

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE PSICOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM PSICOLOGIA SOCIAL E INSTITUCIONAL

OFICINAS DE CONSTRUÇÃO DE JOGOS DIGITAIS EM ETOYS: APRENDER EM
MOVIMENTO

PORTO ALEGRE, 2014

Bruno Fagundes Sperb

OFICINAS DE CONSTRUÇÃO DE JOGOS DIGITAIS EM ETOYS: APRENDER EM MOVIMENTO

Dissertação apresentada ao Programa de Pós-Graduação em Psicologia Social e Institucional da Universidade Federal do Rio Grande do Sul como parte dos requisitos para obtenção do grau de Mestre em Psicologia.

Orientadora: Prof^a Dr^a Cleci Maraschin

AGRADECIMENTOS

À eterna professora e orientadora Léa Fagundes, pelo exemplo de amor, entusiasmo e dedicação ao trabalho.

À avó Léa Fagundes pelo afeto, apoio incondicional, e presença iluminada.

À orientadora Cleci, pela paciência, entrega, solidariedade e competência na construção de um saber possível.

Aos colegas do grupo Oficinando em Rede pela acolhida e alegria no trabalho de pesquisa.

Aos colegas do Laboratório de Estudos Cognitivos, pelos caminhos percorridos desde 2002.

RESUMO

Este estudo, inserido no contexto das transformações e novas alfabetizações possibilitadas pela cultura digital, busca contribuir para a compreensão de processos de aprendizagem envolvidos na atividade de programação de jogos em Etoys (software de autoria multimídia desenvolvido em linguagem orientada a objetos). Foram realizadas quatorze Oficinas de construção de jogos em uma escola estadual de Porto Alegre com seis sujeitos de 5º ano do Ensino Fundamental cuja turma se inseria na modalidade um computador por aluno (1:1). As intervenções que sustentam o trabalho são guiadas pelo Método Clínico de Piaget. A análise dos processos de construção de jogos digitais pelos sujeitos é fundamentada em conceitos do campo do estudo da cognição, videogames (Game Studies) e da programação. O Etoys se mostrou um dispositivo interessante para a pesquisa desses processos. Através da observação e análise das produções desenvolvidas, pôde-se observar a dinâmica de estratégias, noções de programação e de jogos envolvidas na atividade de programar.

Palavras-chave: Etoys; Squeak Etoys; Aprendizagem; Videogames; Modalidade 1:1; cultura digital

ABSTRACT

This study, placed in context of the digital age's transformations and possibilities, aims to contribute to the comprehension of learning processes involved in programming games with Etoys (multimedia authoring software developed in object oriented language). Fourteen weekly workshops were given for a group of six students (which the class had one laptop per child) from a state public school in Porto Alegre. The interventions were carried out guided by Piaget Clinical Method. The processes of creating a game are grounded on the fields of Cognition, Game Studies and programming. The Etoys has proved to be an interesting tool for researching these processes. Through the analysis of the games produced, the dynamic of strategies, programming and game concepts could be observed.

Keywords: Etoys; Squeak Etoys; Learning; Videogames; One Laptop Per Child; Digital Age

SUMÁRIO

LISTAS DE FIGURAS E QUADROS	7
I. INTRODUÇÃO	8
II. REFERENCIAL TEÓRICO	12
2.1 Papert e o Construcionismo	12
2.2 Caracterização da Linguagem Orientada a Objetos – do LOGO ao Etoys	14
2.3 Caracterização do Etoys	19
2.4 Aprendizagem e programação de jogos digitais	27
2.5 Aprendizagem e Videogames.....	31
III. OBJETIVOS E PROBLEMA DE PESQUISA	38
3.1 Problema de Pesquisa	39
3.2 Objetivos	40
IV. MÉTODO.....	42
4.1 O Projeto UCA e o Projeto Província de São Pedro.....	42
4.2 A Escola, a turma e os participantes	43
4.3 O Projeto Piloto	45
4.4 As Oficinas	46
V. APRENDENDO A PROGRAMAR UM JOGO EM ETOYS.....	53
5.1 O Projeto Piloto	53
5.2 Aprendizagem e Programação de Jogos	57
5.2.1 Primeiros passos com o Etoys.....	57
5.2.2 Coordenação entre os objetos e seus efeitos (design, narrativa e programação)	63
5.2.3 TESTES como marcas da organização de regras e suas hierarquias	68
5.2.4 Monitoramento dos objetivos e subobjetivos	76
IV. CONCLUSÕES	81
REFERÊNCIAS.....	87

LISTAS DE FIGURAS E QUADROS

Figura 1: Integração de mídias no Etoys.....	20
Figura 2: Tela Inicial do Etoys.....	21
Figura 3: O “Mundo” do Etoys.....	21
Figura 4: Criação de um objeto no Etoys.....	22
Figura 5: Ações gráficas possíveis em um objeto no Etoys.....	23
Figura 6: Acesso aos comandos pelo visualizador do objeto.....	24
Figura 7: Criação de um Script com múltiplos comandos.....	25
Figura 8: Brincadeira sobre o plano cartesiano.....	48
Figura 9: Jogo “Onde está o Wally”.....	49
Figura 10: Uso de looping no Etoys no Projeto “Velozes e Furiosos” (Dan).....	59
Figura 11: Dificuldades na organização dos scripts.....	61
Figura 12: O jogo online “Bad Ice Cream”.....	64
Figura 13: O “jogo do sorvete”, (Liv).....	65
Figura 14: O “Jogo da formiga” (Liv).....	69
Figura 15: O “Jogo do Louva-a-Deus” (Joa G).....	69
Figura 16: O Jogo da Velha (Gab).....	70
Figura 17: O jogo “Aventuras da aranha” (Joa V).....	72
Figura 18: O jogo “As Aventuras de Joa” (Joa V).....	73
Figura 19: O jogo “Super Naves” (Joa G).....	74
Figura 20: TESTES compostos no jogo “Super Naves”.....	75
Figura 21: O jogo “Estrada Verde”.....	76
Figura 22: Sequência das versões do jogo “perseguição”.....	78
Quadro 1: Organização das atividades nas Oficinas do Projeto Piloto.....	46
Quadro 2: Organização das atividades nas Oficinas de Construção de Jogos.....	47
Quadro 3. Modos de Apropriação do Etoys a partir dos projetos desenvolvidos no Projeto Piloto.....	54
Quadro 4: Exemplos de produção por Modo de Apropriação.....	55

