**8**

**Design and project approaches in computing education**

Mick Chesterman

https://orcid.org/0000-0002-8005-2390

**Abstract**

The creative processes involved in undertaking computing projects in an educational setting have a significant potential to deliver transformative learning experiences. Pedagogies and frameworks to support design approaches in education are valuable for teachers to guide and maximise the experience of their learners. This chapter describes and explores some of the strategies that can be used to support the delivery of design- and project-based approaches. It covers the value of creative communities in the design process, key teaching techniques aligned with design-based approaches, the benefits and processes of project-based learning (PBL), and tactics for overcoming limitations in delivering PBL in the classroom.

**Introduction**

The creative processes involved in undertaking computing projects in an educational setting have a significant potential to deliver transformative learning experiences. Pedagogies and frameworks to support design approaches in education are valuable for teachers to guide and maximise the experience of their learners. This chapter describes and explores some of the strategies that can be used to support the delivery of design- and project-based approaches. To start, I focus on the value of creative communities in the design process. I then outline three teaching techniques aligned with design-based approaches to computing projects. The second half of the chapter takes a broad look at some of the benefits and processes of project-based learning (PBL) and ends with some tactics for overcoming limitations in delivering PBL in the classroom.

**The power of communities**

A project-based approach to learning coding and computing can be supported outside the boundaries of formal learning. For example, enthusiastic older family members may take young people to Maker Faire or engage in other community coding activities. Family members may buy creative computing kits or access resources such as YouTube videos or online forums for specialist interests like robotics, games or other forms of digital making. However, access to this kind of computer enthusiast community is not available to all young people. The following initiatives aim to address this by providing entry points to community-oriented coding activities.

**Code Clubs** are out-of-hours school clubs run by teachers or volunteers. Running a code club is a good way to build a lunchtime or after school community of coding enthusiasts. Code Club was originally an independent organisation but is now part of the Raspberry Pi Foundation. A large number of high quality, colourful and attractive resources are supplied free of charge on their website.[[1]](#endnote-2) The resources provided can be printed out and serve as step-by-step tutorials. Resources also contain challenges that encourage further experimentation by learners.

**Coder Dojos** are monthly events run by volunteers at weekends. They often focus on creative, engaging computing. I have volunteered at some events and been impressed by the dedication and inventiveness of the other volunteers. There is an inspiring diversity of activities at events. At a typical Coder Dojo, some learners will use existing resources to support multimedia coding in Scratch, while others will try out new and experimental work, perhaps hacking Minecraft, creating games with code engines or trying new physical computing projects. Often volunteers bring their own children, who act as guinea pigs and test activities before the volunteer brings them to a Coder Dojo. To find out more, visit the Coder Dojo website.[[2]](#endnote-3)

**The Coolest Project** is another project of the Raspberry Pi Foundation. It takes the form of an online competition and a real-world showcase that individuals or teams apply to take part in. The Coolest Project helps generate a framework and motivation for projects. This experience allows students to tackle problems in a radically different way than typical classroom teaching. Projects provide opportunities to engage in authentic coding practices, including designing for real users, collaboration with other students, project planning, debugging faulty code and repeated revisions to fine-tune the desired result.[[3]](#endnote-4)

Coder Dojos are family-focussed events that take place outside of school and are therefore less easy for teachers to engage with. They are, however, a good source of inspiration for teachers looking for creative project ideas for their classroom. In contrast, both Code Clubs and the Coolest Project are suitable for computing teachers to run inside schools. Running lunchtime or after-school projects, while not reaching all pupils, can be a great way to showcase the engaging and creative nature of hobbyist computing projects.

