

Meta-Lab: programação de um laboratório interativo

Metal-Lab: the programming of an interactive lab

Sandro Canavezzi de Abreu

Universidade Federal de Minas Gerais, Brasil.
sandroid@ufmg.br

Guilherme Nunes de Vasconcelos

Universidade Federal de Minas Gerais, Brasil.
guilherme@guiv.com.br

Mateus van Stralen

Universidade Federal de Minas Gerais, Brasil.
mateus.stralen@gmail.com

Abstract

Here we discuss the technological and theoretical issues that conform the restructuring proposal of the Computer Laboratory of Escola de Arquitetura e Urbanismo da UFMG, reconfiguring it in what we call "Meta-Lab": a space composed of programmable modules that make up the so called Sistema Hidra(!), a system structured in three levels (sensory, processor and actuator level) which receives environmental information via sensors, processes these information and changes the environment using actuators. We will address in more detail the processing level, a fundamental layer for enabling the implementation of "interactive permanency" through continuous reprogramming of interactions in Meta-Lab.

Keywords: Interactivity; Combinatory; Interactive Architecture;

Introdução

Nos interessa aqui a problematização da ideia de permanência da interatividade em ambientes transitórios (entendendo interatividade como a relação do pré-programado com o indeterminável e espaços transitórios como espaços cujas funções, usos, significados mudam no tempo) através da contínua reprogramação de seus aspectos computacionais. Portanto, procuramos refletir sobre a capacidade desse espaço de permanentemente assimilar alterações e redesenhos de sua interatividade. Essa questão tem como contexto a busca pela abertura dos espaços arquitetônicos para novos usos e novas apropriações por parte de seus usuários. Assim, a permanência da interatividade se refere a possibilidade de estruturar um sistema interativo aberto à mudanças que, por sua vez, devem ser mobilizadas pelo usuário. No caso aqui abordado (a programação do Meta-Lab), descreveremos e analisaremos como o Sistema Hidra(!) (a dimensão computacional desse laboratório) é estruturado para receber alterações que viabilizem e potencializem uma utilização aberta desse espaço. Discutiremos também como a natureza aberta e distribuída do Sistema Hidra(!) possibilita uma reprogramação coletiva e colaborativa desse espaço a partir da recombinação contínua entre os módulos e elementos espaciais/ambientais que compõem o Meta-Lab. Os usuários seriam, assim, geradores de novas opções, deixando de ser operadores de espaços programados para ser programadores aptos a reinventar sinergicamente as regras deste espaço, reprogramando certos aspectos físicos e "informacionais" desse espaço de modo colaborativo.

Sobre a abertura e permanência da interatividade

Tomada como uma interface, que de forma ambivalente obstaculiza ao pré-determinar em algum grau a interação (Flusser, 2008) e potencializa novos usos pois inaugura possibilidades não alcançáveis antes dessa mediação (Zielinski, 1997), a arquitetura interativa pode mediar desde as mais simples às mais complexas relações entre os usuários, o espaço e o ambiente. Como exemplo do primeiro caso, pensemos nas situações em que o usuário e as variações ambientais são rastreados e transformados /codificados em informações de intensidades de luz, som, calor, umidade etc., que, por sua vez, quando relacionados e processados no sistema computacional, conformam o que chamamos aqui nesse artigo de "estados". Esses estados são utilizados para reconfigurar o espaço e o ambiente, alterando seus comportamentos e, conseqüentemente, modificando as possibilidades de interação entre os usuários e esse espaço. Essa reconfiguração pode ser automática ou operada pelo usuário a partir de opções pré-estabelecidas. Já em uma relação mais complexa, como a busca no Meta-Lab, a arquitetura interativa daria ao usuário a possibilidade de reprogramar a dimensão programável da interatividade deste espaço, isto é, seus estados. Essa reprogramação pode, por sua vez, ter graus de abertura quanto a interação: desde uma interação a partir de opções pré-estabelecidas (tornando o usuário um mero operador, selecionador) até uma interação que potencialize a geração de escolhas, isto é, uma apropriação onde o usuário pode reinventar continuamente o espaço e sua relação com ele.