I. INTRODUÇÃO

As transformações possibilitadas pelo desenvolvimento tecnológico, sobretudo nas últimas décadas, têm operado mudanças nos mais diversos setores e instituições da sociedade. Podemos facilmente pensar em como algumas mudanças vêm modificando nossa vida cotidiana: ligamos eletrodomésticos ou dirigimos carros cada vez mais automatizados, usamos dispositivos móveis para nos comunicarmos com alguém em outro lugar do planeta, passamos um cartão magnético para obter dados online de nosso próprio computador.

Operamos ações outrora complexas ou mesmo impensáveis simplesmente apertando um botão. Se pensarmos, como Levy (1999), que a relação com as tecnologias muda nossas formas de ser, aprender e de viver, torna-se bastante complexo estabelecer os limites e as direções dessas transformações.

Mesmo em um país com tantas desigualdades como o Brasil, o acesso às tecnologias digitais vem se disseminando pelo conjunto da população. Conforme os últimos relatórios do Comitê Gestor da Internet, por exemplo, o telefone celular está presente em 88% dos domicílios brasileiros, superando inclusive os aparelhos de rádio. Em 2012, pela primeira vez, a proporção de usuários brasileiros que já usaram a internet (51%) superou aos que nunca usaram (CGI, 2012).

Recentemente, o neurocientista brasileiro Nicoletis¹ (2011) afirmou que em um futuro próximo as comunicações que hoje realizamos pela rede mundial de computadores, a internet, poderão ser feitas por pensamento, pela interface de nosso próprio corpo com máquinas mais sofisticadas.

Ao mesmo tempo em que podemos divisar novas implementações das tecnologias existem desafios à aprendizagem. Na medida em que máquinas e artefatos se tornam cada vez mais complexos tornamos mais difícil o conhecimento que temos sobre como funcionam. Em outras palavras, dependendo das interações que nos são ou não possibilitadas, podemos progressivamente nos tornar mais consumidores e menos produtores de conhecimento técnico sobre o mundo que nos rodeia.

Preocupados com os modos de apropriação que esse cenário acarreta e seus

¹. Baseado em seus experimentos de "ICMC" (interface cérebro-máquina-cérebro), Nicolélis prevê um futuro em que os seres humanos se comunicarão cerebralmente, ao que ele denominou "uma nova versão da internet, a 'brainet'" (NICOLELIS, 2011, p.25).

efeitos em termos de múltiplas mudanças e transformações culturais para o campo da psicologia e da educação, há autores que apontam para a necessidade de atentarmos para o que chamam de uma nova inteligência, a inteligência digital² (BATTRO e DENHAM, 2007; VEEN e VRAKKING, 2009).

As discussões em relação à transformação dos modos de aprender têm se constituído em um tema controverso na escola, que enfrenta desafios ao se apropriar dessas novas tecnologias. Parece haver consenso entre diversos pesquisadores em relação à ideia de que o uso das novas tecnologias na educação está ainda acoplado a métodos transmissivos e massificadores de ensino (FAGUNDES, MAÇADA e SATO, 1999; KAY, 2007; ALMEIDA e PRADO, 2011).

Resnick (2002) sustenta, como já o havia feito Papert (1994), que há a necessidade de se repensar tanto as abordagens de aprendizagem quanto a proposta de suporte das novas tecnologias para as transformações necessárias nas instituições educacionais.

O campo de pesquisa que possibilita e motiva este trabalho se encontra relacionado, tanto em âmbito nacional como estadual, em políticas públicas que visam incluir a escola nas transformações da cultura digital. Criado em 2006, o Programa UCA (um computador para cada aluno e professor) nasce como uma iniciativa do governo federal de investigar a possibilidade de adoção da modalidade 1:1 como forma de melhoria dos processos de aprendizagem na escola. Após as fases pré-piloto (entre 2007 e 2009) e piloto (2010 a 2012), o PROUCA, instituído na Lei Federal nº 12249, fomenta e avaliza centenas de iniciativas locais (estaduais e municipais) por todo país.

Atualmente, o Estado do Rio Grande do Sul adotou a iniciativa de inserção na modalidade 1:1 através do Projeto Província de São Pedro (PSP). Assim como o programa nacional, o PSP tem como objetivos principais a melhoria de nosso sistema educacional, pela “modernização tecnológica da rede estadual e a apropriação da tecnologia por alunos e professores, principalmente através da utilização de dispositivos móveis” (SEDUC-RS, 2012). A iniciativa do Estado do Rio Grande do Sul encontra paralelo no Uruguai, que desde 2008 iniciou o Projeto CEIBAL (Conectividad Educativa de Informática Básica para el Aprendizaje en Línea), através do qual distribuiu laptops para os alunos e professores de toda a rede pública de ensino fundamental, e parte do

². Frente aos novos hábitos e possibilidades de interação decorrentes da cultura digital, os autores ensaiam a identificação de uma inteligência digital, segundo os critérios da teoria das inteligências múltiplas de Howard Gardner. Battro e Denham (2007), em seu trabalho, atentam principalmente para a recente imersão das pessoas nas tecnologias digitais, e em como poderíamos identificar, nas capacidades cognitivas desenvolvidas nessas experiências, uma inteligência digital.

ensino médio daquele país.

