore would suggest that they would be as likely to pursue STEM majors in college as their male counterparts.

While gender gaps in math achievement remain small throughout K-12 and women are more likely to complete high school and attend college, women are less likely to complete an undergraduate degree in a STEM field most notably in engineering, the physical sciences, or computer science. In turning his attention to the background of faculty in secondary and post-secondary institutions, Sass [92] finds that the number of freshman-year STEM courses taken increased when students were exposed to high school teachers with a bachelor's degree in a relevant subject area. In addition, exposure to female math and science teachers in middle and high school is correlated with increases in the number of STEM courses taken by female college freshmen.

Sass [92] concludes that future research should be conducted into the composition of the classroom at both the K-12 and post-secondary levels to individual students in math and science courses, to explore peer effects on both college coursework and major choices.

3.4 Hard Skills

Besides the development of social and working competencies, it is important to document what type of learning and other benefits the game jams promote. With the purpose of analyzing student learning in game jam participation, the authors conducted a review of published papers that address this subject. Based on the literature about learning through video game programming we have identified three broad learning categories across all papers [13]. These categories introduce new frameworks for studying Computational Thinking (CT), a concept that is being currently discussed.

3.4.1 Conceptual Learning

This category includes essentials computer science and programming concepts such as: sequence, cycle, conditional, variable, or events. Denner et al. [30] coded 108 video games using a rubric that helped to identify CS concepts using the platform Stagecast Creator[5]. 59 Latino girls from a middle school developed these games during an after school program during 1 to 2 hours a week. Researchers coded the game in three categories: programming concepts, code organization and design and usability. In terms of programming concepts they found that most games include conditions or events, door functionality (moving from one stage to another) and conditional character interaction. Less used were variables, random and parallelism. Denner also points out that Clancy [20] found that young students do not understand the use of *while* and *if*, and that novice programmers appear to expect tests to be handled as events that when satisfied would result in a jump to the code within the *while* (or within the *if* part of the conditional). Even when conditionals were placed in sequential code segments, students did not expect the conditional expression to only be tested when the statement holding the conditional is executed. Maloney et al. [71], Wilson et al. [109] and Stolee [97] also found certain concepts were used less often in their students' projects made using the programming language, such as random numbers, variables, and Boolean Logic. Martinez et al. [72] study show primary school students incorporate essentials CS concepts developing video games and animations with Alice, and how students could transfer CS concepts incorporated while they were developing video games to other platforms or challenges. Games provides students with visual feedback, about what they have recently programmed, so they can watch the CS concept working. Kölling and Henriksen [62] emphasize that: "One of the problems in traditional programming environments is that object behavior is not directly observable."

3.4.2 Competencies or Practice to learning

This category includes the learning of four types of practices typical of computer programming.

- **Incremental and iterative planning:** developing a plan to design and code.
- **Test and debugging:** through trial and error, transfer of activities or support from knowledgeable?
- **Reuse and remix:** building on other people's work, it supports the development of critical code reading capacities and provoke important questions about ownership and authorship.
- **Abstract and modularize:** problem solving, building something large by putting together collections of smaller parts.

[5] http://acypher.com/creator/

Werner[106] in an after school club for middle school students to incorporate important skills through developing video games using Macromedia Flash FX. They focus on algorithmic thinking, programming and code modularization.

3.4.3 Perspectives learning

This category includes CS attitudes. Brennan and Resnick [13] found three perspectives about Computational Thinking using Scratch.

- **Expressing:** a computational thinker sees a computation as more than something to consume, the computer is something they can use for design and self expression as a medium to think.
- **Connecting:** are social practices enriched by interaction with others, through face to face interactions or online networks, online forums.
- **Questioning:** young people should feel empowered to ask questions about and with technology.

3.5 Tangible Outcomes

One opportunity of game jams is not only the aspects of the development process, but also the tangible outcomes. Usually participants have first game prototypes for their portfolio. In particular in terms of career chances, having a strong portfolio and game projects to show at interviews is extremely important [80]. But not only a playable game can be a tangible outcome of game jams. It can be also a new algorithm, working infrastructure, or other resources, which can be reused for further projects.

3.6 Format

The format that can be applied to a game jam with children, it is necessary to concentrate on some aspects that define what can really be done by children and what kind of benefits are more tangible.

The first item to consider is the high intensity of game jams. The limited time available (48 hours in most game jams), may not be suitable for children. The sleep deprivation has a negative an undesirable effect on their health and cognitive function [29].

Their parents also have security concerns, if they are not able to stay together in the event. In some countries there is also a concern about getting a special declaration to authorize children to participate. This context justifies that it is necessary to adapt the game jam duration or time constraints when considering creating a game jam for young children.

First, children are typically not permitted to stay overnight. Then, if they have to learn about time constraints, this learning has to be tangential, not reinforced by the rules. That is, the process to get to the result need to be more valued than the game itself. Learning under pressure is an import skill to develop during the jam, but it's less relevant than the iterative design developed.

They will learn about how time constraints will be important in their academic and professional lives, but they will not be punished by not delivering a complete game. This justifies creating awards for those who reach the end of the event and validates their participation and will encourage future participation.

3.7 Essential (Soft) Skills

The lack of teaching students soft skills is still an issue in modern education environments [107] and it is crucial to teach students relevant soft skills to make the transition to the labor market easier and enhance their employability [16]. Soft skills include in particular social skills, such as the ability to work in a team, the respect for peers, and to communicate, but also organization and life skills, such as the ability for self-directed organization, critical thinking or creativity [16]. However, it is unlikely that all of these skills can be taught within a classroom environment [26].

The pressure environment of game jams has a high potential of learning different skills related to working as a team together on a project towards a common goal within a limited time-span. Thus, one of the most important effects of game jams for participants is the development of different skill sets, which are important to work in different project-related industries.

3.7.1 Social and Communication Skills

In this category we find different attitudes describing the way of communication and interaction with others.

- **Communication** Written and verbal communication, as well as teamwork are listed by many companies as an essential core skill of future employees [17]. Game jams are an ideal place to learn how to communicate effectively directly with peers, but also in different online tools [79, 82].
- **Working as a team:** Cooperation and learning how to work in and as a team, in particular as an interdisciplinary team (programmers, audio and sound engineers, artists, designers) is often a unique and very new experience to many [80].
- **Working in an interdisciplinary team:** The interdisciplinary character of the game development process requires jammers to learn how to communicate and work together with people from different disciplines with different skill sets [6, 80, 79].
- **Respect for others:** Respect for project partners is an integral part to work in teams and an important soft skill. Working together in interdisciplinary teams is also a powerful tool to enhance the participants' understanding and respect for others [16].

3.7.2 Organizational Skills

In this category different skills related to self- and group-organization are described in the context of game jams.

- **Independent working and self-organization:** In game jam environments participants report on improving various skills related to self-organization such as time management, or personal project organization. They have the possibility to learn how to optimize their personal working flow, and gain experiences organizing rapid prototyping projects from the very beginning [80].
- **Group Coordination:** Due to the collaborative and social nature of most game jams, participants are required to learn how to organize and coordinate a rapid prototyping project within a very short time span involving team-members with various skill sets, using different sets of tools working on the same project at the same time.

3.7.3 Personal Development and Learning Skills

Aspects like problem-based thinking and creative problem-solving are skills, which are rated very high key for working in industry.