Nesse espaço, onde a interatividade é continuamente reprogramada, o próprio sistema computacional é redesenhado para assimilar novas escolhas ao traduzi-las em novas opções disponíveis a novas interações. Assim, instala-se um processo de retroalimentação: de um lado,

novas demandas de uso geram novas reprogramações; por outro lado, uma nova reprogramação pode ativar usos e apropriações latentes, que não se manifestariam antes dessa nova demanda. Tal relação envolve, portanto, dois universos distintos: o universo do pré-determinado (o território da programação do computador) e o universo do indeterminável (dos eventos acidentais, imponderáveis). Quando a reconfiguração é autônoma, sistemas baseados em Inteligência Artificial (I.A.) e Algoritmos Genéticos (A.G.) são implementados e tornam-se capazes de tomar decisões complexas. Todavia, a partir de um entendimento construtivista da tecnologia (Feenberg, 1991), optamos pela ênfase na participação do usuário como *designer* e reprogramador das reconfigurações físicas possíveis do espaço e de sua interação com ele. Vale dizer que isto não exclui a incorporação de I.A. e A.G. em sistemas abertos à reprogramação, apenas transfere maior poder de interferência e de tomada de decisões para os usuários sobre o comportamento/estados do sistema. No construtivismo tecnológico de Feenberg (1991), o usuário deve ser ator e agente transformador dos objetos-técnicos que o cercam.

No caso dessa pesquisa, essa atuação acontece em uma micro-escala: o usuário é um ator capaz de redesenhar o “comportamento” dos objetos-técnicos que compõem um espaço interativo e, conseqüentemente, ativar novas possibilidades para o seu comportamento. Espaços permanentemente interativos não são novidades em Arquitetura, tampouco a evolução dos modos de interação nestes espaços através de programas que se reorganizam autonomamente no tempo, no entanto, a permanência de uma interatividade cuja programação possa ser refeita continuamente pelos usuários, precisa ainda ser melhor investigada. É, pois, neste sentido, que este artigo pretende contribuir ao propor estratégias que explorem a questão, ampliando a reflexão sobre o papel ativo dos usuários na organização de espaços e ambientes interativos.

Justificativa

De modo geral, a temática deste projeto de reprogramação de um laboratório interativo insere-se nas discussões sobre o uso da tecnologia computacional no âmbito da Arquitetura em um contexto de colaboração coletiva. Especificamente, apontaremos estratégias iniciais que potencializem a transitoriedade de um espaço (suas variações de uso, de significado, de performance) a partir da possibilidade de um redesenho/reprogramação contínua dos modos de interação com esse espaço. Portanto, o design de um espaço dessa natureza, no caso o próprio Meta-Lab, passa pela coordenação de seus níveis programáveis e pré-determinados (ainda que este possua a possibilidade de se auto-organizar, de se adaptar automaticamente, até mesmo “aprendendo” com o comportamento do usuário) com a indeterminação de eventos futuros nascidos da tensão entre o pré-estabelecido e o acidental, imponderável, imprevisível enfim, dimensões virtualizantes (Lévy, 1996) e criativas do real. Indicamos, dessa maneira, que a idéia de permanência da interatividade de um espaço transitório poderia passar pelo (re)desenho dessa interatividade e, em termos técnicos, pelo problema de como programar algo para ser reprogramável. Nesse sentido, apresentaremos o Sistema

Hidra(!), ainda em desenvolvimento, como estratégia inicial de abertura da camada computacional do Meta-Lab. Detalharemos a seguir os componentes desse sistema e posteriormente analisaremos como o conceito de “combinatória” foi fundamental para a estruturação desse sistema, caracterizando o que chamamos de “motor recombinante”.

Sistema Hidra

O Sistema Hidra(!) (o “!” aqui é uma notação matemática para “fatorial”, recurso utilizado em análise combinatória) promove a coordenação e a interação dos aspectos computacionais do ambiente e seus usuários. É importante ressaltar que assumimos o usuário/reprogramador como parte indissociável desse sistema interativo (Pask, 1975). Para que essa interação se efetive, o Sistema Hidra(!) é dividido em três níveis: nível sensório, nível processador e nível atuador.

O **nível sensório** é formado por objetos técnicos que incorporam diversas formas de sensores de luz, de movimento, de temperatura, de distância, de umidade, de distância, etc. Esses objetos técnicos, aliados a um hardware específico (controladores como Arduinos, Raspberry Pi, LittleBits e similares), convertem sinais analógicos em digitais. É o nível, portanto, onde se realiza a digitalização de sinais analógicos.

O **nível processador** é a parte que contém a programação, e que é reprogramável. Constitui-se do conjunto de algoritmos que processa as informações digitalizadas de diversas formas, reconhecendo padrões e/ou tomando decisões (via operações booleanas simples ou complexas como I.A. e A.G., por exemplo).