Desde a origem da experiência 1:1 no Brasil, ainda em 2006, até o presente momento, em 2014, o LEC – Laboratório de Estudos Cognitivos – do Instituto de Psicologia da Universidade Federal do Rio Grande do Sul vem, no âmbito estadual, acompanhado e investigando, através de diversos projetos interdisciplinares de pesquisa e formação de professores, as experiências de inserção das escolas públicas na modalidade um computador por aluno. Como Laboratório de Pesquisa, o LEC tem o propósito de investigar processos cognitivos de estudantes que fazem uso da interação com as tecnologias digitais, contribuindo para a solução de dificuldades de alunos no processo de construção de conhecimento.

Mesmo antes de meu ingresso no curso de mestrado venho acompanhando os trabalhos desenvolvidos no LEC no trabalho de assessoria e acompanhamento das escolas. A experiência inicial, na modalidade 1:1 ocorreu em uma escola estadual, em Porto Alegre, onde foram feitas pesquisas sobre diferentes linhas de investigação, que se constituíram em algumas unidades de análise (FERRETI, 2007; KIST, 2008; SCHAFER, 2008; LIRA, 2008; LOPES, 2008; LIDNER 2009). No ano de 2011, uma nova doação de 500 laptops foi feita a outra escola estadual de Porto Alegre. Como mencionado, o LEC vem coordenando até o presente momento a execução do projeto, com a formação de professores e com o desenvolvimento de pesquisas sobre a apropriação das linguagens e ferramentas digitais.

Uma das unidades de análise do trabalho nas escolas é a atividade de programação, tendo como suporte o trabalho em um ambiente de autoria multimídia chamado Squeak Etoys (atualmente, apenas Etoys, um dos softwares presentes nos laptops). Desenvolvido por um grupo de pesquisadores do Media Lab liderados por Alan Kay (2011), do MIT, sobre grande influência das ideias de Seymour Papert (1985; 1994), o Etoys é um *software* livre, construído em linguagem de programação orientada a objetos, que permite ao usuário criar projetos multimídia de sua autoria.

Desde o início das atividades no Programa UCA, passei a trabalhar em Oficinas de Aprendizagem em Etoys com diferentes turmas dentro da escola. A proposta das Oficinas em Etoys, condizente com a missão do LEC/UFRGS e do UCA, é a de oportunizar aos sujeitos experiências de atividade sobre novas formas de se comunicar e interagir com o computador. As oficinas possibilitam trazer evidências de como se dá a apropriação dos alunos e professores de uma nova linguagem computacional e como essa interação facilita a observação de processos sócio-cognitivos-afetivos na atividade

de programação. Essas experiências permitiram reunir os primeiros resultados, apresentados mais adiante como piloto deste projeto e motivaram a sequência do estudo realizado neste trabalho.

Nossa proposta inicial de dissertação era realizar uma investigação psicogenética da noção de objeto digital e das relações entre esses objetos em espaço digital (micromundos), próprios da linguagem do Etoys, por meio da construção de jogos. Como comentaremos adiante, entretanto, essa proposta inicial sofreu deslocamento no andar do próprio projeto. Passamos então a focar os movimentos de aprendizagem realizados nas sessões de programação. O que detalharemos na apresentação do problema e objetivos da pesquisa. A pesquisa e intervenção são sustentadas em oficinas de produção dos jogos e intervenções guiadas pelo Método Clínico Piagetiano (Piaget, 1975).

Para circunscrever a investigação proposta, iniciamos com um breve percurso histórico do construcionismo, como concebido por Papert (1991). A seguir são discutidos alguns fundamentos da Linguagem de Orientada a Objetos e do próprio Etoys, ambiente onde é feita a investigação, para então propormos possíveis relações entre aprendizagem e a programação de jogos.

Na sequência são apresentados alguns resultados obtidos na experiência piloto, que introduzem as questões e delineiam a metodologia adotada para as Oficinas de Construção de Jogos. Por fim, apresentamos a análise da produção de jogos realizada pelos participantes das oficinas, focando os movimentos de aprendizagem destacados nos processos de construção de jogos.

II. REFERENCIAL TEÓRICO

2.1 Papert e o Construcionismo

Ainda no final da década de 60, época em que os computadores eram pensados como solução para fazer cálculos e armazenar grandes quantidades de informação, o matemático sul-africano Seymour Papert desenvolve estudos sobre a inteligência artificial, preocupado em construir objetos que fossem acessíveis e que se tornassem ferramentas para se pensar-com. Alinhado à Epistemologia Genética de Jean Piaget (com quem estudou por cinco anos em Genebra), passa ao estudo da Inteligência artificial no MIT (Massachusetts Institute of Technology) voltado à busca de novos recursos para a Educação, que denominou de uma perspectiva construcionista.

Papert, já naquela época, causou uma importante quebra de paradigma no que se concebia como sendo o uso do computador a serviço da aprendizagem: ao invés dos CAI (Computer Aided Instruction), softwares desenvolvidos para transmitir conhecimento ao usuário (no caso, o aluno), o autor propõe uma forma amplamente inovadora de ver essa relação: não é o computador “inteligente” que ensina um aluno “passivo”, mas um aluno ativo, ou inteligente, que ensina o computador. Desenvolve, então, uma linguagem de programação procedural chamada LOGO, com a qual mesmo uma criança pequena, através da composição de comandos em linguagem natural, “ensina” uma tartaruga virtual – criando programas - a fazer o que ela, criança, deseja (PAPERT, 1994).

Na busca de compreendermos a perspectiva construcionista concebida por Papert (1985), retomamos sua obra “LOGO: Computadores e Educação” para destacarmos algumas importantes contribuições. No trabalho citado, o autor procura explicar a perspectiva construcionista tomando como exemplo seu próprio processo de aprendizagem, remetendo-nos a sua infância. Papert relata que observava as engrenagens do automóvel na garagem de seu pai, “sentia-se orgulhoso por conhecer os componentes do sistema de transmissão, a caixa de câmbio (...) Isso aconteceu, é claro, muito antes de entender como as engrenagens funcionavam”, mas assim que passou a conhecê-las, a brincar com elas, passaram a ser “seu passatempo favorito” (*ibidem*, p.11).