**Communities in educational theory**

The power of communities has been highlighted by academics as part of what is known as the social turn in education. This is a turn away from more individualised ways of learning that focus on the transfer of knowledge from the teacher to the pupil. Instead, the focus is on how learning happens through participation in communities and culture, or in other words a socio-cultural approach. Community in this educational context can motivate and provide support for participation in a creative process. Barbara Rogoff (1994), a researcher of socio-cultural approaches to education, has described an educational process she calls *Communities of Learners*. Rogoff sees this approach as radically different from both instruction-based models of learning and pure discovery learning (where learners are left to their own devices). In communities of learners participants have different levels of expertise and varied roles in a learning system working towards an authentic goal. Rogoff notes that observing this kind of learning can be confusing to teachers and parents used to more instruction-based approaches. Such a learning community in full swing can seem chaotic. However, complex and productive learning is happening in ways that we, as teachers, may be unused to. This chapter helps unpick some of these practices and explore ways that educators have structured their learning environments to take advantage of this powerful approach.

**Checklist - Applying a Community Approach in the Classroom**

Are you making the most of the power of communities in your classroom? Before you start your next unit of work, ask yourself some of the following questions.

* Are there regular opportunities for learners to work together during your unit of work? How often will students give and receive peer feedback?
* Are there examples of similar work from other students available for your students to examine and perhaps build upon?
* Can you draw on the roles or identities that students have adopted in their previous school or home activities? Are learners able to reflect on the specifics of those roles to contribute to the effectiveness of their engagement in teamwork?
* Can you help your learners make connections between their computing activities and other professional or enthusiast communities outside of the classroom?

**Design-based approaches in computing education**

Designing as a discipline involves a community of both producers and users. Design-based approaches have been widely adopted in software production, the creative industries and wider business contexts. These design principles and practices are also relevant to education. For example, we may be used to seeing students motivated by producing something for a real audience. Design projects allow students to develop important 21st century skills like problem solving and communication, and creatively responding to real life contexts. In the following sections I will explore the design-based approaches of iteration, design patterns and the Use-modify-create (UMC) model.

**Teaching technique – iterative design**

Iterative design involves repeatedly coming back to reflect on the initial outcomes of creative goals and revising them based on results. The process involves goal setting, creating quick prototypes, user testing and evaluation, revision and ongoing reflection. The process is iterative in that testing and revision of the prototype design can be repeated until the desired result is achieved. Iteration is also a key part of a more general scientific method of testing a hypothesis and revision based on your analysis of the results. The idea of a repeated (iterative) spiral approach that both deepens understanding and improves the end results is popular both in education and industry. In software and design industries, it is often referred to as design thinking and agile approaches. In education, this approach is referred to in concepts like the spiral curriculum and the promotion of student mastery.

Resnick (2012) describes the foundations of the design-based approaches in education as: engaging in design activities, exploring personally meaningful topics, collaborating with others and deepening understanding through reflection. The key reason to adopt these principles is to increase engagement via sustained participation in computing projects for a broad range of learners.[[4]](#endnote-5) One of the sources for sustained engagement is when, as part of the iterative process, learners are able to test and then revise their creation or experiment based on their own evaluation. Another factor is the importance of a community in the design process, as a real audience for creations, as a source of inspiration and as peer evaluators in the testing process. The above principles embody key elements of inclusive practices contained in universal design for learning (UDL – explained in more detail in Chapter 3), including allowing students to demonstrate their knowledge in a multitude of ways and allowing students to follow their own interests and motivations (Capp, 2017).

**Teaching technique – worked examples and design patterns**

Design patterns are most commonly used by computing students in higher education to teach object-oriented computing, but they are also useful for all levels of learners. Design patterns are rooted in real-life incidences of problems that are often solved in a particular way. They are concrete examples of coding principles in context. Design patterns can help the development of coding communities if more experienced coders take the time to document the patterns they use in an accessible way for novice coders. In computing education there are similarities between design patterns and a technique called *worked examples*. The National Center for Computing Education (NCCE) has promoted working examples as a classroom activity by creating a Quick Read document for teachers on the subject.[[5]](#endnote-6) Both worked examples and design patterns act as a way to demonstrate underlying principles in practice. For both approaches, showing working code used in a particular context helps students to analyse what makes it work and why it is a suitable solution.