O **nível atuador** é formado por objetos técnicos que incorporam diversas formas de atuadores. Ele “atua” diretamente no ambiente através de interferências mecânicas (alterações controladas dos movimentos de elementos físicos/materiais desse espaço: abertura/fechamento de portas, janelas, brises, persianas e demais objetos-técnicos e interfaces móveis programáveis) e demais interferências físicas/tangíveis: síntese e manipulação sonora, controle luminoso, controle de temperatura e umidade, síntese e manipulação de imagens, etc. Esse nível é composto por um conjunto de atuadores (LEDs, buzzers, caixas de som, projetores, óculos de Realidade Virtual, lâmpadas, relays, servo-motores, motores de passo etc.).

Os níveis são programados utilizando a plataforma MAX7 e a programação foi feita de modo a propiciar a recombinação e o mapeamento dinâmico entre sensores e atuadores, seja localmente ou remotamente (via internet), possibilitando, a título de exemplo, alterar o comportamento de um atuador — como um LED, por ex. — a partir de um sensor de luz situado no mesmo local que o LED ou de um lugar muito distante, no caso, acionado remotamente por um controlador. A utilização dessa plataforma se deu por esta utilizar uma linguagem estruturada em *dataflow*: os elementos da programação (as funções, variáveis, objetos,

etc) são organizados graficamente e podem ser dinamicamente dispostos e relacionados entre si.

A camada de processamento do Sistema Hidra(!), como qualquer software programado na plataforma MAX7, possui dois estados principais: estado programação e estado operação. No primeiro, o usuário (ou designer de interface) pode reprogramar o software (que chamaremos de *patch* daqui em diante). Ele pode adicionar ou retirar funções e reorganizar o fluxo de dados. No modo operação o usuário pode operar o *patch*, controlando-o, alterando variáveis, alimentando-o com dados que modificam o comportamento do *patch*. Temos assim, dois níveis de interação possíveis: o da programação e da operação. Ambos podem ser complementares (quando se está programando um *patch*) ou, no caso do modo operação (quando se está controlando o *patch*), pode existir independentemente. É nessa camada que se encontra a possibilidade de recombinar inputs e outputs, isto é: abre-se a possibilidade de canalizar dinamicamente qualquer entrada (input) em qualquer saída (output) através de uma interface que torna visível e manipulável essas possibilidades de canalização.

Para apresentar o Sistema Hidra(!), começaremos ilustrando os componentes principais do software, esclarecendo os passos iniciais para se configurar o sistema.

O software do Sistema Hidra(!) é composto por dois módulos principais: O primeiro, que recebe as informações provenientes dos sensores e as enviam pela internet aos outros terminais (sejam eles locais ou remotos). E o segundo, que recebe essas informações e possibilita a recombinação dinâmica dessas, disponibilizando-as para que possam ser utilizadas como parâmetros para atuadores ou outros programas que possam receber esses parâmetros (Grasshopper, Unity 3D, Unreal4, por exemplo). Cada terminal da rede deve possuir os dois módulos, pois são complementares e interdependentes.

No primeiro módulo, configuramos a entrada de informações provenientes de sensores (nível sensorial), indexando às variações numéricas termos que auxiliem na identificação da natureza da informação (se é sonora, luminosa, gasosa, distância, calor, etc.). Essa informação é “empacotada” junto a demais informações relevantes (qual o número do computador de onde parte essa informação, qual a porta que deverá ser ouvida pelos outros terminais e se o sensor está operante ou não, (Figura 1) e enviada para os computadores que fazem parte da rede local ou remota (Figura 2).

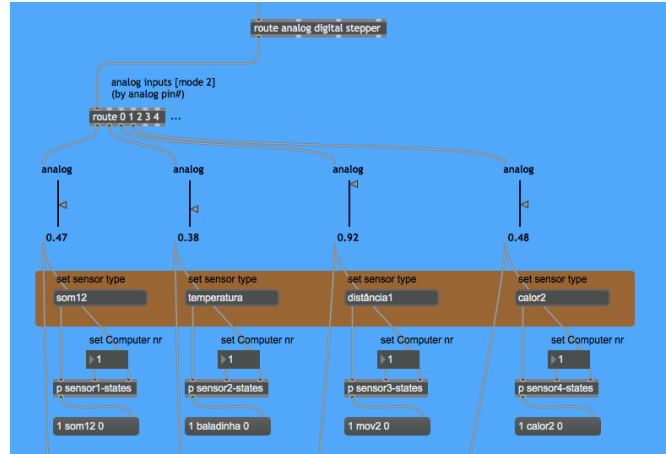


Figura 1: Configuração das entradas de informação. Fonte: Autor, 2016.