Essas experiências, de acordo com o autor, definiram sua relação com a Matemática, que perdura por toda sua vida. Para compreender conceitos matemáticos

tomava o modelo das engrenagens para concretizar ideias abstratas. As “engrenagens da infância” passaram a se constituir “ferramentas para pensar com”. Assim é que nos traz a concepção de que a aprendizagem de qualquer conteúdo torna-se simples ao se incorporá-la aos modelos que constituídos. Ainda acrescenta:

A compreensão da aprendizagem deve ser genética. Deve referir-se à gênese do conhecimento. O que um indivíduo pode aprender e como ele aprende isso depende dos modelos que tem disponíveis. Isso impõe, recursivamente, a questão de como ele aprendeu esses modelos. As leis da aprendizagem devem estar em como as estruturas intelectuais se desenvolvem a partir de outras e em como, nesse processo, adquirem as formas lógicas e emocional. (p.13. Itálico do autor.)

Papert (1985) define essa “aprendizagem das engrenagens”, que desenvolveu ainda muito jovem, como “aprendizagem piagetiana”. Assim como Piaget, acredita que qualquer um é um construtor de conhecimentos, que desenvolverá ou não modelos de aprendizagem de acordo com as possibilidades de interação (em seu caso, as engrenagens) que lhe forem oferecidas. Uma importante qualidade para o autor é que essas ferramentas podem se constituir em modelos concretos, em objetos, por isso construcionismo, pois se constrói-conhecimento-com.

Podemos dizer, sobre a relação do construcionismo de Papert com o construtivismo de Piaget, que aquele procurou ir além deste, dando atenção especial ao afeto³ e aos objetos no processo de aprender. Além disso, as engrenagens de Papert não poderiam ser simplesmente transmitidas para outras pessoas, porque ele “se apaixonou” por elas, e esse componente emocional é também fundamental para que as engrenagens constituam um “objeto transitório” (*ibidem*, p.14), repleto de significações e ferramentas para a aprendizagem.

³. Ainda assim, a seguinte passagem de Papert (1985) vale a pena ser citada, por atentar de forma esclarecedora para possíveis leituras do papel do afeto (ou falta de) na obra de Piaget: “muitos anos depois, quando li Piaget, este incidente (evocação das “engrenagens” para aprender matemática, quando entrara na escola) me serviu como modelo para a noção de assimilação que ele propôs, apesar de ficar muito impressionado pelo fato de sua discussão não fazer justiça total às suas próprias ideias. Ele praticamente só fala sobre os aspectos cognitivos da assimilação, sem levar em conta o componente afetivo”. (...) “Estou certo que tais assimilações fizeram com que a matemática tivesse, para mim, um caráter afetivo que remonta às experiências com carros durante minha infância. Acredito que Piaget concorda com isso. Quando o conheci pessoalmente, entendi que sua negligência do componente afetivo deve-se mais a um sentimento de humildade diante do pouco que se sabe sobre isso do que a um sentimento arrogante de sua irrelevância (p.12).

O construcionismo de Papert se fundamenta no construtivismo, de Piaget, principalmente pelo entendimento de que o conhecimento é *construído* em estruturas que se internalizam a partir de ações, transformações, operações sobre o objeto. Qual seria o papel do computador e das linguagens de programação nesse processo?

A pergunta a essa resposta pode ser encontrada em Papert (1985) quando afirma que:

“o que as engrenagens não conseguem fazer o computador consegue. Ele é o Proteu das máquinas. Sua essência é a universalidade, seu poder de simulação. Por poder assumir milhares de formas e servir milhares de finalidades, pode atrair milhares de gostos” (p.14).

Com a criação e desenvolvimento da linguagem LOGO, Papert aposta em uma maneira de oferecer mesmo a sujeitos ainda não alfabetizados, por exemplo, a aprendizagem “piagetiana” de conhecimentos que, de outra maneira, poderiam ser de difícil compreensão, pois programar um jogo - como é o caso de nossa proposta - pressuporia conhecimentos complexos e abstratos. Muito além do uso da máquina como uma “ferramenta”, entretanto, é a comunicação com o computador (um objeto-para-pensar-com), através de linguagens de programação mais próximas à linguagem cotidiana que pode oferecer suporte para a aprendizagem de outros “modelos para aprender” (PAPERT, 1985, p.18).

Desde a criação da linguagem LOGO, as ideias construcionistas de Papert influenciaram outros pesquisadores a voltar sua atenção para o desenvolvimento de novas linguagens de programação a serviço da aprendizagem, como veremos a seguir.

2.2 Caracterização da Linguagem Orientada a Objetos – do LOGO ao Etoys

Os vínculos que unem o Etoys ao LOGO são intensos, sobretudo na sua origem. Comungam das mesmas concepções de aprendizagem e das formas de uso da IA (Inteligência Artificial) a serviço da aprendizagem. Entretanto, as inovações que diferenciam o Etoys da “tartaruga” de Papert são diversas. A compreensão dessas novidades torna-se fundamental para o entendimento sobre o funcionamento, os recursos, e as possibilidades do Etoys. Esse cotejo é importante para os propósitos da presente pesquisa uma vez que as próprias características da Linguagem também concorrem para o deslocamento dos objetivos do presente trabalho, como explicitaremos

adiante.