- **Thinking in a diverse context** Bridging different disciplines is an important step to support innovative work and think in a broader context [16, 79]. The game development process requires the integration of different skill, such as programming, art, audio engineering, design, or project management. This setting encourages new forms of thinking in a diverse and interdisciplinary context [36].

- **Problem-based thinking and creative problem solving:** Problem-solving skills, the ability to learn from this problems, and learn new skills quickly are important to enhance the employability [16, 101]. In game jams participants are constantly exposed to new problems and have to adapt quickly.

3.8 Types of Learning

Games jams can be used to improve different types and strategies to learning in a more active way [38]. These different strategies can also be applied in the classroom, but game jams can accelerate the possible outcomes. In this section the authors will explore some learning strategies that the authors consider the core benefits of participating in game jams.

3.8.1 Informal Learning

It is important to make it clear that the environment of a game jam is an informal learning environment. The informal learning or informal learning process is characterized by allowing people to construct, acquire and accumulate knowledge, skills, abilities and attitudes in their relationship with the environments in which they move [22].

The formal school or traditional 'Grammar School' [102] is characterized (among other traits) by: a division of knowledge into fragmented subject areas, the organization of time in short blocks, and homogeneous grouping of students. Teaching and learning programming requires classroom dynamics that challenge this traditional grammar of schools. For example, most programming platforms provide immediate feedback to the programmer on their coding skills. In contrast, the majority of teachers provide summative assessments days after the activity was completed. Working hours are flexible for programmers who need to be on task uninterrupted. In opposition, school schedule forces students to change subjects every approximately every hour [33].

Different experiences demonstrate how gaming activities allow young people to learn central concepts of programming and challenge the traditional format of School. When Maloney[71] asked whether Scratch reminded the youth of anything at school, all of the respondents said that Scratch was at least like one school subject, and most cited several subjects that they thought connected to their experiences in Scratch. The most frequent response was generally to identify a connection to the arts and few answers are associated with computing and maths. From these responses, we learned that children felt that Scratch was most similar to schooling activities that support creative, personal expression, such as art and language arts. The majority of the participants didn't identify scripting in Scratch as a form of programming.

In a case of experience of Chamillard [18] with student of 17 and 12-16 years with Game Maker and The Game Factory in school, the young reported that they prefer to leave the normal classroom to use the lab, but the most interesting side effect of this course was that it drove a change to our standard computer lab policies. Those policies prohibit students from playing games on the lab computers. This was clearly an inappropriate policy for students in this course, since they would essentially be required to create their games without being allowed to test them. Its is another sign that video game development defies the formal structures of the school.

Also common in formal learning environments is the clustering of the students by age in courses considering that everyone learns in the same way according to their age. However, the experience of Wilson, Hainey and Connolly [109] using Scratch with mixed grouping in primary school shows that both primary 5/6 and primary 6/7 classes had equal amounts of original games, indeed more than half of each class made their own game. However the primary 5/6 class were more successful in implementing their games with more functionality than the primary 6/7 class.This allows us to infer that the age difference is not necessarily related to their ability to learn programming and teach programming challenges the formal structure of the school [34]. In that way, a game jam also challenges the formal structure and is the appropriate environment to teach programming and STEM Skills in an informal way. The following sections describes more types of learning that can be empowered by game jams and also can be associated to formal or informal learning.

3.8.2 Problem based learning

Students incorporate concepts and knowledge through challenges and problematic situations. Before introducing a specific concept, a teacher may present a challenge that would be resolved with this concept. However, students need to try to resolve it, without knowing this particular concept. They have to look for solutions, situating students on active role.

The Problem Based Learning (PBL) strategy puts the student in an active role, that has a similar relationship with the creation of video games inside a game jam because they work hard on their productions exploring and challenging themselves without the need for a teacher to help them decide what steps they should take. Therefore PBL promotes efficient reasoning and creativity.

PBL encourages the development of skills in search and information management. It also develops research skills in students because in their learning process, students will have to find out and understand what is happening and achieve an appropriate solution [91]. This is an activity highly linked to computational thinking which is characterized by decomposition into subproblems, abstraction of particular cases and design processes.

PBL also has a particular relationship with time, it is typical to be assigned plenty of free time for the student, so that you have the time to read the material you need, seek information and work on the proposed project . This encourages students to have their own learning processes. Programmers experience have a particular relationship with time. Many of them need to keep working continuously for hours, unlike other jobs where schedules may be shorter and regular. Some programmers, even feel a sense of "flow time" when working on the computer [108, 34].

3.8.3 Challenge Based Learning

In a similar way of the Problem Based Learning strategy, Challenge Based Learning (CBL) makes learning more relevant by giving students problems big and important enough so that they have to learn new ideas and tools to solve them. As a multidisciplinary approach to education, CBL creates a space where students can direct their own research and think critically about how to apply what they learn.

Students expect their assignments to have a relationship with real-world tasks and context. Challenge Based Learning is best suited to establish this relation as this strategy provides focus on global challenges with local solutions [55].

The difference between CBL and PBL is that in CBL the question or problem are replaced by a challenge [5], what is more convenient for game developers and one of the main reasons why many people play games [64].

There are some similarities between what is expected when a student works in a game jam and what is expected in Challenge Based Learning as we perceive when the theme of the jam can be always considered a global challenge and the game to be created can be seen as the local solution for this challenge.

The implementation of challenge based learning (CBL) in schools [55] demonstrates that CBL is useful to develop the 21st Century Skills [99] like leadership, collaboration, problem solving and social skills. These are also skills that game jams can improve.

For teachers, game jams can bring some the same meaningful outcomes that can also be considered when we see the environment as a Challenge Based Environment[56]: cross-grade-level interactions, children's excitement, seeing students naturally use math skills, emulating real world problem solving and of course having the chance to be creative.

3.8.4 Flipped Learning

In the flipped learning model we expect the student to move the direct instruction (lectures) from the group learning space to an individual learning space. This change transforms the group space into a more dynamic interactive learning environment. This learning space is best suitable to the educator to guide students to apply concepts and to creative engagement [11].

Flipped learning uses extensively, but not exclusively, teacher-created videos to deliver the direct instruction. Research demonstrates that in that way, teachers have more time inside the group classroom to develop strategies of student-centered learning like projects, challenges and discussions. High school students who are used to flipped environments have a positive impact on both their knowledge and their attitude [7].

Game jams always have time constraints to be considered. But this kind of event also try to be inclusive, bringing together people that never participate before in the production of a game. And for first time jammers, that are always a large percentage of the jammers, are generally interested in improving skills and gain experience [96]. In this context, it‘s sometimes necessary to train jammers to get some game skills like programming, art or even game design.

Unfortunately, because of the time constraints it's challenging to make the necessary training inside the game jam environment. Different solutions can be applied like mentoring or even warm-ups or lectures in the weeks before the jam. Applying a flipped learning approach to jammers in the weeks before the jam can be an opportunity to coach them in an effective way and also can be helpful in team formation.

Others approaches like just-in-time learning [77] can also be introduced through game jams. This strategy involve the teacher adapting his lessons based on student feedback. In this context, the teacher can send videos or materials to his or her students and ask some questions before the game jam to check their possible performance during the event and adapt rules, support, mentoring and even the theme using this feedback.

4. STRUCTURE OF GAME JAMS

4.1 Tools for Game Jams

Through experience in the field, the authors have identified a number of tools that are valuable for creating video games. The following section provides an overview of the various tools used. These have been categorized into the following sections, game development tools, creative tools, collaboration tools, and social tools.