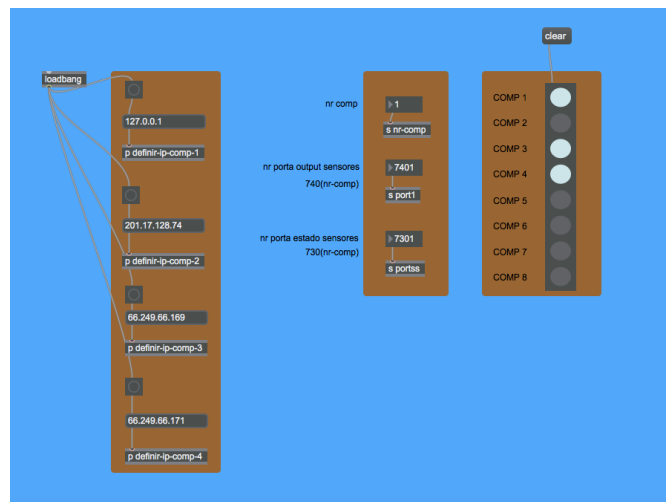


Figura 2: Configuração dos endereços IPs dos computadores e definição de quais computadores participam da rede. Fonte: Autor, 2016.

No segundo módulo, que recebe as informações da rede, é possível estabelecer de qual o computador iremos receber as informações. A informação que chega é “desempacotada” e seus elementos (dados relevantes sobre as origens de cada variação numérica) são tornados visíveis (Figura 3) e manipuláveis através de uma matriz que torna possível associar entradas de informação a diversas saídas. Esse módulo é a interface entre a camada combinatória do Sistema Hidra(!) e o nível atuador.

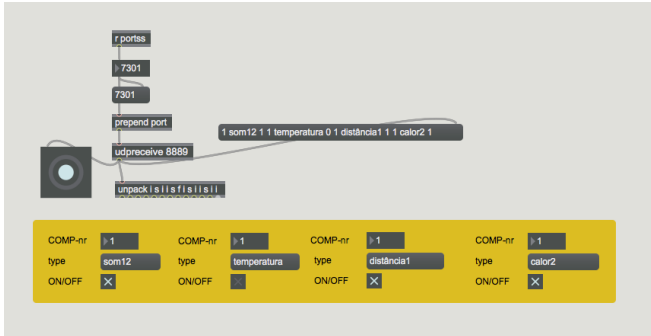


Figura 3: Configuração da porta, com informações provenientes dos sensores de outro computador da rede. Fonte: Autor, 2016.

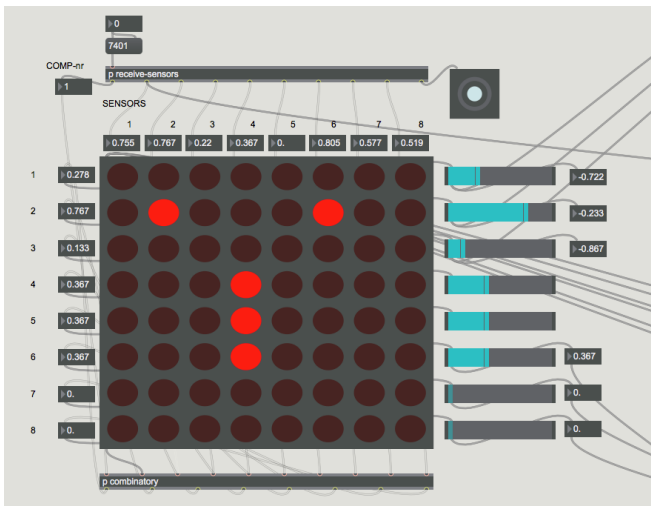


Figura 4: Matriz combinatória, onde é possível indexar uma entrada (parte superior) a qualquer saída (lado esquerdo). Fonte: Autor, 2016.

A incorporação dos princípios da combinatória se tornam mais visíveis nesse segundo módulo: é disponibilizada na interface um painel (Figura 4) onde é possível recombinar, de maneira dinâmica e intuitiva, os diversos *inputs* vindos de sensores (e demais entradas) e os diversos *outputs* (atuadores, imagens de vídeo, modelos 3D parametrizáveis, etc).

Nesse contexto, o Sistema Hidra(!) se propõe como uma máquina recombinante e parametrizante, que gera híbridos entre o digital e o analógico, potencializando a geração de híbridos entre elementos arquitetônicos e elementos da computação física. Para entendermos o Sistema Hidra(!) como tal, precisamos nos aprofundar nos processos de digitalização e processamento da informação como parte de uma lógica combinatória. Como veremos a seguir, o conceito de combinatória faz parte da natureza dos processos computacionais. Torná-la mais visível e parte da dinâmica de reprogramação da interatividade foi uma estratégia fundamental para o desenho desse sistema. No entanto, ela não deve ser alienadamente manipulada, sendo necessária a compreensão e percepção de sua onipresença nos processos digitais para que o princípio combinatório seja efetivamente utilizado como motor de geração de novas possibilidades. Desse modo a percepção da dimensão

combinatória pode construir o distanciamento necessário para o redesenho da interação enquanto redesenho de fluxos de informação.