Foi influenciado pelas ideias construcionistas de Papert que, em 1968, Alan Kay cria a ideia do Dynabook, um pequeno computador móvel de tamanho semelhante a um livro que seria voltado para crianças, que à época não foi desenvolvido por limitações tecnológicas. A partir desse encontro com Papert que, segundo o próprio Kay, “mudou definitivamente sua trajetória” (2013), mais iniciativas surgiram no sentido de viabilizar novas linguagens de programação para o meio educacional.

É o próprio Papert que anuncia, ainda nos 80, alguns outros trabalhos que ocorriam paralelamente no MIT, enquanto trabalhava em novas versões do LOGO (PAPERT, 1985). Aponta para pesquisas desenvolvidas por Alan Kay em uma nova linguagem, orientada a objetos (ou LOO), chamada SMALLTALK. Papert (1985) atenta então para certas limitações do ambiente LOGO, em especial em relação a restrições à programação em paralelo (ou simultânea) de diferentes objetos. A resolução dessa limitação se fazia necessária para tratar de problemas mais dinâmicos e complexos do que aqueles resolvidos por programas em sequência, como a própria linguagem LOGO.

Naquele momento, Papert via o desenvolvimento de uma linguagem de processamento múltiplo como a principal meta de seu grupo LOGO no MIT, indicando o trabalho de Alan Kay em SMALLTALK como a maior aproximação possível do que pretendia fazer com o LOGO, mas que ainda não havia sido desenvolvido por problemas técnicos decorrentes da tecnologia da época (PAPERT, 1985). De fato, versões posteriores da linguagem LOGO a implementaram (STARLOGO, SPRITELOGO).

A diferença de uma linguagem de programação procedural ou sequencial em que os comandos são escritos de forma seriada, para uma linguagem que possibilite a programação em paralelo, ou simultânea, é essencial para o entendimento das inovações da linguagem orientada a objetos, como é o caso da Linguagem Etoys, que utilizamos nessa dissertação.

Cada linguagem de programação tem uma certa identidade, que podemos chamar de “paradigma” no qual foi desenvolvida. Esse paradigma, de acordo com a escolha da linguagem de programação, vai definir a forma como o usuário pensa a própria tarefa de programar (PAPERT, 1985). A fim de que possamos compreender inicialmente quais são as atividades possíveis e o que está em jogo quando um usuário ingressa no “mundo” do Etoys, é necessário caracterizar brevemente a Linguagem Orientada a Objetos (LOO).

Quando pensamos em uma linguagem “orientada a objetos”, estamos falando de um ambiente criado com o subjacente paradigma de que “programas simulam o mundo

real”, em um mundo criado na interação do sujeito com o computador (BARANAUSKAS, 1993). Essa característica, no entanto, não exclui a construção de programas e simulações que combinem a imaginação e relações sem paralelo com o mundo real, como o caso de um jogo. Os games comportam duas dimensões: consistem em regras reais através das quais os jogadores interagem agregando uma narrativa ficcional. Ganhar ou perder um jogo consiste em um evento real. Entretanto, ganhar um jogo derrotando um dragão indica sua dimensão ficcional. Jogar um jogo eletrônico consiste em interagir com regras reais em um mundo ficcional (Juul, 2011, p. 1). No presente trabalho incluímos ainda a perspectiva da própria criação do jogo, que implica, da mesma forma, as duas dimensões.

A ideia central, quando tratamos da linguagem orientada a objetos, é a de que vários objetos autônomos podem coexistir, receber e trocar mensagens entre si. Essa noção, no entanto, assim como toda a sintaxe da linguagem, não é dada ao sujeito, tampouco pode ser obtida intuitivamente, já que há regras arbitrárias que ditam como deve ser feita a comunicação aos e entre objetos.

A ideia central da LOO, descrita acima, diferencia a referida linguagem das demais por colocar a ênfase da programação na *relação* entre os múltiplos objetos, que podem funcionar autonomamente, através de mensagens trocadas entre eles. Mas do que se trata esse “objeto” de que falamos? Para buscar compreender tal conceito, voltamos para Alan Kay.

Em seu artigo “Computer software” (1984), Kay explicita muitas de suas ideias sobre a inteligência artificial, e mais especificamente no que tange a conceitos e técnicas necessários para colocar o computador a serviço da intenção do usuário. O software, de acordo com o autor, é o que dá forma e propósito a uma máquina programável, tal como a argila para o escultor, ou os instrumentos para o músico (KAY, 1984). A concepção que guia suas ações no estudo da IA é a de tornar a programação de um computador (por meio dos softwares) o mais próxima possível do usuário comum. Prosseguindo com a metáfora de uma obra feita de argila, cita o autor:

como na maior parte dos meios que usamos para construir um objeto, seja ele uma catedral, uma bactéria, um soneto (...) ou um processador de textos, a arquitetura domina o material. Entender a argila não é entender o vaso feito dela. A natureza do objeto pode ser melhor compreendida apenas se entendermos seus criadores e usuários, e esses precisam tanto dar significado ao material quanto retirar significado de sua forma. (KAY, 1984, p.3)

A relação de um usuário comum com o computador compreende um certo “mistério”

que ocorre pela dificuldade de acesso ao entendimento de seu processamento que se produz em vários níveis, desde o físico às diversas traduções lógicas. Se, por exemplo, aprendemos que se trabalharmos de certo modo a argila, podemos dar muitas formas possíveis, essa oportunidade geralmente não nos é dada em relação ao computador. A preparação da argila para aceitar determinada forma é um conhecimento que se produz na experiência tátil-cinestésica, ou como dizem muitos cozinheiros, “estar no ponto de”. Nossa interação com o computador se dá através de interfaces gráficas, softwares, que são desenvolvidos com propósitos de aproximar as mudanças de estado físico da máquina a uma linguagem. Essas interfaces são desenvolvidas para tornar o uso da máquina o mais simples e amigável possível, tornando a “linguagem das máquinas” cada vez mais distante do usuário. Kay (1984) utiliza a expressão “ilusão do usuário” para designar essa distância, e os significados atribuídos pelo usuário ao funcionamento da interface gráfica que utiliza.