For educators the use of design patterns and worked examples can help support learners develop coding proficiency by providing scaffolding and through modelling good design decisions. However, one of the challenges for teachers of using worked examples and design patterns is how to integrate them into student-led design challenges. You may be able to create a menu of printed or online patterns or examples that students can draw on as needed. Common examples can be modelled to the whole class when it is clear that many students will benefit from that approach.

**Teaching technique – the use-modify-create model**

The teaching technique UMC has the potential to both limit learner anxiety for novice coders and to scaffold the acquisition of coding and computational thinking concepts (Lee et al., 2011). A breakdown of each stage follows:

**Use:** In the *Use* stage, coders build familiarity with coding interfaces, code structures and syntax through scaffolded approaches that involve interacting with the program code and what it produces.

**Modify:** In the *Modify* stage, learners progress to working on real projects created by others. Learners deepen their knowledge of coding structures and practices by altering existing projects and templates to suit their own aims.

**Create:** After novice coders become more familiar with patterns of code design in use in the modify stage, they can progress to replicate such patterns in other code that they create from scratch.

A study involving five hundred 9- to 14-year-olds found that the UMC approach can balance a structured approach with more student-led exploration (Franklin et al., 2020). The researchers also found that the students enjoyed the UMC approach as they had more choice and agency in the process. Similarly, other research compared UMC with a starting-from-scratch approach and found higher student engagement for those in the UMC group (Lytle et al., 2019). The researchers found that because students using UMC had more time to play around with code, they were able to add their own personal touches and this ownership over the code sustained their continued engagement.

Researchers Kafai and Burke (2013) argue that a shift from writing programs from scratch to modifying and remixing them is in line with socio-cultural teaching approaches. They coin the term computational participation to reflect this change of focus. They also note that such remixing is helped by online coding communities, especially those aimed at novice coders like the Scratch community (described below). They encourage educators to avoid focussing solely on technical possibilities of coding environments to embrace the potential of online coding communities despite associated challenges. One challenge of teachers embracing remixing practices is to distinguish the legitimate remixing of work from less productive kinds of copying. In addition participation in online communities requires learners to concurrently build both technical and participatory skills which may also be challenging. The following case study examines ways in which the online Scratch community is facilitating design-based learning. It makes a strong case for the value of these approaches and asks how some of these benefits can be made more inclusive for learners who would struggle to take part in such a community independently.

**Case study – how the online scratch community supports design-based learning**

Scratch is educational software that uses a block-based coding approach and a set of tools to develop audio and graphical assets to help in the creation of multimedia coding projects.[[6]](#endnote-7) Scratch has an extensive community with over 75 million users who have created 80 million projects. Activity increased during COVID restrictions in 2020 and 2021, with over 20 million user comments in the month of March 2021 alone. The online community allows young creators to connect with others to share and get feedback on their work. Such community interaction sustains repeated effort to build student mastery in the form of fluency in the design and coding processes. Here are some of the key features of the online Scratch community, along with tips to integrate them into your computing teaching.

**High diversity of creations:** The process of keeping such a large community up and running and safe for young people requires a lot of resources. However, the effort is justified, as it has become an extremely rich source of inspiration for young creators. A simple search of the site for projects like games, creative greeting cards, storytelling projects and pretty much any digital product you can imagine will yield a multitude of results. As teachers we can draw on this resource to demonstrate diverse creations and encourage our learners to adapt existing work based on their own interests.

**Diverse ways to participate:** Learners can engage with the online community in a great variety of ways. Learners may just play or comment on the games of others. They may use it to create their own projects but not engage in the more social elements of the creative process. They may, like a smaller section of the community, become extremely active in creating and collaborating with others on shared projects.

