“Yeah, But is it Art?” How many times do we hear this? In magazines, web forums, Twitter, TV and radio: from Damien Hirst’s pickled sharks and Tracey Emin’s unmade bed, to whether videogames are art. The argument is never ending and often bitter. Thank goodness for that! Imagine how much more boring life would be if we all suddenly agreed. As a society, we have agreed to disagree about art, and the resulting arguments drive innovation enhancing our world.

Back in the 1950s, the philosopher Gallie introduced the idea of essentially contested concepts: concepts that lead to endless arguments that can’t be settled by logic or collecting evidence, ones where everyone claims the others are using the concept the wrong way. There are things in life that we are meant to disagree about! How interesting is that? And scary, especially for scientists trying to study such a disputed concept.

It’s no wonder that nobody has come close to defining what creativity is, or to explaining why some people are more creative than others: it is an essentially contested concept. We need to disagree about creativity so it can be a driving force for change. But it raises the serious question of how to get software to act in creative ways, and to get society to accept machines are creative, if we’re not prepared to define what creativity is. Most other areas of Artificial Intelligence are based on concrete definitions, with increasingly sophisticated software written to perform intelligent mini-miracles safely within those definitions. But, in Computational Creativity research, if we embrace discord and uncertainty about the key concept, how can we proceed scientifically?

All is not lost. Some of us have been concentrating on the perceptions that people have of software being creative or not, with specific emphasis on the ‘or not’ part. It’s easy to dismiss software as not being creative, and people’s reasons tell us how we can change their views, enabling them to appreciate more what software does, and what wonderful things it produces. There is no creativity gene or algorithm, only the perceptions of people. Think of a white wall. In the day, it’s obviously a white wall, isn’t it? Go back at night with the lights off, and that white wall will be as black as can be. The change in circumstances has changed the perception you have of it. The same is true of human qualities, like being funny or being creative. And the same is true of computer programs. So, if we can make it hard for people to perceive software as not creative, then maybe, just maybe, they might one day be prepared to call it ‘creative’.

One argument is that software is uncreative, because the programmer supplies all the intentionality. “You write and run the software”, they say, “it’s you that wants to create something – so it’s not being creative: you are!” And they are right, but that can change. By taking these criticisms, we can write software that addresses the issues and makes it difficult to claim that the program is not creative (for example, see the Painting Fool, on the opposite page, and intentionality), “it’s still not creative”, they say, “because it’s…erm… I mean, it’s not…erm…”. That’s the hope, anyway. Only time will tell.