4.1.1 Game Development Tools

Any Game Development Tool can be applied in the context of a game jam. However, there are tools that are more suitable to create prototypes in a short time, what is one important point in a game jam. Table 1 provides examples of the programming tools popular in game jams. Another tool that has been used successfully to teach the children the foundations of programming in a formal educational context is CS Unplugged [8].

4.1.2 Creative Tools

The creative tools that are available to game jam participants for young children are primarily embedded within the game development platform. Tools like Scratch, Alice, and Kodu provide the available assets for developing games. Therefore, there is not much need for participants creating their own content. However, the other development tools noted in Table 1 do provide scope for creating the graphics and audio needed for a game. Table 2 provides examples of some of the creative tools used in game jams.

4.1.3 Social Tools

The social media tools available for most game jam participants included options like Facebook, Twitter, Github, and many more. However, due to the age restrictions of many of these choices, there is a limit to what tools are available or suitable for a younger demographic. While, some nine year-old children may have access to Facebook or twitter, this is not prevalent. Kodu game lab and Scratch have the facility to share the games made on a moderated website. Moreover, there are tutorials and lessons on YouTube.

4.1.4 Improv for team formation

Game jams require a creative and improvisational mindset and teamwork. Improv theater exercises are useful in promoting teams which are leaderless. In the CSU East Bay Game Jam, improv is used at several stages throughout the 54 hour event to facilitate experiential learning about key concepts related to the event, build greater community and encourage a risk taking environment and an innovation mindset.

Improve exercises follow game like structures of being separate in time and space, rule based and unpredictable. Within improv exercises one can distinguish between low and high risk exercises. Low risk means that there is minimal personal investment. Circle clap and sound ball exercises fall into this category. In both cases players are not required to share any personal thoughts or ideas and are not required to demonstrate great personal creativity. Higher risk activities include the physical prototyping and name gesture activities. Both of these put the participants more into the spotlight in terms of ability.

4.1.5 Circle Clap

In circle clap, players arrange in a circle in such a way that each person can see the person the left and to the right with a simple twist of the upper body. The facilitator instructs the players and turns to the person on the left. Looking into each other's eyes the facilitator and the first player clap at the same time. Now it is the turn of the first player to turn to the left, look the second player in the eyes and clap together. In this manner the clap is passed from one player to the next. After the first round the speed can be increased and additional claps added. The game continues to speed up until it is called to an end. In the debrief of the game, the facilitator stresses the importance of the players to check in with each other, alternating between sender and receiver position. This game serves as a model for how game jam team members should be attentive to each other in their teams.

4.1.6 Sound Ball

In Sound Ball, players arrange in a circle in such a way that each person can see all the persons in the circle with minimal movement. The facilitator instructs the players and throws an invisible ball to another member in the circle. As the first person throws the ball the produce a sound. The person to whom the invisible ball is thrown catches the ball while making the same sound. Once in possessions of the invisible ball, it is now their turn to throw the invisible ball and make a new sound. Again the ball is caught by the next person who copies the sound and then passes on the ball. The game illustrated the concept of building onto what has been received,

4.1.7 Name Gesture

In Name Gesture players arrange in a circle in such a way that each person can see all the persons in the circle with minimal movement. The facilitator instructs the players and moves quickly into the center of the circle in a personalized and individual fashion. While they are moving into the circle they pronounce their name. Once in the middle, all the other players rush into the center of the circle, copying the gesture and saying the name of the person in the middle.

This game serves as a way to introduce people to each other by name, understand people in terms of physical writing and celebrates each person, but putting them into the center of attention.

4.1.8 Physical Prototyping

In the Physical Prototyping exercise players form groups of 5-7 members and arrange themselves in the space with enough room to move. The facilitator explains the game and calls out the first item to be enacted. Without being able to talk to each other, teams now have to arrange themselves to create a physical, embodied version representational of the item called out. Teams have 60 to 90 seconds to create the object. Once completed, teams are encouraged to look a the other teams. Objects to create can vary from simple to complex. Airplanes being relatively simple, while blenders, toasters and abstract concepts such as extension cords, electrical plugs and electricity are more complex.

Exercises like the circle clap and sound ball are designed to emphasize attention giving and receiving whereas name gesture and prototyping gesture are more designed to encourage teamwork and creativity.

Name gesture achieves this by collectively celebrating individual in name. Physical prototyping depends on creativity, non-verbal communication and team cohesion.

As participation in game jams is voluntary, models of collaboration and leadership must take into account the contingent and time constrained nature of the event and the teams. Leadership in game jam teams, therefore must be leaderless, encouraging participants to be yielding and asserting as various times to succeed as a team.

The alternations between flat and hierarchical leaderships have been explored in depth and should be understood as a balancing act depending on whether the teams are trying to meet a series of tasks and deadlines or a creative challenge [112, 1, 86]

CS unplugged [8], is another program that has been used to teach some of the concepts of programming through physical activities.

4.1.9 Misc Tools

There are some tools that cannot be classified in the previous categories. Chronolapse, for instance is a tool that can be used not only for artists but also for programmers. This tool allows the participants to create time-lapse / stop-motion videos of his or her web-cam or screen. It's useful when the participants wish to value the process of participate in a game jam.

4.2 Retaining Interest and Participation

Through experience in the field, the authors found that providing tangible and visible indicators of student success and mastery of each level, provided a compelling and motivating rewards system. Using digital varieties of such incentives has been referred to as gamification [31]. To increase continued participation and engagement (and learning), the authors provides badges (aka buttons) to students for completion of each level. The first badge that could be obtained after completing a jam is green. The next level is blue, the next level is red, the next level is brown, and finally, the highest level a participant can obtain is black. If participants continue to join in the game jam, they will be awarded additional black badges.

Having external indicators of achievement (badges, belts, stars), provide two benefits. For the person wearing or displaying the indicator of achievement, this r the people who have not yet obtained this achievement level, it provides a very visible form of evidence of achievement which can provide extrinsic motivation (see Figure 1).

Figure 1: Game Jam Junior Badges

5. OBSERVATIONS / EXPERIENCES IN THE FIELD

Though experience in the field, the authors have gathered experiences from game jams and similar events. The following section provides an overview of these experiences and the affordances and challenges they provide.

5.1 Game Jam with Brain Bee: STEAM learning through neuroscience-themed game development.

This project brought university faculty and students from art and science departments together with high school teachers and students in developing games around a neuroscience theme, bringing two promising programs, game jams and Brain Bee to students who have been traditionally underserved in opportunities to participate in engaging STEM activities. University student teachers met weekly with high school students for mentorship, tutorials, and working on game development in teams. The program prepares high school students to participate in the annual Brain Bee (an SFN sponsored competition), and in a game jam and culminates with a public presentation of their game development.

The overall goal of the project was to provide a unique opportunity for under-served high school students to participate in a fun, educational STEAM-oriented after school program that will spark their continued interest in STEAM as they gain knowledge and practice in applying it, hone their communication, teamwork, and idea development skills, and see themselves as being able to contribute to a complex project in a way they may not have before. They also built their portfolio for applying to college and jobs through their development of a game and as a Brain Bee competitor.