**Encouraging project iteration:** Scratch encourages the remixing of others’ projects and makes it easy to create different versions of your own projects. This encourages sharing drafts for feedback via peer comments. This public sharing and feedback have been shown to encourage the development of new features in users’ creations. In the classroom, teachers may need to balance the more disruptive possibilities of the ability to publish inappropriate material online with the potential to build student autonomy and reflection.

**Supportive and authentic audience of fellow creators:** Due to the high numbers involved in the online Scratch community, there is a good chance of finding peers who are also interested in specific subject matter. Peer collaboration between community users is motivated by these shared interests. The potential and depth of collaboration in this community can be impressive. Roque and colleagues (2016) have described this in detail.[[7]](#endnote-8) The researchers describe how individuals find each other and group together by forming *studios* and then recruit other members to work on joint projects. This is sophisticated behaviour, which mimics real production processes. It is carried out by young people with a high degree of independence. However, the researchers concede that such collaborative production is only carried out by a very small proportion of the online creators. Teachers should be aware of a key challenge identified by the researchers, namely, how to replicate the benefits of collaborative community activity for young people who have less experience or confidence. In response, Roque (2016) went on to develop related programmes including online project exhibitions, competitions and off-line family-based projects to engage under-represented groups. As educators, we can take inspiration from this process of replicating the highly engaged, organic feedback and support of the chaotic online community into a more offline and structured design-based environment. The second half of this chapter addresses ways you may be able to rise to this challenge using project-based approaches.

**Reflection point – using design-based approaches in the classroom**

You can ask yourself the following questions to try to help you use some of the beneficial aspects of design-based approaches in your classroom.

* Are learners able to explore exemplar materials to inspire and shape their creative ideas?
* As they plan, are learners able to think about and articulate the perspective of the real or imagined users of the designed projects?
* Are learners helped to come up with ideas through ideation techniques that scaffold the creative process?

**Follow-up resources:** I have created several online courses that explore hands-on ways to use design thinking in education and community work as part of Manchester Met’s Rise initiative, which provides students with opportunities to gain new skills.[[8]](#endnote-9)

**Projects and project-based learning (PBL*)***

PBL is a wide set of approaches that seek to facilitate learning through undertaking practical projects. Students often complete projects in groups. Students develop knowledge and skills in the context of a real or simulated problem that they must solve. Using projects to support learning is one of the 12 teaching principles advocated by NCCE.[[9]](#endnote-10) In the next section, I cover the characteristics and potential of PBL.

Researchers Blumenfeld and colleagues (1991) argue that school disengagement is caused by work that bores students. They found that project work incorporating learner choice and involving real outputs is motivating and can sustain student engagement. The pervasiveness of digital products in our lives offers a wide range of choices for computing projects, including websites, games, wearable technology, phone apps, robotics and other physical computing projects. Thus, computing education is an excellent vehicle for a project-based approach to learning. However, Blumenfeld’s research cautions that implementing PBL in classrooms is not straightforward. I cover barriers to PBL and ways to overcome them in the final part of this chapter.

Academics have worked with expert practitioners to create PBL frameworks to help teachers plan and deliver projects and recognise the complexity of some of the learning that takes place. The following outline of PBL elements is a synthesis of several of these frameworks with additional commentary on how this may apply to computing projects.

**Challenge:** The focus of the project should be a relatable problem or question that does not have one straightforward solution. Software and electronics projects that fit this brief are thus very suitable candidates.

**Authenticity:** The real-life relevance of projects helps engage students as they make connections to their interests and communities. As mentioned in the section above, many forms of coding projects, including phone apps, websites and games, have real and relatable goals.

**Sustained and collaborative work:** Adequate time must be allocated for sustained project work. Students should work together and be given the chance to revise their projects. This is perhaps one of the greatest challenges to delivering computing projects in a school setting.

**Public project:** The creation of a shareable, public object helps learners focus and design for others. It can also act as a focus for discussion within the classroom. Sharing computing projects within the classroom could be supplemented by presentations to a wider audience.