O Sistema Hidra(!) e a Combinatória

Independentemente da velocidade de processamento de um sistema digital, do fato de o sistema ser binário ou quântico, ou do nível de complexidade dos cálculos e algoritmos, sempre haverá a relação de um sistema discreto com um contínuo/físico analógico, e essa relação sempre se dará tendo como base a combinatória. Assim posto, da digitalização interessa o que poderíamos chamar de “dimensão arbitrária” e as formas da sua “aparição”, isto é, o mecanismo que efetiva a conversão de entidades analógicas (físicas) em entidades digitais (numéricas/binárias/elétricas) e vice-versa: a combinatória.

A seguir mostraremos de que maneira a combinatória é utilizada como artifício tradutor, atentando-se para sua ambivalência: ela é **reduzora** e, ao mesmo tempo, um **motor hibridizante** que, em último caso, poderia ser considerado como aliado de um processo criativo. Podemos considerá-la reduzora, pois para a sua existência e efetivação, ela faz parte de um conjunto de processos que traduzem, como já dito, o universo analógico/físico em números. Esse processo transforma entidades contínuas (ondas, por exemplo) em “partículas” numéricas (números binários). Essa atividade discretiza e fragmenta coerentemente o universo físico. Ressalta-se aqui a ideia de coerência: a fragmentação tem que ser organizada de forma a ser assimilada pelo computador. A essa coerência chamamos de “coerência computacional” e dela trataremos mais adiante.

A tradução “em números” não vem sem um custo, pois sempre se perde informação ao se digitalizar algo analógico, uma vez que o mundo físico é convertido em números a partir de taxas de amostragem: recortes no tempo e no espaço que convertem o infinito entre dois pontos em quantias mensuráveis, fazendo que intensidades que variam continuamente sejam fragmentadas em degraus abruptos. A quantidade de recortes que se promove no objeto analógico é o que chama-se de resolução de um sistema.

Esses degraus, medições relativas às entidades analógicas, precisam sofrer uma conversão a mais: precisam se transformar em bits. Essa transformação, que foi vislumbrada e modelada por Claude Shannon (1991) em sua Teoria da Informação (Thomas, 1991), deve organizar esses degraus em linguagem binária (“zeros e uns”). Cada fragmento medido terá disponível um certo número pré-definido de zeros e uns para representá-lo. A possibilidade de se converter uma informação em zero ou um é o que se conhece por “bit”. Quanto maior for o número de bits disponíveis para essa representação, maior também será a resolução desse sistema. No caso de uma imagem, por exemplo, temos a resolução espacial (fragmentos da superfície da imagem) e uma resolução de cor e luminosidade (quantos *bits* cada fragmento da superfície tem para representar uma cor e seu brilho).

A determinação de quais sequências de zeros e uns são utilizadas para representar algo não obedece a qualquer relação causal ou indicial com o que representa: é puramente simbólica, por contiguidade (a informação não possuiria vínculo necessário com um conteúdo ou com um corpo/meio). Portanto, essa arbitrariedade não contém nada que nos permita identificar em uma sequência binária se ela representa, por exemplo, uma cor ou um som: uma mesma sequência numérica pode ser tratada como som ou como imagem pelo sistema. E aí reside algo sem precedentes na geração de entidades analógicas: podemos facilmente “ler” uma sequência binária que foi gerada a partir da conversão de um fragmento sonoro como sendo uma imagem, traduzindo essa sequência em variações de cores em uma superfície sendo válido também o contrário, podendo uma imagem ser lida como som.

Embora a conversão de som em imagem (e vice-versa) possa ser feita analogicamente, é com a digitalização que essa possibilidade se radicaliza: a recombinação das sequências binárias pode sintetizar entidades analógicas que não existiriam de outra maneira. A informação poderia, portanto, assumir qualquer corpo ao ser trazida ao universo analógico. Com isso, podemos mapear um domínio no outro, através de uma tradução que converte números “extraídos” de uma dimensão (sonora, visual, etc) em outra dimensão de natureza diferente. Esse mapeamento pode ser ainda mais elaborado: variações de intensidades podem ser mensuradas (a partir da digitalização) e posteriormente serem organizadas como parâmetros que alteram o comportamento de entidades de outra natureza.