A intenção de Kay, no trabalho citado, é mostrar – além da distância que o usuário comum tem do que subjaz a sua interação com a máquina – que cada linguagem de programação é desenvolvida através de certo tipo de pensamento, ou de metáforas. Esses tipos de pensamento utilizados na criação de uma linguagem podem servir tanto para “frear” quanto para auxiliar no desenvolvimento de novas formas de se comunicar com a máquina e, assim, a formação de novas culturas ou formas de pensar.

O que vai fundamentar suas ideias, então, no desenvolvimento da Linguagem Orientada a Objetos? Um dos princípios na sua forma de pensar está no raciocínio metafórico de que “essa parte da ideia é bem como essa outra, exceto por...”. Segundo o autor, esse método de pensamento é frequentemente usado, por exemplo, em programas que têm como propriedade a “herança” (inheritance) e a recursão; cada parte é também o todo: “uma descrição de todo o modelo é necessária para gerar a representação de uma de suas partes” (KAY, 1984, p.6). A LOO é uma das maneiras efetivas de colocar em prática esse tipo de pensamento, ou seja, um pensamento relacional e sistêmico.

Dentro da concepção da LOO, o computador é dividido (conceitualmente, através de seus poderes de simulação) em vários outros computadores, ou *objetos*, para cada qual pode ser atribuído um papel assim como a um ator de uma peça. A mudança de perspectiva – ou mesmo de paradigma – proporcionada pela linguagem orientada a objetos proporciona, segundo Kay (1984), um enorme ganho na capacidade de expressão. Assim ele ilustra essa possibilidade de mudança:

(...)“houve uma mudança similar quando cadeias de moléculas flutuando aleatoriamente em um oceano pré-biológico tiveram sua eficiência, robusteza e possibilidades energéticas aumentadas em bilhões de vezes ao se integrarem em uma membrana” (p.6)

Esses objetos, ou “atores”, são colocados em um mundo pelo usuário para poder criar novos sistemas, novas formas de pensar. O objeto é uma espécie de unidade fundamental de uma LOO, ao mesmo tempo em que, como exposto, reproduz, ou herda, as regras de todo o programa. É uma unidade autônoma, mas que pode modificar e ser modificada por todo o sistema que integra, já que “herda” as regras e o próprio funcionamento do qual faz parte, recursivamente.

No caso de um jogo criado em Etoys, por exemplo, o comportamento e as regras dadas a um objeto criado no mundo (os objetos existem em um background do programa, que igualmente é um objeto) como “personagem” podem ser instantaneamente herdados por múltiplos outros objetos, que por sua vez podem tanto modificá-lo como também serem modificados pelo objeto original. É importante frisar, novamente, que essa característica da linguagem pode ser ou não observável para aquele que o produz, configurando-se, então, em aspecto de interesse na observação dos mecanismos cognitivos de aprendizagem postos em evidência na atividade de programar.

Para Kay o computador está muito além de ser uma ferramenta, trata-se de um sistema que pode agir como outros sistemas (linguagem, máquinas) para ser explorado e alterado. No trabalho citado (1984), Kay e seus colaboradores não haviam ainda concebido o Etoys. Entretanto, sua forma atual foi desenvolvida a partir da Smalltalk (LOO original), especialmente para o uso no contexto escolar, e mantém – em seu funcionamento – os princípios descritos acima.

Cabe ressaltar que uma questão é a intencionalidade do criador; outra, são os modos de uso e de apropriação da LOO. Nesse estudo nos interessa analisar o quanto a interação com o Etoys pode contribuir para compreendermos mecanismos de aprendizagem que ocorrem com esses novos objetos. Interessa também flagrar se o processo de exploração a construção de jogos no Etoys favorece um fazer e um compreender sistêmicos, nos quais a interação entre os atores (no caso, objetos) pode fazer emergir novas ações e ideias.

2.3 Caracterização do Etoys

O Squeak Etoys, como chamado no começo de sua criação, é um ambiente de autoria multimídia desenvolvido em linguagem Smalltalk (LOO) a partir da interface Squeak. Pretende ser uma linguagem criada especialmente com a finalidade de facilitar à criança (ou ao usuário com pouca ou nenhuma experiência prévia) a aprendizagem de ideias poderosas (KAY, 2007) através do convite à programação (em nosso caso, de jogos).

Poderíamos relacionar as possibilidades de programação do Etoys com outras linguagens que serviram de inspiração a seus criadores: LOGO, STARLOGO e HYPERCARD.

Da linguagem LOGO, guarda a estratégia de facilitar a comunicação e programação – portanto, construção - de um objeto que serve de “vetor” às ações do programador, objeto esse com o qual ele se comunica através de comandos. Como inovação em relação ao LOGO, entretanto, permite que esse objeto projetado (antes uma tartaruga “pixelizada”) use qualquer “traje” definido -através de diversas mídias- pelo usuário, e ainda se multiplique em infinitos objetos, que podem se relacionar.

A possibilidade de trabalho com múltiplos objetos tem sua origem no STARLOGO, e consiste em poder lidar com dezenas, centenas, ou até milhares de objetos simultaneamente. Pretende permitir a simulação de fenômenos complexos da vida real, como a dinâmica de uma epidemia, a interação de moléculas em um composto, o comportamento de grupos em uma dada comunidade. Torna-se possível, assim, ao programador pensar em diferentes partes de um problema complexo, nas relações entre essas partes, e nas relações que essas partes têm com o todo de um determinado projeto.