**Student voice and choice:** Giving student’s choice over the focus of their project increases their engagement. Participation in discussions about project direction builds student autonomy. The high level of saturation of digital products into the experience of many young people’s day-to-day lives can help shape students’ interests.

**Reflection and critique:** Self-reflection may be informal at times but also guided by classroom processes like learning journals. It can also involve peer feedback or input beyond the classroom to bring authentic perspectives. Reflection could also happen in digital form via an online journal or templated digital documents.

**Teaching technique: Supporting PBL in computing classrooms**

PBL and other design approaches are aligned with UDL in many ways. For example, they all benefit from a collaborative community and a real or imagined audience for public products as a way to increase learner motivation. They also advocate for structures for students to chart their progress and support feedback. One of the central tenets of these approaches, the importance of learner choice in projects, is also one of the most challenging for teachers to enact. One critique of PBL, especially where it involves student experimentation and student discovery, is that it can be chaotic and that it is therefore challenging to communicate high-level concepts. PBL also requires skills, support and planning that are very different from traditional teaching and may therefore be difficult for teaching staff to implement. For example, practitioners must build their ability to switch between facilitating students operating freely to then guiding them in the process of revision and critique. Having resources and clear stages in your project plan to help this process is vital. This section outlines the typical stages of PBL and how to adapt them to a computing context.

**Start with a driving question or mission:** The project goal for computing projects is often to create a digital product in response to a need or design brief for a specific audience. Add detail and link to real-world problems at this stage to maximise learner engagement. Decide the limits for students projects and outline these clearly from the start to avoid having to dampen their enthusiasm. For example, if creating a 2D game instead of a 3D one is better suited to the technical abilities of students, then be clear about that limit from the start of the project.

**Designing a plan and resources for the project:** Decide what part of the curriculum the project work will develop. Use your knowledge of the curriculum to put resources in place to support the learners as they undertake the project. Not everything needs to be explicitly taught if you can signpost your learners to those resources. Having an online or paper-based resource bank for students to access and navigate can be extremely useful.

**Monitor pupil progress:** As the project unfolds, keep students on track by having a realistic schedule for project stages. Check that you are consistently signposting students to the relevant resources for the project choices they have made and for the stage that they are currently undertaking.

**Assess emerging project processes and outcomes:** Ongoing feedback and assessment are vital. Build in opportunities for reflection, peer feedback and revision. Can students share prototypes of their digital products? Can you support them to recognise if they are working effectively as a team? How can you support them in making connections to the underlying curriculum knowledge?

**Evaluation:** You may evaluate the end piece of work created by students, the way they have worked together and the skills used to undertake different stages of the project. You can validate what the students have learned and identify areas for future development.

If you would like detailed information and case studies on PBL, there are online resources provided by numerous organisations, including Edutopia,[[10]](#endnote-11) PBLWorks[[11]](#endnote-12) and the UK-based Edge Foundation.[[12]](#endnote-13)

**Creatively overcoming limitations to PBL**

This section considers barriers to PBL and provides tips and strategies that have been used by other educators and researchers to overcome them in the context of computing.

**Sustaining the effort – Time challenges:** In research on barriers to undertaking projects in schools, teachers commonly cite time restrictions due to curriculum pressures. Resnick and Rusk (2020) suggest that, if possible, double lessons are helpful for hands-on coding work and to allow the design process time to unfold. They also advocate that at times a whole term should be devoted to undertaking a project. This sustained effort allows pupils to return to tweak and improve trickier coding and design challenges, thus supporting an iterative approach. In addition, cross-curricular projects may free up more time by linking with other subjects that are allocated more time, especially in KS3. For example, you could link a computing project with mathematics by asking students to create a game that teaches mathematical concepts. This could both deepen students’ learning of particular mathematical concepts and allow for the kind of repeated hands-on practice that builds coding fluency.

