Examples of some cool HCl projects

by Pooya Khaloo

November 21\textsuperscript{th} 2019

University of Massachusetts Amherst
• Human-Computer Interaction (HCI) is the study of how people use computers throughout their lives.

• HCI research seeks to develop user interfaces that are useful, usable, and enjoyable.

• It focuses on activities ranging from design to development to evaluation of computer systems, with a goal of understanding how computers and technology affect people and society.
Outline

- My masters project – Code Park
- My PhD project – CommunityClick
- Math Boxes
- Electrick
- LumiWatch
Purpose of showcasing Code Park

• How you can improve a project through an iterative design process?
• How to design a user study to evaluate your system?
Introduction
Problem

- IDEs have been around for decades and they are all inherently 2D
  - Writing code
  - Learning new language
  - Understanding a new codebase
- Not an issue for expert users
BUT

How about the novice users?
Learning new Codebase:

01 Uses memory
02 Has cognitive load
03 It is boring
Code Visualizers

• Assist developers in gaining insights into a codebase
• Make learning and understanding code easier
Design Process

Code House

Code Park: 3D Visualization Tool

User Study1

User Study2

User Study3

Code Park V2: 3D IDE
Code House
• Each class is attached to wall as a wallpaper

• Each wallpaper is syntax aware

• User explicitly define location of each wallpaper

• User can freely walk in the house environment
The House is fixed model

Each room has different color and furniture

It helps user remember where they are and location of each wallpaper
Why we started with house?

Can the users remember where they placed class X in this code-filled house, similar to the way they remember where they left their toothbrush in real life?
Navigation

• Mini Map
• Follow Path
• Follow Path, automated
• Teleport
Teleport
Code House Evaluation
Can users remember where they put their specific code in the house and which navigation methods is better to help them remembering?
User Study

- 5 participants (all male ranging in age from 22 to 28)
Result

• 3/5 remembered the location of all 8 classes

• Other 2 forgot location of classes they reached with Teleport

• They scored each navigation method between 1 - 10
Post questionnaire result
Discussion

• Positive answer to our main question
• Appealing and fun (Based on participants’ opinions)
• Not easy to use
• Not expandable
• Leads to next version …
Code Park: 3D Visualization Tool
Rooms create automatically
Each room represents a class
Files in a same directory resulted in adjacent rooms
Size & color of each room represent size of the class
This Version supports C#
Code Park supports different views
God View

Users can see all the classes

Users can choose each class for more details

Transfer user to **Player View** by selecting each room
Users can walk freely between classes like FPS games.

Users can see codes in detail.

Transfer user to **Code View** by clicking on code canvas.
[Serializable]
public class GameEngine
{
    const int maxLevel = 3;
    const int point = 2;
    public CardBack[,] CardBack { get; set; }
    public CardFace[,] CardFace { get; set; }
    public Player Player { get; set; }
    public List<CardPosition> Check { get; set; }
    public int Level { get; set; }

    public GameEngine(CardBack[,] cardBack, CardFace[,] cardFace, Player player, int level)
    {
        this.CardFace = cardFace;
        this.CardBack = cardBack;
        this.Player = player;
        this.Check = new List<CardPosition>();
        this.Level = level;
    }
}
All transition between God View, Player View and Code View is animated to maintain the users’ sense of spatial awareness.
Use syntax parsing to get list of methods

Show these methods both on top of classes in God View

And on wall in the room in Player View
With syntax parsing we also add Go To Definition

Quickly transfer to the location inside a file where a user-defined type or a variable is defined for the first time

This transfer is also animated
Code Park Evaluation
Code Park Goals

• Ease of use
• Help with code understanding
• Being fun and engaging
Comparing Visual Studio with Code Park
User Study

• 28 participants (22 males and 6 female ranging in age from 18 to 31 with a mean age of 22.8)
User Study

• To avoid learning effect we used two different codebases
• Therefore we had four groups:

<table>
<thead>
<tr>
<th>Group Name</th>
<th>First Tool</th>
<th>First Code Base</th>
<th>Second Tool</th>
<th>Second Code Base</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Visual Studio</td>
<td>LM</td>
<td>Code Park</td>
<td>MG</td>
</tr>
<tr>
<td>B</td>
<td>Code Park</td>
<td>MG</td>
<td>Visual Studio</td>
<td>LM</td>
</tr>
<tr>
<td>C</td>
<td>Visual Studio</td>
<td>MG</td>
<td>Code Park</td>
<td>LM</td>
</tr>
<tr>
<td>D</td>
<td>Code Park</td>
<td>LM</td>
<td>Visual Studio</td>
<td>MG</td>
</tr>
</tbody>
</table>
Tasks

• T1: Find a valid username to login into the program.
• T2: Find an abstract class in the code base.
• T3: Determine the relationship between classes A and B.
• T4: Find a designed bug that causes a program crash.
• T5: Pinpoint a reasonable location in the code for adding the necessary logic to support feature X.
Metric