High school students in this program participated in an intertwined curriculum during their weekly after school meetings that prepares them for Brain Bee (through learning neuroscience) and developed the foundation they need to make a game (i.e. game making, game theory, and practice with different technologies that could be used in their game development). The program began in October 2015, with a greater percentage of time (approximately 70%) at the outset spent on preparing for the Bay Area Brain Bee, which was in January 2016. After that, the curriculum was more heavily focused on game development and technology (approximately 70%) in preparation for the High School Game Jam in March. The neuroscience aspect of the curriculum continued, moving towards higher order thinking and application. Participating students were invited to the final presentation of the California State University, East Bay (CSUEB) May Game Jam to gain inspiration and networking opportunities.

This unique intertwined curriculum for the program was derived from a Brain Bee curriculum at Michigan State University further developed and documented with detailed lesson plans for eventual dissemination and adoption at other school sites. We are assessing the program and participants for increased content knowledge and science self-efficacy.

5.2 Game Jam with children from 'Associacao Sagrada Familia de Nazare'

One game jam that includes both children and adults was organized in Brazil in 2014. With the support of the Associacao Sagrada Familia de Nazare, thirty children were recruited from the community. A presentation was introduced about what is involved in creating a game and asked them to draw a character or an environment. The only constraint was: the participants have to follow a peace theme, not involving a violence of any type. This constraint was challenging to this children, because this community has violence problems.

In the next step, a game jam with adults (graduate students) was organized and they had to use the drawings that the children produced to inspire the games. Two weeks before the jam, the children were shown the games. They immediately recognize the drawings and are also interested in game development.

5.3 'Espaco Kids' inside Global Game Jam Curitiba 2016

The previous controlled experiment was expanded in January 2016 during the Global Game Jam Curitiba. This time a separated space was created for the children ('Espaco Kids'). In this space the children could participate in recreational activities like drawing, music production or programming with Construct 2. Everything that was created from these children was available to normal jammers and one of the diversifiers was: 'Won't somebody think of the children? Your team must use the works from a child or children - Meta'. We have almost 500 jammers involved and almost 40 children participate on 'Espaco Kids'.

As it can be perceived, children and adults working together is possible with this kind of separation of environments. This approach is possibly the first step to provide core skills and make a child comfortable with game jam environment. Of course, the next step will to be bring the children to their own environment, like what the authors are proposing with Game Jam Junior.

5.4 SEEDS' Game Jam

This game jam promoted integration between different professionals and the Civil Defense of Curitiba. Jammers are supposed to create games that explain Civil Defense's daily work in a more interesting way, compared to lectures and folders.

The produced games were applied with children. That's another way to format a game jam: make the jam's goal something that can be used in the classroom.

5.5 CSU East Bay Game Jam

The CSUEB Game Jam is a quarterly, 54-hour game-making event. Following a voluntary series of weekly tutorials by put on by senior students and industry professionals. This game jam opened on Friday afternoon with a mixer event, improv exercises, instructions and a keynote address by an industry or independent (indie) game developer.

The opening remarks outline and unveil the rules and common theme for the particular event, sustainability. This is followed by an ideation, a pitching session and team formation. Teams arrived in the morning Saturday and Sunday for a small breakfast and pep talk before beginning to work on their ideas. The themes for the game jam are presented at the opening event. The themes for 2-14-25 were sustainability, diversity and heroines.

The event is a collaboration between the graduate Multimedia Program, the Art department The Computer Science department, and the Institute for STEM Education,

The 2014-15 events served a total of 150 students.

CSUEB Game Jam create students who understand and have experienced the linkages in cross functional teams and the synergies needed for innovation thinking and rapid prototype creation. They gain skills and personal appreciation for AGILE and team based work flows and learn to negotiate team dynamics and milestone driven development.

5.6 Interdisciplinary and International Game Jams at TU Graz

Pirker et al. [79] use an interdisciplinary and international game jam format to promote the collaboration between computer science students (at university level) and students from other disciplines, such as law, biology, or design. Computer science students from Graz University of Technology (Austria) have the possibility to develop games as part of the computer science curriculum in multi-disciplinary teams. In the first pilot study, these students were mixed with students from University of Westminster (UK) coming from very different fields, which are not essentially related to games engineering. In a first two-day game jam they build together on-site a first game concept and the initial game design. In the following weeks the teams can work remotely on their projects. In a final on-site game jam the teams meet again to finalize the game projects within another two days.

Working together with peers from different cultures and different backgrounds was described as a totally new situation for most students and is identified as a promising way to prepare them for similar situations that they might find in industry. Results, as reported in [79], indicate that such formats are perceived as engaging learning experiences and raise the students' interest in international collaborations, and boost their employability. Beside having typical game development experiences and gaining coding and software engineering skills, the students learn how to collaborate in a multi-disciplinary and remote setting.

5.7 Kodu Game Jam

Fowler et al. [40, 39], used Kodu Game Lab (Kodu) to engage middle school and high school students in learning programming concepts. Fowler found that the game development tool was useful for engaging young learners in learning fundamental CS principles. Through introducing the tool as part of the School schedule, the researchers were able to teach the students how to use the development platform and introduce some game design concepts. When the instructional component was complete, the students were challenged to make a game based on a specified theme within a specified time constraint (game jam). The curriculum was introduced using a combination of elements of CS Unplugged [8] and in-class instruction. The authors implemented this program in four schools a Middle School (in New Zealand), two High Schools (one in New Zealand and the other in the United Kingdom), and a Summer Camp (in the USA). These programs gave the authors the opportunity to understand the educational potential of game jams and using game creation tools to introduce young students to programming concepts and game design in both formal and informal learning environments.

The authors reported that in New Zealand 71% of participants (N=23) reported that using Kodu was an enjoyable experience. Moreover, the class teacher reported 89% engagement [40]. When the same technology was used in the USA, the authors reported that 60% of the participants (N=10) reported a positive experience. In the UK, the authors found that 38% of the participants (N=14) reported a positive experience. While only 23% reported a negative experience. However, the technology was still in beta at the time and the issues were focused around the software crashing.

The authors also found notable changes in perceptions of computer science and programming. In New Zealand, the authors found that there was a 60% increase in the number of participants that strongly agreed with the statement "I could be good at computer programming". In the USA there was a 50% increase in the number of participants that strongly agreed with this statement. Moreover, in the UK, there was a 70% increase in the number of participants who strongly agreed with this.

5.8 Dale Aceptar

Dale Aceptar[6] is a free Argentinean online competition organized by the Sadosky Foundation[7] . It is performed annually with the aim of interesting more students into pursuing CS-related careers. The competition is atypical, in the sense of being aimed at students with no prior background in programming, who sign in because they see the commercials on national TV and feel like having fun, win a prize or both. While they participate, students are also exposed to short pieces of information about CS and its advantages as a career choice.

The site offered 23 short video based lessons on Alice. These videos included the basics, and more advanced curriculum such as making a turn-taking, timer-based game.

The students could choose another platform in order to participate. Chatbot [9] is an educational software tool whose design goal is to motivate students to learn basic CS concepts through the construction of chat automata. It has a mode of operation where it can connect to social networks (such as Gtalk and Facebook) and reply to chat conversations automatically. The chatbots can be programmed to answer in different ways depending on who it is talking to, what the person is saying, which topic they talked about before, etc. Chatbot can also be used to explore more advanced concepts such as the Turing test and Natural Language Processing concepts (e.g. lemmatization and syntactic analysis).

The competition has attracted more than 27000 students in high school from 2012 to 2014. In this paper we report on the characteristics and findings of the competition which took place in 2013.