**Advocating the value of PBL for inclusion:** As with design-based approaches, PBL aligns well with UDL, specifically in the way students can bring their own interests into their creation of a public project. Both frameworks encourage teachers to guide and support project work with scaffolding and ongoing feedback. As educators, we can highlight the importance of creating inclusive classroom environments to our line managers and advocate for the time, training and resources needed to undertake project-based learning.

**Artefact-based assessment:** The tension between rote-learning approaches that may be used to prepare students to reproduce knowledge in written exam questions and the need for more fluid programming experiences raise an important question. How can some of the more flexible techniques for observing and assessing learner’s progress be brought into classroom practice? As a possible response, the NCCE promotes the use of artefact-based questions (ABQs) to assess project work. ABQs are questions based on the digital or physical artefacts that students create as part of their projects. ABQs allow teachers and students to link the problems students have encountered and solved in their computing projects to the requirements of the computing curriculum. Teachers can focus on specific areas of the students’ work and ask about details of the code structure and implementation. ABQs can also address design issues and processes. For example, how the project outcomes compare to the original goals, how feedback was implemented, group work and overcoming challenges and the design challenges.

**Authenticity of projects:** As mentioned before, computing is blessed with the potential to create digital and physical projects that are recognised by or relevant to young people. However, sometimes the process of finding authentic audiences and processes to motivate learners within a school setting is not simple. The following activity may help bring authenticity to student projects.

**Activity – Meeting the Challenge of Authenticity in the Classroom**

Here are some tactics you may be able to use to link projects to real issues and activities beyond the classroom.

* Draw on community members to set a local challenge that resonates with your learners.
* Use other members of staff in other subject areas to pose a school-based problem. This could be subject-specific or a pastoral or cross-curricular issue.
* Establish links with industry or social enterprises to set an authentic challenge within a work context.

If you draw on experts, staff or community members, they do not need to be there for the full term of the project. You can use visits or video calls at the start and end of the project. Most importantly, be sure to draw on the experience of students and use their ideas to shape possible responses to the challenge in the early stages.

**Conclusion**

Motivation, participation and peer learning do not happen in a vacuum. In this chapter, we have explored the value of creating a learning community of coders working on projects that are both authentic and linked to their own interests. To help this happen, we can draw on some of the rich research and resources available from different streams of practice, including PBL, UDL and design-based approaches. What many design and project-based approaches have in common is their focus on learner choice, sustained hands-on making and frameworks for facilitation, observation and assessment. For an accessible and convincing summary of project-based approaches and their adoption in a classroom setting see the review by Barron and Darling-Hammond (2008).

We have explored the tension between creative processes involving learner choice and teaching the more prescriptive requirements of the computing curriculum. To help bridge this gap the NCCE have created resources drawing on socio-cultural research to offer guidance on PBL, observation and pair programming. As teachers seeking to close an achievement gap between higher and lower achieving students, we have a responsibility to integrate such methods and evaluate their potential for our learners. I hope that this chapter has encouraged you to keep exploring authentic coding practices in the classroom and to share your experiences with others. To continue this journey, there are many forums where teachers share practice; these include CAS forums, blogs, Twitter posts and so on. In particular, there is a Twitter hashtag called #casinclude that encourages computing teachers to share inclusive practices. To fully explore the potential of projects, let’s share how we have used design and PBL approaches in our work and encourage others to do the same.

**Notes**

**References**

Barron, B., Darling-Hammond, L., 2008. Teaching for meaningful learning: A review of research on inquiry-based and cooperative learning. Book excerpt. George Lucas Educational Foundation.

