UbiCom TicTacToe: Blog

**Blog 1 Tic Tac Toe for Arduino:**

**Hi and welcome to our blog!**

In this blog we want to show our progress in our mini project as a part of ubiquitous computing.

The team consists of Julian Altmeyer and Lukas Luschin. We are two students of Applied Computer Science at HTWG in Konstanz Germany in the 5th semester.

**What ist the project about?**

As the title mostly reveals, our project is about implementing the well known game TicTacToe on an arduiono uno microcontroller board. The game as itself will be controlled with a small display on top, which allows users input over touch. Furthermore there will be the option to select between playing round based versus another human (or other creatures which are able to tab on a screen) or a single player mode, where your opponend will be emulated by the arduino.

**So how to start?**

We decided to devide our project into several parts which we will blog here separatly. First of all we have to create a reliable library for the display which offers us all the needed functionality to create a userfriendly and good-looking graphical user interface. Next Step is to create exactly this UI. What we want to define is which possibilities the user has and how to establish that. If this is done we can start with creating an architecture for our game and code it. At last we implement the single players ‘AI‘, because this will be a less important feature.

**Blog 2 Connecting to the Display:**

greetings!

After a lot of formality we finaly want to start with our project.

As meantioned we start with the connection to the display. We use a 2.8“ TFT Display with touchscreen. Here we found a great tutorial and also a software library with lot of functionality and demonstration projects from adafruit.

Tutorial:

<https://learn.adafruit.com/adafruit-2-8-tft-touch-shield-v2/overview>

Library:

https://github.com/adafruit/Adafruit\_ILI9341

With this knowledge and features we created a for the project more specific software library with functionality like UI-elements for the menu and the game.

Therefore we try to consider every need of our game.

**Creating a screen**

So we abstract the idea of an screen based architecture, where you can add several elements to. These elements are defined by several classes with different content. All those element classes have the same base-class which includes an abstract „drawing“ and an „onClick“ function.

Furthermore we need an instance which manages this Screens to avoid unneccessary redrawing of a screen.

The final order of functions calling is that our main instance calls a rendering function in the screen manager, which decides if the current screen needs to be rendered. If yes, the render function of the actual screen instance is called. It includes the creation of the screen and the filling with the defined elements.

**Touch events**

The Screen Manager also takes responsibility for touch events. It gets the last touched position on the screen and hands it over to the actual screen instance. It checks if there is an element at this position and calls the corresponding „onClick“ function where appropriate.

So the final architecture looks like this:

* UML hier einfügen!

**Blog 3:**

Welcome back

After creating a framework for our graphical user interface it’s time for another design phase.

**Blog 4:**

Gamelogic

**Blog 5:**

KI