Dale Aceptar attracts mainly male students (partly because the prize of the competition is a gaming console), the percentage of female registration is twice higher with Chatbot (23%) than with Alice (12%), this difference is statistically significant. We do not know what motivations students have for choosing one platform over another, but similar female

[6]http://www.daleaceptar.gob.ar

[7]http://www.fundacionsadosky.org.ar/

preferences were seen in the classroom Girls' self-reported interest was higher than boys' both for Chatbot and Alice.

During 2013, 9371 teenagers signed in to participate in Dale Aceptar, 8137 (86.83%) being male. 8502 participants decided to participate with Alice, and 1454 decided to go with Chatbot. 585 participants signed in to participate with both.

5.9 Mantovani's video game development

In 2014, the UNC++[8] research group implemented a CS learning experience [72] in a real school setting. With the purpose of both investigating how children learn basic CS concepts in schools creating video games and using programmable toys and contributing to a CS curriculum selection and scope; we designed an exploratory study to compare how preschool children (ages 3 and 5), and elementary school children (ages 8 to 11) learn some basics CS concepts. Preschool students programmed the N6 programmable robot [9], using a multilanguage platform developed by UNC++ team called UNC++Duino. The platform allows students to program the robot with different languages: Icony (a language completely based on images), Blockly [44], Python and C/C++.

The intervention in the elementary school started in May and finished in December of 2014. A member of UNC++ research team was the CS teacher for one hour a week. Students started working the first weeks with the Code.org tutorials [10] as an introduction. Then they developed video games and animations using the Alice platform [23]. And finally they learned how to program the robot using the UNC++Duino platform. We used three different platforms because we want students realized that basic concepts of computer science are always the same no matter what platform or language they are using.

The preschool and elementary experiences were designed following the challenge and problem solving learning paradigms. Students worked in groups of 2 or 3 members. Some of them were mixed gender groups. Students choose the groups.

First, the teacher briefly introduced the main characteristics of the platform that the students used. Then the students explored the platform by themselves. Also they walked around the classroom to see what their partners had discovered. Before introducing a CS concept, the teacher proposed a challenge to the students. Their task was to explore the platform and think how to resolve the problematic situation. As before, they walked around the classroom asking for help or helping their partners to solve the challenge, promoting collaboration. The teacher also walked around the class, in order to give some clues to their students if they felt they were stuck in the challenge.

135 primary school students participated in the complete experience. But only 85 took the test, which was confirmed by seven multiple choice questions. The test had two kinds of questions: simple (students have to use only one concept to resolve the challenge) and complex (students have to use various CS concepts to solve it). Sequence, loops, conditionals statements and parameters were the CS concepts introduced to the students.

[8]http://masmas.unc.edu.ar/

[9]http://robotgroup.com.ar/index.php/es/productos/robots/n6-max-detail

[10]https://code.org

Figure 2: Mantovani's Experience

6. FUTURE DIRECTIONS

The experience in the wild and the supporting evidence suggests that game jams have the potential for effectively engaging young learners in STEM activities. These events have the potential to increase participation in STEM subjects and careers. The authors plan to develop an annual game jam that specifically targets young learners - Game Jam Junior. Game Jam Junior is a program that includes curriculum to teach the basics of programming and game design fundamentals. These programs are delivered through summer camps and other formal and informal learning opportunities. Game Jam Junior was developed to increase awareness and participation in STEM subjects. However, the long term goal is to increase gender and ethnic diversity in computer science. Through introducing programming and game design to children at an early age, the authors intend to dispel perceptions that programming and game design is difficult and that it is only suitable for white males (a demographic that is well represented in the computer sciences and, game design and development). However, while an intervention at an early age would be a valuable tool for dispelling these perceptions, it is equally important to maintain these perceptions throughout a child's teens. Therefore, the authors also plan to introduce a Game Jam for Teens, which will be supported through curriculum and summer camps and other informal (and formal) learning environments.

It is important that the program is evaluated to establish the impact and significance of the intervention. Craig, Fisher and Lang [25] provide a valuable framework for evaluating intervention programs. To evaluate the effectiveness of Game Jam Junior, the following measures will be monitored:

- **Persistence:** the degree to which this program maintains participation.
- **STEM self-efficacy:** the degree to which this program affects personal perceptions of the participants of their ability to apply computational thinking to problem solving.
- **Knowledge of STEM careers in CS:** the degree to which participants perceive CS careers.
- **Perceptions of careers in CS:** problem solving, building something large by putting together collections of smaller parts.
- **Network building:** the degree to which the participants build and maintain a support network.

- **Transportability:** the degree to which the intervention can be implemented in formal and informal learning environments.

The effectiveness of the intervention will be evaluated through using qualitative and quantitative measurement tools. Baseline measures and post-exposure evaluations will be obtained through using child-friendly measurement tools (see for example the smileyometer [83]).

7. CONCLUSIONS

From the evidence provided, the authors are convinced of the pedagogical benefits of game jams for young learners. Although more evidence is needed to generalize these findings, this paper provides a contribution to an understanding of the benefits of game jams for young learners. The outcomes of the study can be applied for guidance in the adoption of these programs by educators and educational policy makers.

8. ACKNOWLEDGMENTS

The authors thank the ITiCSE 2016 conference chairs and community for their support and encouragement through the development of this paper. We also thank the reviewers for their valuable and constructive feedback. Finally, we thank the many children and volunteers who participated in the events that made this possible.

9. REFERENCES

[1] D. Abfalter. Authenticity and respect: Leading creative teams in the performing arts. *Creativity and innovation management*, 22(3):295–306, 2013.

[2] P. S. Adler and S.-W. Kwon. Social capital: Prospects for a new concept. *Academy of management review*, 27(1):17–40, 2002.

[3] P. S. Adler and S.-W. Kwon. Social capital: Prospects for a new concept. *The Academy of Management Review*, 27(1):17, 2002.

[4] T. Bah. *Inkscape: Guide to a Vector Drawing Program (Digital Short Cut)*. Pearson Education, 2009.

[5] N. Baloian, H. Breuer, K. Hoeksema, U. Hoppe, and M. Milrad. Implementing the challenge based learning in classroom scenarios. In *Proceedings of the symposium on Advanced Technologies in Education*, 2004.

[6] J. D. Bayliss and S. Strout. *Games as a Flavor of CS1*, volume 38. ACM, 2006.

[7] M. R. Bell. An investigation of the impact of a flipped classroom instructional approach on high school students' content knowledge and attitudes toward the learning environment. 2015.

[8] T. Bell, J. Alexander, I. Freeman, and M. Grimley. Computer science unplugged: School students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology*, 13(1):20–29, 2009.

[9] L. Benotti, M. C. Martínez, and F. Schapachnik. Engaging high school students using chatbots. In *Proceedings of the 2014 Conference on Innovation; Technology in Computer Science Education*, ITiCSE '14, pages 63–68, New York, NY, USA, 2014. ACM.

[10] Z. L. Berge. Guiding principles in web-based instructional design. *Educational Media International*, 35(2):72–76, 1998.

[11] J. Bergmann and A. Sams. *Flip your classroom: Reach every student in every class every day.* International Society for Technology in Education, 2012.

[12] K. Bielaczyc and A. Collins. Learning communities in classrooms: A reconceptualization of educational practice. *Instructional-design theories and models: A new paradigm of instructional theory*, 2:269–292, 1999.

