Research Project - Revised Presentation 2

Understanding Code through Visualisation

[SLIDE 1]

Introduction
-code is often difficult to quickly understand
-some observers may lack the experience to understand the software or the programming process

-how can we improve source code comprehension?
-how can we aid understanding of the programming process?
-better yet, how can we better communicate the programmers intention?

-techniques such as modelling or code documentation aren't dynamic or flexible
-don't allow for close to realtime understanding
-an effective technique is the use of visualisations
-it would be valuable to use visualisations as a means to communicate the programmer's intention

[SLIDE 2]

Project (what is the project about?)
-this project will explore code visualisations
-specifically, it will investigate visuals within the domains of software and music
-will be using *live coding* as a platform and case study for this (will discuss later)
-will develop and test code visualisations on audiences with audiences of varied levels of experience with programming, addressing code comprehension

[SLIDE 3]

Visualisations
-but first; I just want to run through a couple of existing visualisations with the domains of software and music
-show what role visualisations play in the understanding of specific domain knowledge
Software Visualisations
-gource - shows program file structure, based on data from code repository, shows dev progress
-scheme bricks - shows program structure clearly, relevant components light up when code is run

[SLIDE 4]

Music Visualisations
-frequency spectrum visualisations - frequency spectrum while playing music
-midi visualiser - relevant to its domain, clear representation of purpose

-key question is...

[SLIDE 5]

Big Question
-can we combine these domain-specific visualisations in a meaningful way?
-how can we use this to aid in code comprehension?

[SLIDE 6]

Live Coding
-live coding is a platform for bridging these two domain visualisations
-what is live coding?
-method of programming in front of an audience for artistic or informative purposes

-the live coder displays their screen to an audience, showing their code as they are working on it building a functional program
-makes use of interactive programming environments
-program running while changes are being made
-often focusses on improvisation - the programmer often has to think on their feet

-what does live coding achieve?
-gives the audience insight into the programming process - i'll be taking advantage of this

-I'll run a quick video

[Video]

-a typical audience may have no previous exposure to programming - may have a hard time understanding
-we need the audience to be able to get a better sense of what is going on
-we need a way to better convey the programmers intent

-this will be the goal of the project

[SLIDE 7]

Timeline and Methodology
-survey to get initial understanding of public audience - [plug weekend performance]
-develop taxonomy (categorising existing visualisations, determine best approach)
-and develop a variety of visualisations (April)
-first lab study examining initial visualisations (start of May)
-iterative process of refining visualisations (June)
-second lab study testing refined visualisations (start of July)

-overall methodology following IEEE definition of software engineer - "the application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software".

[SLIDE 8]

Feasibility
-in terms of schedule, some good progress has been made so far
-realistic timeline
-enough time to refine visualisations and plan studies
-studies should be completed before August ready for write up
-good basis in literature -
-rich potential - application of visuals to live coding not yet formalised/analysed

[SLIDE 9]

Questions?