Seria necessário acrescentar, para atentarmos às múltiplas possibilidades que temos de uso dos recursos do Etoys, sua inspiração em HYPERCARD, uma interface gráfica multimídia desenvolvida ainda na década de 80, antes da própria criação da internet. A ideia que influencia a interface do Etoys é a de que seja um integrador de diferentes mídias em um único espaço. Assim, múltiplas representações de funções diferentes podem ser combinadas sem a necessidade de programas especiais: tudo que se cria são objetos, e todos os objetos podem ser programados a partir dos mesmos princípios, que estão na própria linguagem do Etoys. Em um exemplo prático, seria como trabalhar com um vídeo, um processador de textos, um objeto de simulação, uma foto,

simultaneamente, como mostrado na figura abaixo:



Figura 1. Várias mídias integradas no Mundo

Através da integração de mídias, o Etoys possibilita interagir com o objeto digital em diversos níveis, do mais gráfico, ou “físico”, até os mais próximos da linguagem da máquina, ou de programação. Essa característica se faz muito importante, uma vez que permite trabalhar com os objetos computacionais de acordo com suas condições cognitivas e de apropriação de cada usuário, que se configura em um interessante dispositivo de pesquisa sobre processos de aprendizagem uma vez que cada sujeito ao modificar o objeto que programa põe em evidência os mecanismos cognitivos e de aprendizagem que lança mão.

As características descritas acima são facilmente constatáveis quando entramos em contato com o Etoys. Não há mecanismos que guiem, desde o princípio, o pensamento do usuário. Na primeira tela, constam apenas as opções de acesso a tutoriais, a galeria de projetos ou a abertura de um novo projeto, enquanto um objeto “carro” se movimenta pela tela, com seu script à mostra, como mostra a figura 2:



Figura 2. Tela Inicial

Quando, com um clique, iniciamos um novo projeto, deparamo-nos com o “Mundo” (figura 3, abaixo) do Etoys (o “fundo” que, como veremos, é de fundamental importância porque, assim como tudo no programa, também é um objeto programável).

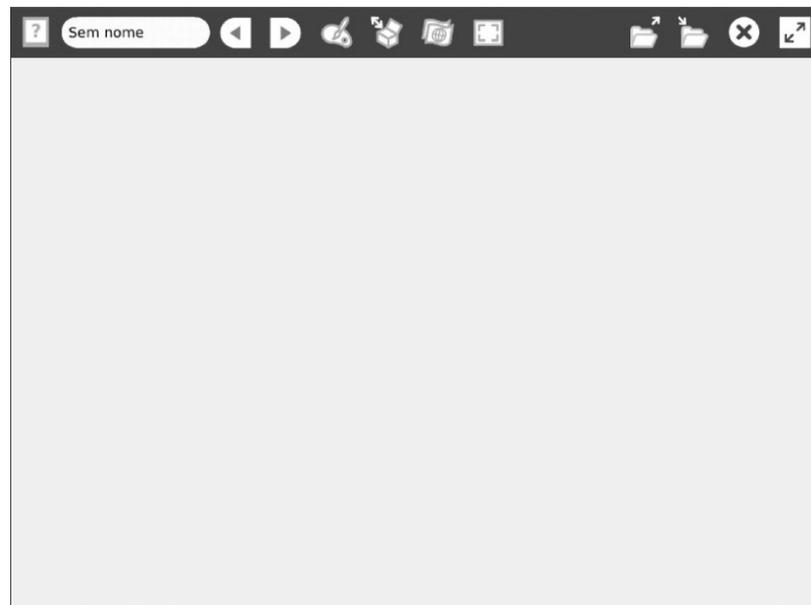


Figura 3. O Mundo do Etoys é também um Objeto

Há maneiras diferentes de iniciar a interação com o programa: podemos tanto abrir projetos já salvos, explorar os já existentes, quanto manipular objetos prontos ou, enfim, criar objetos os desenhando. Optamos por mostrar (na figura 4) a última opção, por ser

a atividade mais frequente com os estudantes que participaram das oficinas e, dessa forma, ilustrativa das situações que iremos analisar posteriormente.

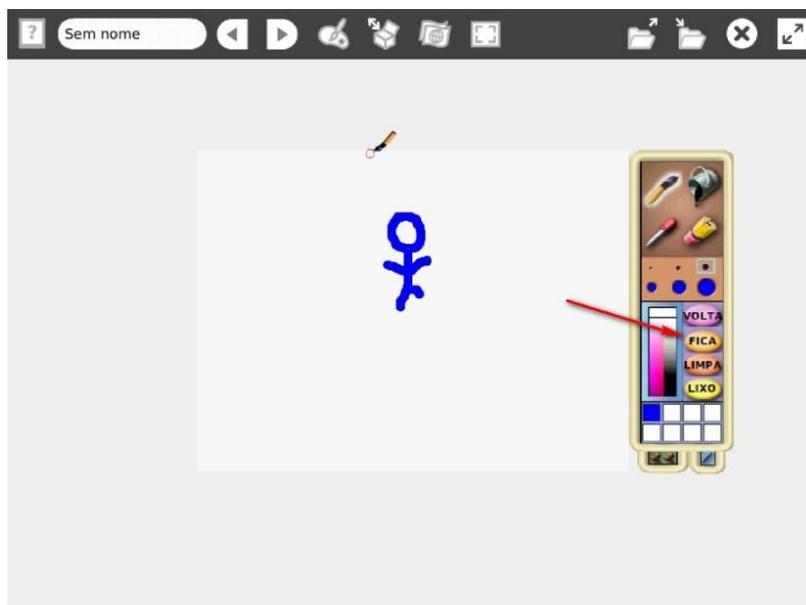


Figura 4. Criação de um Objeto a partir do desenho

Ao iniciar o desenho de um objeto, são disponibilizadas ao usuário diversas ferramentas de pintura, como as presentes em diversos softwares básicos dessa função. É a partir da imagem produzida, entretanto, que há uma diferenciação essencial: a possibilidade de transformar esse desenho “simples” em um “objeto”. Para realizar essa transformação, nada mais é exigido do que pressionar a tecla “fica” (conforme figura 4). Ao fazê-lo temos, então, a mesma figura desenhada, mas que transformada em objeto ganha uma série de funcionalidades, que possibilitam múltiplas ações diferentes (figura 5). Se a transformação de um desenho em um objeto ocorre com um simples toque de uma tecla não quer dizer, necessariamente, que o sujeito ao acionar tal tecla também realize essa transformação conceitual de modo cognitivo. Explorar essa compreensão pode trazer interessantes elementos de análise aos processos de aprendizagem.