[13] K. Brennan and M. Resnick. New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada*, pages 1–25, 2012.

[14] S. L. Brown. *Play: How it shapes the brain, opens the imagination, and invigorates the soul.* Penguin, 2009.

[15] B. Burton. Learning mobile application & game development with corona sdk. *Abilene, Texas, United States of America*, 2013.

[16] V. Callier, R. H. Singiser, and N. L. Vanderford. Connecting undergraduate science education with the needs of today's graduates. *F1000Research*, 3, 2014.

[17] L. Carter. Ideas for adding soft skills education to service learning and capstone courses for computer science students. In *Proceedings of the 42nd ACM technical symposium on Computer science education*, pages 517–522. ACM, 2011.

[18] A. T. Chamillard. Introductory game creation: No programming required. *SIGCSE Bull.*, 38(1):515–519, Mar. 2006.

[19] M. J. Chang, O. Cerna, J. Han, and V. Saenz. The contradictory roles of institutional status in retaining underrepresented minorities in biomedical and behavioral science majors. *The Review of Higher Education*, 31(4):433–464, 2008.

[20] M. Clancy. Misconceptions and attitudes that interfere with learning to program. *Computer science education research*, pages 85–100, 2004.

[21] M. J. Conway. Alice: easy-to-learn 3d scripting for novices. 1997.

[22] P. H. Coombs and M. Ahmed. Attacking rural poverty: How nonformal education can help. a research report for the world bank prepared by the international council for educational development. 1974.

[23] S. Cooper, W. Dann, and R. Pausch. Alice: a 3-d tool for introductory programming concepts. In *Journal of Computing Sciences in Colleges*, volume 15, pages 107–116. Consortium for Computing Sciences in Colleges, 2000.

[24] S. Cooper, W. Dann, and R. Pausch. Teaching objects-first in introductory computer science. In *ACM SIGCSE Bulletin*, volume 35, pages 191–195. ACM, 2003.

[25] A. Craig, J. Fisher, H. Forgasz, and C. Lang. Evaluation framework underpinning the digital divas programme. In *Proceedings of the 16th annual joint*

conference on Innovation and technology in computer science education, pages 313–317. ACM, 2011.

[26] S. Cranmer. Enhancing graduate employability: best intentions and mixed outcomes. *Studies in Higher Education*, 31(2):169–184, 2006.

[27] D. M. Cruz and R. M. de Albuquerque. A produção de jogos eletrônicos por crianças: narrativas digitais e o rpg maker/the production of electronic games by children: digital narratives and the rpg maker. *Comunicação & Educação*, 19(1):111–120, 2014.

[28] M. Csikszentmihalyi. Some paradoxes in the definition of play. *Play as context*, pages 14–26, 1981.

[29] R. E. Dahl. The impact of inadequate sleep on children's daytime cognitive function. In *Seminars in pediatric neurology*, volume 3, pages 44–50. Elsevier, 1996.

[30] J. Denner, L. Werner, and E. Ortiz. Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers & Education*, 58(1):240–249, 2012.

[31] S. Deterding, D. Dixon, R. Khaled, and L. Nacke. From game design elements to gamefulness: defining gamification. In *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*, pages 9–15. ACM, 2011.

[32] R. R. P. Diniz. Uma trilogia perfeita: Rpg maker xp, educação e adolescentes. *Monografia (especialização em Informática na Educação). Universidade Federal do Rio Grande do Sul, Porto Alegre*, 2006.

[33] M. E. Echeveste. Challenges of introducing computer science into the traditional grammar of k-12 schooling. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE '16, pages 359–359, New York, NY, USA, 2016. ACM.

[34] M. E. Echeveste and M. C. Martínez. Desafíos en la enseñanza de ciencias de la computación. *Virtualidad, Educación y Ciencia*, 7(12):34–48, 2016.

[35] M. Eichler. *Consensus organizing: Building communities of mutual self interest.* Sage Publications, 2007.

[36] M. S. El-Nasr and B. K. Smith. Learning through game modding. *Computers in Entertainment (CIE)*, 4(1):7, 2006.

[37] N. Firth. Code generation. *New Scientist*, 223(2985):38–41, 2014.

[38] A. Fowler. Informal stem learning in game jams, hackathons and game creation events. In *Proceedings of the International Conference on Game Jams, Hackathons, and Game Creation Events*, pages 38–41. ACM, 2016.

[39] A. Fowler and B. Cusack. Kodu game lab: improving the motivation for learning programming concepts. In *Proceedings of the 6th International Conference on Foundations of Digital Games*, pages 238–240. ACM, 2011.

[40] A. Fowler, T. Fristoe, and M. MacLaurin. Kodu game lab: a programming environment. *The Computer Games Journal*, 1(1):17–28, 2012.

[41] A. Fowler, F. Khosmood, and A. Arya. The evolution and significance of the global game jam. In *Proc. of the Foundations of Digital Games Conference*, 2013.

[42] A. Fowler, F. Khosmood, A. Arya, and G. Lai. The global game jam for teaching and learning. In *Proc. of the 4th Annual Conference on Computing and Information Technology Research and Education New Zealand*, pages 28–34, 2013.

[43] A. Fowler, G. Lai, R. Hill, F. Khosmood, and A. Arya. Organizing philosophies of game jams and game hackathons. In *In Proceedings of the Workshop on Game Jams, Hackathons and Game Creation Events*, 2015.

[44] N. Fraser et al. Blockly: A visual programming editor. *URL: https://code. google. com/p/blockly*, 2013.

[45] P. Freire. Pedagogy of the oppressed, trans. *Myra Bergman Ramos. New York: Continuum*, pages 65–80, 1970.

[46] T. Fristoe, J. Denner, M. MacLaurin, M. Mateas, and N. Wardrip-Fruin. Say it with systems: expanding kodu's expressive power through gender-inclusive mechanics. In *Proceedings of the 6th International Conference on Foundations of Digital Games*, pages 227–234. ACM, 2011.

[47] W. A. Fryer. Hopscotch challenges: Learn to code on an ipad! *Publications Archive of Wesley Fryer*, 1(1), 2014.

[48] R. M. Goodman, M. A. Speers, K. McLeroy, S. Fawcett, M. Kegler, E. Parker, S. R. Smith, T. D. Sterling, and N. Wallerstein. Identifying and defining the dimensions of community capacity to provide a basis for measurement. *Health Education & Behavior*, 25(3):258–278, 1998.

[49] K. Groos. *The play of man.* D. Appleton, 1908.

[50] D. B. Harlow and A. E. Leak. Mapping students' ideas to understand learning in a collaborative programming environment. *Computer Science Education*, 24(2-3):229–247, 2014.

[51] R. Hess. *The essential Blender: guide to 3D creation with the open source suite Blender.* No Starch Press, 2007.

[52] J. Huizinga. *Homo Ludens Ils 86.* Routledge, 2014.

[53] S. Hurtado, C. B. Newman, M. C. Tran, and M. J. Chang. Improving the rate of success for underrepresented racial minorities in stem fields: Insights from a national project. *New Directions for Institutional Research*, 2010(148):5–15, 2010.

[54] I. Illich. *Deschooling society.* Harmondsworth, Middlesex, 1973.

[55] L. Johnson and S. Adams. *Challenge Based Learning: The Report from the Implementation Project.* ERIC, 2011.