Isso posto, podemos dizer que o mapeamento e a parametrização, estruturados por processos combinatórios, caracterizam a essência do Sistema Hidra(!). No entanto, cabe esclarecer que esses processos só são possíveis pois existe antes o que chamamos de descorporificação da informação, isto é, a autonomia da informação quanto a matéria ou suporte. Essa autonomia propicia a própria existência do conceito de entrada e saída de dados em processos e computadores digitais. Isso só é possível se tomarmos o sentido de informação nos moldes de Shannon (1991). Nele, a informação não possui e não depende de um corpo, ela apenas transita por corpos e qualquer ação desses corpos nessa transmissão deve ser prevista para se compor a informação. Se não for prevista, deve ser ao menos mensurável. Essa ação indesejada, mas mensurável, é o ruído. Ainda: a informação pode se utilizar de energia luminosa ou elétrica ou mecânica para se propagar, mas a informação em si não se altera. Uma mesma informação pode trafegar por todos esses meios e sua integridade deverá ser assegurada (Papert, 1965).

Essa autonomização da informação é um conceito chave para entendermos os processos de digitalização e demais processos combinatórios digitais (que chamaremos de computadores daqui em diante). Os *inputs* e *outputs* de um sistema digital (um computador, por exemplo) não são energia (luminosa, mecânica, elétrica) e sim informação. O *input* não é som, luz, sinal elétrico, mas sua transformação em informação. O *input* de um microfone, por exemplo, não é a onda sonora (ou um conjunto delas), mas a onda sonora tornada movimento mecânico que, por sua vez, é convertido

em sinal elétrico. O que permanece nessas traduções é a informação. Assim, em concordância com a Teoria da Informação, a mesma informação ora é som, ora é movimento, ora é eletricidade.

Algo semelhante acontece no nível sensorial do Sistema Hidra(!): intensidades de luz, som, calor, etc., são convertidos em sinais elétricos que por sua vez são traduzidos em informação quantizável, isto é, bits de informação.

Experimentos

Os experimentos relativos ao Sistema Hidra(!) se dividem em duas fases: a primeira, com ênfase técnica, onde são testadas diversas configurações dos elementos externos ao sistema — sensores, Arduinos, atuadores, microfones, óculos de realidade virtual etc. — e definidos os protocolos de comunicação entre o Sistema Hidra(!) e demais softwares, no caso, Unity3D, Unreal4 e Grasshopper (optou-se pelo protocolo de comunicação UDP; este protocolo já é bem disseminado como artifício para promover a comunicação entre softwares), bem como seus elementos internos, como a definição de protocolo de comunicação entre terminais com Sistema Hidra(!), definição da comunicação entre Arduinos e o Sistema Hidra(!) (optou-se pelo Maxuino, <http://www.maxuino.org>, pela facilidade de utilização devido a maior amigabilidade da interface), rotinas de calibragem automática de sensores e layout de uma interface amigável.

Na segunda fase está prevista uma série de atividades ocupando três salas distantes entre si, onde serão exploradas dinâmicas de interação entre elas. Nessa fase a ênfase é na sensibilização dos alunos quanto aos aspectos combinatórios da interface e de como devemos “desenhar” os *inputs* enquanto informação a ser digitalizada para posterior computador. No momento estamos finalizando a primeira fase com alguns aspectos da segunda fase já encaminhados.

Por se tratar do design de aspectos técnicos do sistema, na primeira fase foi fundamental a consolidação dos conceitos aqui abordados por parte dos pesquisadores e alunos envolvidos: combinatória, digitalização, informação. Sem essa consolidação as estratégias relativas à organização do fluxo de informação, suas possibilidades combinatórias e otimização da distribuição da informação não teriam sido realizadas com clareza. E, principalmente, não teríamos alcançado um sistema com alto grau de abertura para sua re-programação (e recombinação) por parte dos usuários. Podemos dizer, assim, que o Sistema Hidra(!) é uma expressão, no formato de uma rede de aplicativos complementares e interconectados, dos conceitos aqui abordados.

Na segunda fase, iniciamos um trabalho de tensionamento desse sistema na direção de uma maior abertura para a interação. Nela buscaremos redesenhar a interface a partir de estudos de casos onde exploraremos diversas configurações de entrada de dados no sistema. Testaremos, portanto, a robustez e a escalabilidade do sistema, isto é, o quanto ele é ampliável sem perder sua coerência.

A interação com os ambientes interativos, assim, não se restringem em explorar configurações pré-estabelecidas e sim buscar reconfigurá-las buscando novas combinações e orquestrações entre *inputs* e *outputs*. Buscaremos explorar o caráter combinatório do Sistema Hidra(!) enquanto motor hibridizante através do redesenho do sistema a partir das demandas da interação, sendo que esse redesenho deve acontecer com a participação do usuário/designer.