Blumenfeld, P.C., Soloway, E., Marx, R.W., Krajcik, J.S., Guzdial, M., Palincsar, A., 1991. Motivating project-based learning: sustaining the doing, supporting the learning. Educational Psychologist 26, 369–398. https://doi.org/10.1080/00461520.1991.9653139

Capp, M.J., 2017. The effectiveness of universal design for learning: a meta-analysis of literature between 2013 and 2016. International Journal of Inclusive Education 21, 791–807. https://doi.org/10.1080/13603116.2017.1325074

Franklin, D., Coenraad, M., Palmer, J., Eatinger, D., Zipp, A., Anaya, M., White, M., Pham, H., Gökdemir, O., Weintrop, D., 2020. An Analysis of Use-Modify-Create Pedagogical Approach’s Success in Balancing Structure and Student Agency, in: Proceedings of the 2020 ACM Conference on International Computing Education Research. Presented at the ICER ‘20: International Computing Education Research Conference, ACM, Virtual Event New Zealand, pp. 14–24. https://doi.org/10.1145/3372782.3406256

Kafai, Y., Burke, Q., 2013. The Social Turn in K-12 Programming: Moving from Computational Thinking to Computational Participation, in: Proceedings of the 44th ACM Technical Symposium on Computer Science Education. https://doi.org/10.1145/2445196.2445373

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., Werner, L., 2011. Computational thinking for youth in practice. ACM Inroads 2, 32–37. https://doi.org/10.1145/1929887.1929902

Lytle, N., Cateté, V., Boulden, D., Dong, Y., Houchins, J., Milliken, A., Isvik, A., Bounajim, D., Wiebe, E., Barnes, T., 2019. Use, Modify, Create: Comparing Computational Thinking Lesson Progressions for STEM Classes, in: Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education. Presented at the ITiCSE ‘19: Innovation and Technology in Computer Science Education, ACM, Aberdeen Scotland UK, pp. 395–401. https://doi.org/10.1145/3304221.3319786

Resnick, M., 2012. ScratchEd: Working with Teachers to Develop Design-Based Approaches to the Cultivation Of Computational Thinking. MIT. https://web.media.mit.edu/~mres/proposals/NSF-ScratchEd.pdf

Resnick, M., Rusk, N., 2020. Coding at a crossroads. Communication of the ACM 63, 120–127. https://doi.org/10.1145/3375546

Rogoff, B., 1994. Developing understanding of the idea of communities of learners. Mind, Culture, and Activity 1, 209–229. https://doi.org/10.1080/10749039409524673

Roque, R.V., 2016. Family creative learning : designing structures to engage kids and parents as computational creators (Thesis). Massachusetts Institute of Technology.

Roque, R., Rusk, N., Resnick, M., 2016. Supporting Diverse and Creative Collaboration in the Scratch Online Community, in: Cress, U., Moskaliuk, J., Jeong, H. (Eds.), Mass Collaboration and Education. Springer International Publishing, Cham, pp. 241–256. https://doi.org/10.1007/978-3-319-13536-6\_12

Figure 8.1 Diagram of creative learning spiral Imagine – Create – Play – Share – Reflect – Imagine.

1. https://coderdojo.com/ [↑](#endnote-ref-2)
2. https://projects.raspberrypi.org/en/codeclub [↑](#endnote-ref-3)
3. https://online.coolestprojects.org/ [↑](#endnote-ref-4)
4. https://mres.medium.com/ten-tips-for-cultivating-creativity-fe79e7ebb83e [↑](#endnote-ref-5)
5. https://blog.teachcomputing.org/using-worked-examples-to-support-novice-learners/ [↑](#endnote-ref-6)
6. Scratch is available as a free download at https://scratch.mit.edu/ [↑](#endnote-ref-7)
7. http://tiny.cc/scratch-community [↑](#endnote-ref-8)
8. https://rise.mmu.ac.uk/category/enterprise/design-thinking/. [↑](#endnote-ref-9)
9. https://teachcomputing.org/pedagogy [↑](#endnote-ref-10)
10. https://www.edutopia.org/project-based-learning [↑](#endnote-ref-11)
11. https://my.pblworks.org/ [↑](#endnote-ref-12)
12. https://www.edge.co.uk/edge-future-learning/project-based-learning/ [↑](#endnote-ref-13)