[56] L. F. Johnson, R. S. Smith, J. T. Smythe, and R. K. Varon. Challenge-based learning: An approach for our time. *New Media Consortium*, 2009.

[57] D. H. Jonassen and D. H. Jonassen. *Learning to solve problems with technology: a constructivist perspective.* Merrill, 2003.

[58] C. Jones. *An Exploration of a Graphical User Interface (GUI) to Facilitate the Creation of Internet Interventions.* PhD thesis, University of Dundee, 2014.

[59] Y. B. Kafai. Playing and making games for learning

instructionist and constructionist perspectives for game studies. *Games and culture*, 1(1):36–40, 2006.

[60] J. Kasurinen, S. Mirzaeifar, and U. Nikula. Computer science students making games: a study on skill gaps and requirement. In *Proceedings of the 13th Koli Calling International Conference on Computing Education Research*, pages 33–41. ACM, 2013.

[61] H. Koenitz. *Interactive Digital Narrative: History, Theory and Practice*. Routledge, 2015.

[62] M. Kölling and P. Henriksen. Game programming in introductory courses with direct state manipulation. *SIGCSE Bull.*, 37(3):59–63, June 2005.

[63] A. Kultima. Defining game jam. In *Proc. of Foundations of Digital Games Conference*, volume 15, 2015.

[64] N. Lazzaro. Why we play games: Four keys to more emotion without story. 2004.

[65] O. Lecarme and K. Delvare. *The book of GIMP: A complete guide to nearly everything*. No Starch Press, 2013.

[66] E. Lee. Enterprize 2.0, an introduction to socialtext.com, Apr 2010.

[67] A. S. Leh, B. Kouba, and D. Davis. Twenty-first century learning: communities, interaction and ubiquitous computing. *Educational Media International*, 42(3):237–250, 2005.

[68] J. Liu, C.-H. Lin, J. Wilson, D. Hemmenway, E. Hasson, Z. Barnett, and Y. Xu. Making games a snap with stencyl: a summer computing workshop for k-12 teachers. In *Proceedings of the 45th ACM technical symposium on Computer science education*, pages 169–174. ACM, 2014.

[69] J. Maloney, L. Burd, Y. Kafai, N. Rusk, B. Silverman, and M. Resnick. Scratch: a sneak preview [education]. In *Creating, connecting and collaborating through computing, 2004. Proceedings. Second International Conference on*, pages 104–109. IEEE, 2004.

[70] J. Maloney, M. Resnick, N. Rusk, B. Silverman, and E. Eastmond. The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, 10(4):16, 2010.

[71] J. H. Maloney, K. Peppler, Y. Kafai, M. Resnick, and N. Rusk. *Programming by choice: urban youth learning programming with scratch*, volume 40. ACM, 2008.

[72] C. Martinez, M. J. Gomez, and L. Benotti. A comparison of preschool and elementary school children learning computer science concepts through a multilanguage robot programming platform. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, pages 159–164. ACM, 2015.

[73] M. Mateas, P. Mawhorter, and N. Wardrip-Fruin. Intentionally generating choices in interactive narratives. pages 292–300, 2015.

[74] G. S. May and D. E. Chubin. A retrospective on undergraduate engineering success for underrepresented minority students. *Journal of Engineering Education*, 92(1):27–39, 2003.

[75] L. McCarthy, C. Reas, and B. Fry. *Getting Started with P5. js: Making Interactive Graphics in JavaScript and Processing*. Maker Media, Inc., 2015.

[76] W. McGugan. *Beginning game development with Python and Pygame: from novice to professional*. Apress, 2007.

[77] G. M. Novak, E. T. Patterson, A. D. Gavrin, and W. Christian. Just-in-time teaching blending active learning with web technology. 1999.

[78] M. Overmars. Teaching computer science through game design. *Computer*, 37(4):81–83, 2004.

[79] J. Pirker, D. Economou, and C. Gütl. Interdisciplinary and international game projects for creative learning. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*. ACM, 2016.

[80] J. Pirker, A. Kultima, and C. Gütl. The value of game prototyping projects for students and industry. In *Proceedings of the International Conference on Game Jams, Hackathons, and Game Creation Events*, pages 54–57. ACM, 2016.

[81] J. Pirker and K. Voll. Group forming processes-experiences and best practice from different game jams. In *Workshop Proceedings of the 10th International Conference on the Foundations of Digital Games (Pacific Grove, California, Asilomar Conference Grounds)*, 2015.

[82] J. A. Preston, J. Chastine, C. O'Donnell, T. Tseng, and B. MacIntyre. Game jams: Community, motivations, and learning among jammers. *International Journal of Game-Based Learning (IJGBL)*, 2(3):51–70, 2012.

[83] J. C. Read, S. MacFarlane, and C. Casey. Endurability, engagement and expectations: Measuring children's fun. In *Interaction design and children*, volume 2, pages 1–23. Shaker Publishing Eindhoven, 2002.

[84] C. Reas and B. Fry. *Processing: a programming handbook for visual designers and artists*. Number 6812. Mit Press, 2007.

[85] M. Resnick, J. Maloney, A. Monroy-Hernandez, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, et al. Scratch: programming for all. *Communications of the ACM*, 52(11):60–67, 2009.

[86] M. Rice. Leadership ensemble: Lessons in collaborative management from the world's only conductorless orchestra. *Nursing Management (Harrow)*, 14(9):7–8, 2008.

[87] L. P. Rieber. Seriously considering play: Designing interactive learning environments based on the blending of microworlds, simulations, and games. *Educational technology research and development*, 44(2):43–58, 1996.

[88] M. Roussou. Learning by doing and learning through play: an exploration of interactivity in virtual environments for children. *Computers in Entertainment (CIE)*, 2(1):10–10, 2004.

[89] K. Roy. App inventor for android: report from a summer camp. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education*, pages 283–288. ACM, 2012.

[90] K. Roy, W. C. Rousse, and D. B. DeMeritt. Comparing the mobile novice programming

environments: App inventor for android vs. gamesalad. In *2012 Frontiers in Education Conference Proceedings*, pages 1–6. IEEE, 2012.

[91] P. D. Sáez and C. E. Monsalve. Aprendizaje basado en resolución de problemas en ingeniería informática. *Formación universitaria*, 1(2):3–8, 2008.

[92] T. R. Sass. Understanding the stem pipeline. working paper 125. *National Center for Analysis of Longitudinal Data in Education Research (CALDER)*, 2015.

[93] V. L. Semenovich. *Thought and language*. MIT press, 2012.

[94] G. Siemens. Connectivism: A learning theory for the digital age. 2014.

[95] W. Slany. A mobile visual programming system for android smartphones and tablets. In *2012 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 265–266. IEEE, 2012.

[96] P. A. Smith and C. Bowers. Improving social skills through game jam participation. In *Proceedings of the International Conference on Game Jams, Hackathons, and Game Creation Events*, pages 8–14. ACM, 2016.

[97] K. T. Stolee and T. Fristoe. Expressing computer science concepts through kodu game lab. In *Proceedings of the 42nd ACM technical symposium on Computer science education*, pages 99–104. ACM, 2011.

[98] S. Tilton. etools: Using audacity in the classroom.

[99] B. Trilling and C. Fadel. *21st century skills: Learning for life in our times*. John Wiley & Sons, 2009.

[100] L. Tsui. Effective strategies to increase diversity in stem fields: A review of the research literature. *The Journal of Negro Education*, pages 555–581, 2007.