A seguir traremos um breve relato dos experimentos previstos na segunda fase.

Experimentos Previstos

Sensores (a princípio, luz, temperatura, distância, microfones, câmeras e joysticks) e atuadores (a princípio, caixas de som, LEDs, fitas LEDs, óculos de realidade virtual HTC Vive e projetores) serão distribuídos em três ambientes situados em três salas distintas e fisicamente distantes umas das outras, a saber: sala do Laboratório de Computação Radamés, sala principal do Laboratório LAGEAR e sala do Laboratório de Informática PCs.

Na primeira sala, Laboratório Radamés, projeta-se a imagem do ambiente virtual. Sensores de luz e calor distribuídos pela sala informam os vários estados do ambiente quanto a sua iluminação e temperatura.

Na segunda sala, LAGEAR, o HTC Vive também exibe o mesmo ambiente virtual. Cada terminal (projetor ou HTC Vive) tem a imagem atualizada a partir da perspectiva e controle de cada usuário). Essa sala também conta com sensores de luz e microfones. Esses microfones, associados a *patches* programados em MAX7, medem a variação de amplitude e frequência dos sons produzidos nessa sala.

Na terceira sala, Laboratório PC, a ênfase é na produção de paisagens sonoras distribuídas em várias caixas de som (dez no total). Essas caixas formam um *grid* de pontos de saída de áudio distribuídos por toda a sala. Em cada ponto também existe um sensor de distância que mede a proximidade do usuário em relação à caixa de som.

O exercício proposto consiste em disponibilizar todos os *inputs* provenientes dos sensores para toda a rede. Assim, parâmetros de variação sonora — no caso da terceira sala — e variações geométricas, de iluminação, cor e tamanho dos objetos virtuais (figuras 5, 6 e 7) são remotamente gerados e localmente indexados aos *outputs* disponíveis. Assim, em um dado momento, a variação de luz da primeira sala pode parametrizar a variação de cor do ambiente virtual acessado na segunda sala. Em um segundo momento, essa mesma variação de cor pode ser controlada pela variação de distância de um usuário em relação a uma das caixas de som da terceira sala.

Desta fase, já realizamos a comunicação entre os patches do Sistema Hidra(!) e um ambiente de Realidade Virtual programado no MAX7, sendo prevista agora a utilização do software Unreal4 para a geração de um ambiente mais elaborado e robusto.

Nesse experimento exploramos a variação dinâmica de *inputs* provenientes do ambiente (luz, som, movimento, etc.), indexando-os à parâmetros de uma superfície. Assim, transformamos a geometria (os vértices da geometria podiam ser manipuláveis), a escala, as cores e a iluminação do objeto virtual a partir desses parâmetros dinamicamente indexados via Sistema Hidra(!). Assim, quando o usuário se movimentava pela sala, quando falava ou quando manipulava objetos luminosos, qualidades do objeto virtual se alteravam. Logo, o usuário podia observar, manipular ou visitar o objeto (a escala do objeto variava desde a escala de um objeto até a escala de uma edificação de três andares) (Figuras 5 a 7).

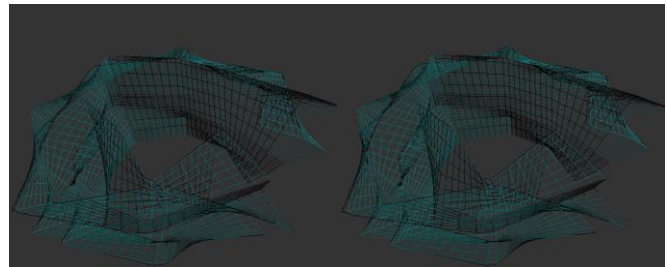


Figura 5 – Alteração da forma e das cores do objeto virtual a partir de *inputs* recombinaíveis via Sistema Hidra(!) – visão estereó gerada para imersão no ambiente virtual. Fonte: Autor, 2016.

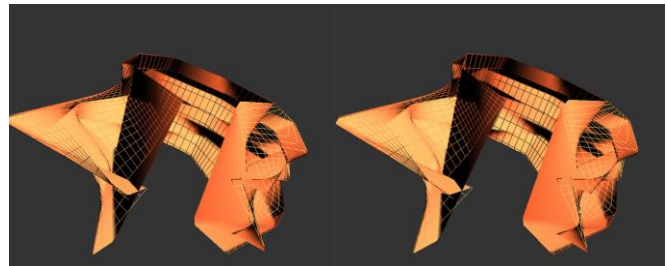


Figura 6 – Alteração da forma e das cores do objeto virtual a partir de *inputs* recombinaíveis via Sistema Hidra(!) – visão estereó gerada para imersão no ambiente virtual. Fonte: Autor, 2016.