• Quantitative:
  • Each task was timed

• Qualitative:
  • Post Task Questionnaire (Likert scale 1 - 7)
  • Post Study Questionnaire (Choose between VS/CH)
Result (Quantitative)

- Beginners took more time to finish their tasks

Mean time to task completion by experience level
Result (Quantitative)

• Task took more time to complete with Code Park
• Code Park has animation
• 3D interaction is slower
• First time using Code Park

Mean time to task completion based on the tool used
Result (Qualitative)

<table>
<thead>
<tr>
<th>Post Task Questionnaire</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1 I found it easy to work with Code Park/Visual Studio.</td>
</tr>
<tr>
<td>Q2 I found it easy to become familiar with Code Park/Visual Studio.</td>
</tr>
<tr>
<td>Q3 Code Park/Visual Studio helps me become familiar with code base’s structure.</td>
</tr>
<tr>
<td>Q4 It was easy to navigate through the code with Code Park/Visual Studio.</td>
</tr>
<tr>
<td>Q5 It was easy to find the definition of some variable with Code Park/Visual Studio.</td>
</tr>
<tr>
<td>Q6 How much did you like Code Park/Visual Studio?</td>
</tr>
<tr>
<td>Q7 How did you feel when using the tool?</td>
</tr>
<tr>
<td>Q8 It was easy to find what I wanted in the code using Code Park/Visual Studio.</td>
</tr>
</tbody>
</table>

Mean responses to the post task questionnaire for Code Park and Visual Studio
Result (Qualitative)

All participants unanimously agreed that Code Park is **fun** to use.

<table>
<thead>
<tr>
<th>Post Study Questionnaire</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SQ1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQ2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQ3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQ4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQ5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQ6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQ7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQ8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQ9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQ10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQ11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQ12</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question</th>
<th>Visual Studio</th>
<th>Code Park</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQ1</td>
<td>21</td>
<td>7</td>
</tr>
<tr>
<td>SQ2</td>
<td>8</td>
<td>20</td>
</tr>
<tr>
<td>SQ3</td>
<td>16</td>
<td>12</td>
</tr>
<tr>
<td>SQ4</td>
<td>17</td>
<td>11</td>
</tr>
<tr>
<td>SQ5</td>
<td>0</td>
<td>28</td>
</tr>
<tr>
<td>SQ6</td>
<td>11</td>
<td>13</td>
</tr>
<tr>
<td>SQ7</td>
<td>5</td>
<td>23</td>
</tr>
<tr>
<td>SQ8</td>
<td>7</td>
<td>21</td>
</tr>
<tr>
<td>SQ9</td>
<td>19</td>
<td>9</td>
</tr>
<tr>
<td>SQ10</td>
<td>15</td>
<td>13</td>
</tr>
<tr>
<td>SQ11</td>
<td>19</td>
<td>9</td>
</tr>
<tr>
<td>SQ12</td>
<td>16</td>
<td>11</td>
</tr>
</tbody>
</table>
• Participant 1: “I enjoyed a lot as it was the first time I was viewing the code in 3D environment.”

• Participant 6: “It is much easier to understand the overall code structure.”

• Participant 17: “It is very friendly and easy.”

• Participant 19: “We could easily access classes and methods.”

• Participant 26: “The use of spatial representation was well used.”
Discussion

• We achieved our goals
  • Ease of use
  • Help with code understanding
  • Being fun and engaging

• Users ask for two important features
  • Editing code
  • Compiling code

• Leads to next version …
public void borrowBook(String title)
{
    if (books.containsKey(title)) {
        if (!books.get(title).isBorrowed()) {
            books.get(title).borrow();
            System.out.println("You have successfully borrowed " + title);
        } else {
            System.out.println("Sorry, this book is already borrowed.");
        }
    } else {
        System.out.println("Sorry, this book is not in our catalog.");
    }
}

d // additional code...
Add support for Java

Users can create new project

Users can add classes, replace and delete them

Users can continue working on their project later
public void borrowBook(String title) {
    if (books.containsKey(title)) {
        if (!books.get(title).isBorrowed()) {
            books.get(title).borrow();
            System.out.println("You have successfully borrowed " + title);
        }
    } else {
        System.out.println("Sorry, this book is already borrowed.");
    }
}

public void returnBook(String title) {
    if (books.containsKey(title)) {
        if (books.get(title).isBorrowed()) {
            books.get(title).returnBook();
            System.out.println("You have successfully returned " + title);
        }
    } else {
        System.out.println("Sorry, this book is not in our catalog.");
    }
}
Terminal
Compiled without error

Compiled program's output

Compiled with errors
public class Book {
    private String title;
    private boolean isBorrowed;

    ; expected
    this.isBorrowed = true
}

public String title() {
    return this.title
}

public boolean isBorrowed() {
    return this.isBorrowed;
}

public void borrow() {
    // method implementation
}
public void borrowBook(String title)
{
    if (books.containsKey(title)) {
        if (!books.get(title).isBorrowed()) {
            books.get(title).borrow();
            System.out.println("You have successfully borrowed " + title);
        } else {
            System.out.println("Sorry, this book is already borrowed.");
        }
    } else {
        System.out.println("Sorry, this book is not in our catalog.");
    }
}
User Study Goal

• Evaluate our two new features
• Evaluate usability of this version
User Study

- **9 participants** (8 males and 1 female ranging in age from 18 to 29 with a mean age of 22.8)

- Pre Questionnaire
- Tutorial & Install Code Park on their laptop
- Develop an assignment within a week

- Post Questionnaire
- Organize an existing project in Code Park
- Explain their reason
All the participants finished the assignment.