[101] A. Tugend. What it takes to make new college graduates employable. *The New York Times*, 28, 2013.

[102] D. Tyack and W. Tobin. The "grammar" of schooling: Why has it been so hard to change? *American Educational Research Journal*, 31(3):453–479, 1994.

[103] J. Varma. Codea. In *Learn Lua for iOS Game Development*, pages 279–302. Springer, 2013.

[104] L. S. Vygotsky. Play and its role in the mental development of the child. *Soviet psychology*, 5(3):6–18, 1967.

[105] L. S. Vygotsky. The problem of the environment. *The Vygotsky Reader*, pages 338–354, 1994.

[106] L. L. Werner, S. Campe, and J. Denner. Middle school girls+ games programming= information technology fluency. In *Proceedings of the 6th conference on Information technology education*, pages 301–305. ACM, 2005.

[107] M. White. The real reason new college grads can't get hired. *Retrieved Jul 9th*, 2013.

[108] D. William. Assessment, learning and technology: prospects at the periphery of control. In *Keynote speech at the Association for Learning Technology Conference in Nottingham, England Transcript: http://www. alt. ac. uk/docs/altc2007_dylan_wiliam_keynote_transcript. pdf*, 2007.

[109] A. Wilson, T. Hainey, and T. Connolly. Evaluation of computer games developed by primary school children to gauge understanding of programming concepts. In *European Conference on Games Based Learning*, page 549. Academic Conferences International Limited, 2012.

[110] D. Wolber. App inventor and real-world motivation. In *Proceedings of the 42nd ACM technical symposium on Computer science education*, pages 601–606. ACM, 2011.

[111] A. Y and D. A. Kolb. Learning to play, playing to learn: A case study of a ludic learning space. *Journal of Organizational Change Management*, 23(1):26–50, 2010.

[112] R. S. Zander and B. Zander. *The art of possibility*. Harvard Business Press, 2000.

Table 1: Overview of game development tools

Name	URL	Cost	OS	Affordances	Limitations	Lit.
Alice	alice.org	Free	WIN, OSX, LINUX	Mature and active online community, designed for teaching	Adding objects and characters is not so friendly. The distance between objects is difficult to calculate. Text editor (drag and pop) is small.	[23, 21, 24]
AppInventor	appinventor.mit.edu	Free	Browser	Object Based, Drag ad Drop, desiged for teaching. Mature and active online community. Easy access to mobile API	Limited to mobile games on Android. Blocks take up a lot of space and complex projects become unwieldy and hard to read. No export to other editors possible.	[110, 89] [11]
Codea	twolivesleft.com/Codea	Paid	iOS	Language Lua, can export app	Only iOS, difficult to code in a phone/tablet	[103]
Construct 2	scirra.com/construct2	Free*	WIN	Suitable for teaching, development and publishing on multiple platforms.	Projects can not be easily shared between authors. Difficult to collaborate.	[60]
Corona SDK	coronalabs.com	Free	WIN, OSX	Suitable for teaching, development and publishing on multiple platforms, Lua Language	Closed source, 2D	[15]
Defold	defold.com	Free	WIN, OSX, LINUX	Cloud based 2D game development tool. Exposts to Windows,iOS, Linux,and Androind	Limited to 2D games only	
Game Maker	yoyogames.com/gamemaker	Free	WIN	Suitable for teaching, development and publishing on multiple platforms.	Limited to 2D games	[78]
GameSalad	gamesalad.com	Paid	WIN, OSX	Object Based, Drag ad Drop, desiged for teaching. Export games to iphone and android (html5)	Limited to 2D.	[90]
Hopscotch	gethopscotch.com	Free	iOS	Focused on children, easy to use, visual programming.	Limited language, not possible to export,	[47]
Inkle Writer	writer.inklestudios.com	Free	Browser	Browser based, Platform Agnostic	Mostly text based	
Kodu	kodugamelab.com	Free	WIN	Designed for classroom use. Features pre-made lessons.	Limited options to personalize. Kodu is restricted to existing modules	[40, 39]
Minecraft edu	education.minecraft.net	Free for EDU	WIN, OSX, LINUX, iOS	Suitable for teaching CS Concepts using the Minecraft platform	Not really suited to making games	
P5.JS	p5js.org	Free	Browser	Suitable for designers and creative people, web version	Only web, need to convert using PhoneGap, lack of good editor.	[75]
Pilas Engine	pilas-engine.com.ar	Free	WIN, OSX, LINUX	Objected Based. Could included as python library. Python coding. Export to multiple platforms. Designed to teaching CS.	The platform and documentation are in Spanish. Starter community. Limited to 2D games.	
PlayMaker	hutonggames.com	Paid	WIN, OSX, LINUX	Suitable for teaching, development and publishing on multiple platforms.	Requires Unity.	
Pocket Code	catrobat.org	Free	Android	Visual programming language tools for smartphones, tablets, and mobile browsers similar to Scratch	Limited to mobile games and Android only	[95]
Processing	processing.org	Free	WIN, OSX, LINUX	Suitable for designers and creative people, multiplatform (Android including)	Difficult to export to Android, lack of good editor.	[84]
PyGame	pygame.org	Free	WIN, OSX, LINUX	Easy to use, Python Language	Only 2D, don't export to iOS	[76]
RPG Maker	rpgmakerweb.com	Paid	WIN	Easy to use, flexible	Genre specific	[27, 32]
Scratch	scratch.mit.edu	Free	Browser / WIN, OSX, LINUX	Object Based, Drag ad Drop, desiged for teaching. Mature and active online community.	Limitated to 2D games. Developing complex games add many complexities.	[85, 70, 69]
Scratch Jr	scratchjr.org	Free	Browser / Android, IOS	Object Based, Drag and Drop, desiged for teaching. Mature and active online community.	Image Based Language. Simple productions. Limitated to 2D games.	[50, 37]
Stencyl	stencyl.com	Free	WIN, OSX, LINUX	Object Based. Video games developed could be exported for computers, mobile devices, and the web (Flash).	Limitated to 2D games. To publish the game, you've to buy the version.	[68]
Twine	twinery.org	Free	WIN, OSX, LINUX, Browser	Browser based, Platform Agnostic	Mostly text based	[61, 73]

Table 2: Overview of art and audio tools

Name	URL	Cost	OS	Affordances	Limitations	Lit.
Audacity	audacityteam.org	Free	WIN, OSX, LINUX	Cross-Platform. Run in all kind of computers. Open Source. Big community.		[98]
Audition	adobe.com/products/audition.html	Paid	WIN, OSX		Not for all versions of OS. Paid.	
Blender	blender.org	Free	WIN, OSX, LINUX	Cross-Platform. Run in all kind of computers. Open Source. Big community.		[51]
Gimp	gimp.org	Free	WIN, OSX, LINUX	Cross-Platform. Run in all kind of computers. Open Source. Big community.		[65]
Illustrator	adobe.com/products/illustrator.html	Paid	WIN, OSX		Not for all versions of OS. Paid.	
Inkscape	inkscape.org	Free	WIN, OSX, LINUX			[4]
Photoshop	adobe.com/products/photoshop.html	Paid	WIN, OSX		Not for all versions of OS. Paid.	
Sketch	adobe.com/products/sketch.html	Paid	IOS		Not for all versions of OS. Paid.	
Sketchup	sketchup.com	Paid	WIN, OSX		Not for all versions of OS. Paid.	