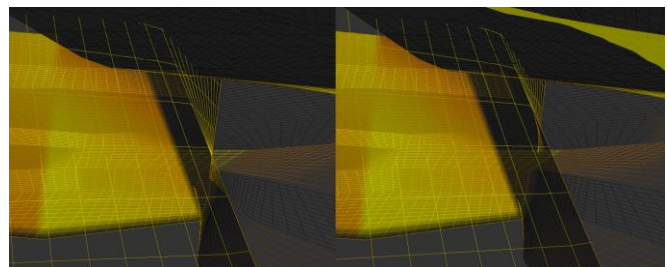


Figura 7 – Interator visitando internamente o objeto virtual – visão estereó gerada para imersão no ambiente virtual. Fonte: Autor, 2016.

Conclusão

Muitos encaminhamentos possíveis poderiam derivar dos experimentos acima descritos. Entendemos que esses

encaminhamentos devem tentar responder às seguintes questões: 1) De que maneira a abertura do sistema deve promover e engajar o usuário em sua relação com o espaço físico? Quais as possíveis formas desta interação? 2) Como se dá uma interferência direta no comportamento do Meta-Lab por meio de sua reprogramação? No desenho e sucessivos redesenhos do Sistema Hidra(!) o que está em jogo, portanto, é a fricção entre o mundo determinístico da programação desse sistema e a possibilidade desse sistema (ao organizar e tornar tangível combinações possíveis entre informações captadas e processadas) de proporcionar apropriações novas, não previstas. Essa abertura para o novo só é possível na contínua tensão entre possibilidades latentes pré-determinadas e um uso/programação que explora intensamente a geração dessas combinações entre informações captadas. Nesse sentido, em essência, o Sistema Hidra(!) é um mero recurso determinístico que facilita a aparição de possibilidades latentes (Lévy, 1996), pois fornece ao usuário a facilidade de transformar eventos captados de um ambiente em informação manipulável e recombinação. Assim, mais importante que a atual configuração desse sistema são os princípios que norteiam seu desenho, tornando-o, a um só tempo, manifestação tangível de um conjunto de conceitos (combinatória, informação e digitalização) e ferramenta que possibilita a exploração desse conjunto ao explorar interações imprevisíveis. Dessa forma, o Sistema Hidra(!), em conjunto com o usuário/interator/programador, funciona como **catalizador de possibilidades latentes** (Simondon, 1989) que fazem sentido, que respondem a um problema ou ajudam a problematizá-lo.

Essa camada combinatória introduzida pelo Sistema Hidra(!) pode, portanto, potencializar a composição entre sistemas interativos com ênfase na permanência da interatividade ao disponibilizar a recombinação dinâmica entre seus elementos. Inputs provenientes de sensores podem ser endereçados a qualquer processo de computação vinculado ao Sistema Hidra(!). O resultado desses processos pode ser encaminhado a qualquer atuador dessa rede,

proporcionando o controle de seu comportamento. Dessa forma, estados do sistema composto por terminais contendo módulos do Sistema Hidra(!) conectados a sensores, atuadores e demais softwares podem ser continuamente reorganizados.

A partir dos experimentos realizados e propostos, pretendemos consolidar estratégias e práticas que se adequem a um maior número de casos, incorporando essas soluções ao Sistema Hidra(!). Daremos ênfase aos experimentos remotos, conectando terminais situados em diversos Laboratórios de Informática com os quais estabeleceremos parcerias. Nesse sentido, pretendemos organizar *kits* de equipamentos e softwares e promover workshops para a difusão dessas práticas e tecnologias.

Agradecimentos

Agradecemos à FAPEMIG pelo financiamento para participação do SIGraDi 2016.

Referências

- FEENBERG, A. (1991). *Critical Theory of Technology*. New York: Oxford University Press.
- FLUSSER, V. (2008). *Design: obstáculo para remoção de obstáculos?*. O mundo codificado, São Paulo: Cosac Naify, pp. 193–98.
- LÉVY, P. (1996). *O que é o virtual?*, São Paulo: Editora 34.
- PASK, G. (1975). *Conversation Theory*. London: Hutchinson.
- SIMONDON, G. (1989) *Du mode d'existence des objets techniques*. Paris: Aubier.
- THOMAS, JA. (1991). *Elements of Information Theory*. New York: Wiley-Interscience.
- ZIELINSKI, S. (1997). *Interfacing Realities*. Rotterdam: Uitgeverij De Baile and Idea Books.