Some of them finished with three classes and organize them as a triangle shape.

Other finished it with only two classes and organize them in a line.
<table>
<thead>
<tr>
<th>Question</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>I found it easy how to use Code Park.</td>
</tr>
<tr>
<td>Q2</td>
<td>I found it easy how to learn using Code Park.</td>
</tr>
<tr>
<td>Q3</td>
<td>I think that I would like to use Code Park frequently.</td>
</tr>
<tr>
<td>Q4</td>
<td>I found the various functions in Code Park were well integrated.</td>
</tr>
<tr>
<td>Q5</td>
<td>It was easy to navigate through the code with Code Park.</td>
</tr>
<tr>
<td>Q6</td>
<td>It was easy to write code in Code Park.</td>
</tr>
<tr>
<td>Q7</td>
<td>It was easy to work on a project with Code Park.</td>
</tr>
<tr>
<td>Q8</td>
<td>How much did you like the Code Park?</td>
</tr>
<tr>
<td>Q9</td>
<td>How did you feel when using the interface?</td>
</tr>
<tr>
<td>Q10</td>
<td>It was easy to find what I wanted in the code using Code Park.</td>
</tr>
</tbody>
</table>

- Except Q3, Q6 and Q9 all the other answers are beyond the average scale
- The results are similar to previous study
- Easy to learn and use Code Park (Q1 & Q2)
- Easy to work on the project (Q7)
- All features integrated well in Code Park (Q4)
Organizing Existing Project

- The project contains 33 classes
- Divided participants into two groups
Group 1

Given project that are already organized into directories
Participant 1: “Folders were arranged spatially in groups. Classes that appeared related by name were sub-grouped.”
Participant 2: “[I kept] directories grouped together.”
Participant 3: “I just arranged classes of a particular folder in each row.”
Participant 4: “The classes were arranged alphabetically for each folder and I arranged the classes in the same folder in the same line.”
Participant 5: “I tried to group the related class together based on the usefulness and field.”
Group 2

Given project that are not organized in any particular manner
Participant 6: “When arranging the classes my first concern was to group similar classes together.”
Participant 7: “I tried to place the rooms in the chunk of similar classes. My priority was to place them in such a way that they are easy to find again.”
Participant 8: “I grouped the rooms based on their classes' name.”
Participant 9: “Big models together. Smaller ones in the middle so I can find them easier.”
Discussion

• There is a possible relation between the user's cognitive understanding of the codebase and their decisions in organizing building block

• The users mostly chose to organize the constituent parts of the project based on their relationship with respect to each other
CommunityClick
Problem

- Notetaking is **distracting** during a meeting
- Collecting all the ideas in a meeting is **hard**
Demo
User Study

<table>
<thead>
<tr>
<th>Study</th>
<th>Objective</th>
<th>Method</th>
<th>Participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Needfinding</td>
<td>Understanding domain problem</td>
<td>Field observation and Survey</td>
<td>20 organizers</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>66 community members</td>
</tr>
<tr>
<td>Study 1 - Controlled Lab Experiment</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phase 1: Meeting Simulation</td>
<td>Simulating meeting and utilizing iClicker</td>
<td>Observation and Survey</td>
<td>8 participants</td>
</tr>
<tr>
<td>Phase 2: Report Creation</td>
<td>Comparing CC with G-Doc</td>
<td>Within-subject comparison</td>
<td>20 participants</td>
</tr>
<tr>
<td>Phase 3: Report Evaluation</td>
<td>Evaluating reports</td>
<td>Blind review</td>
<td>5 participants</td>
</tr>
<tr>
<td>Study 2 - Expert Feedback</td>
<td>Collecting overall feedback and adoption possibility</td>
<td>Demo and Phone interview</td>
<td>6 civic leaders</td>
</tr>
<tr>
<td>In-Class Deployment</td>
<td>Exploring real-world usage</td>
<td>Field observation and Survey</td>
<td>18 students</td>
</tr>
</tbody>
</table>
Other cool Projects
Math Boxes 2015
Electrick 2017
LumiWatch 2018

LumiWatch
On-Arm Projected Graphics and Touch Input

Robert Xiao
Teng Cao
Ning Guo
Jun Zhuo
Yang Zhang
Chris Harrison

Carnegie Mellon University
Thank you