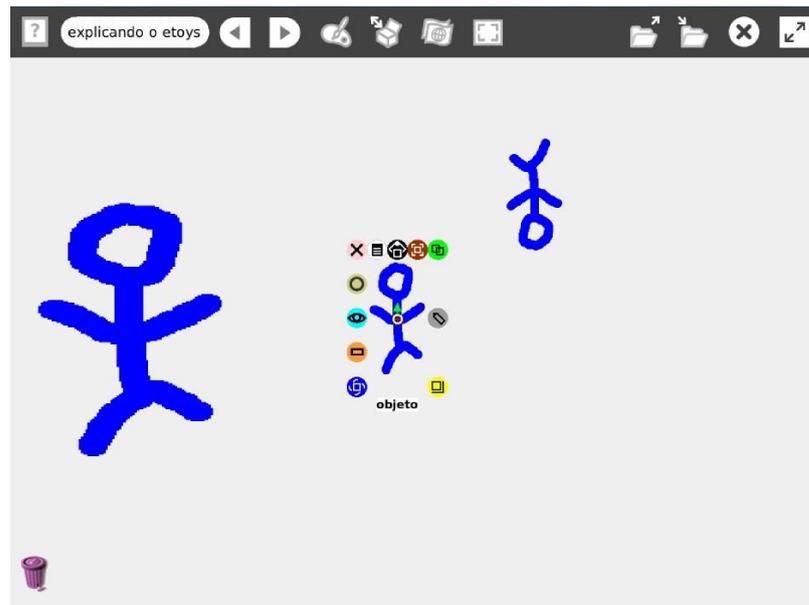


Figura 5. Ações gráficas a partir das handles

Várias interações gráficas são possíveis a partir da criação de um objeto. Ele pode ser manipulado graficamente pelo movimento de “arrastar” sobre a tela, com o cursor do mouse e tem certas funcionalidades gráficas apresentadas pelas “handles” (ferramentas), quando sobre ele clicamos com o botão direito. Essas propriedades contidas nas *handles* - comuns a qualquer objeto - permitem agir graficamente sobre ele de diversas formas: girando, duplicando, aumentando ou diminuindo seu tamanho, repintando, etc. Todas essas ações são possíveis sem que haja a necessidade de entrar em contato com o código subjacente ao objeto: sem que seja necessário programá-lo através desse código. Podemos, pois, criar um cenário - com um ou muitos objetos diferentes - com diversos tamanhos e cores, etc., sem que seja necessário qualquer contato com o código. Esses diferentes níveis de programação, postos ou não em ação através da construção dos jogos, se configuram em um aspecto interessante de estudo quanto aos processos cognitivos envolvidos na interação com o ambiente.

A programação do objeto é que cria a necessidade de se entrar em contato com o código subjacente ao ambiente. Os códigos da linguagem, na verdade, estão sempre presentes e definem as propriedades do objeto, mas são apenas visualizados através da handle “visualizador”, conforme figura 6.

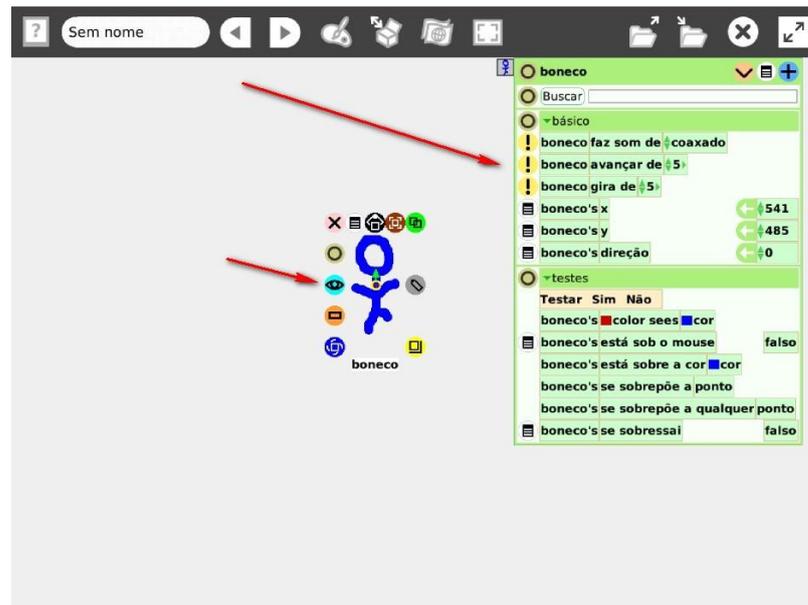


Figura 6. Acesso aos comandos pelo visualizador

O clique no visualizador fornece, então, acesso ao código que está “por trás” do objeto. Passa a ser disponibilizada uma lista de comandos, propriedades e atributos por blocos, organizados em categorias de programação (básico, movimento, gráfico, etc). Todas as propriedades dos objetos, mesmo as gráficas, estão – em forma de comandos – apresentadas nos blocos de programação. Assim, através da sintaxe, temos acesso e podemos agir sobre as diversas propriedades e atributos, incluindo as geométricas (posição na tela, por exemplo), gráficas (largura, altura, cor), de som (som emitido pelo objeto, frequência desse som), etc. Também pela lista de comandos disposta pelo visualizador, podemos executá-los sem que se tornem automáticos em *scripts* de programação, testando-os um a um e observando o efeito sobre o objeto correspondente.

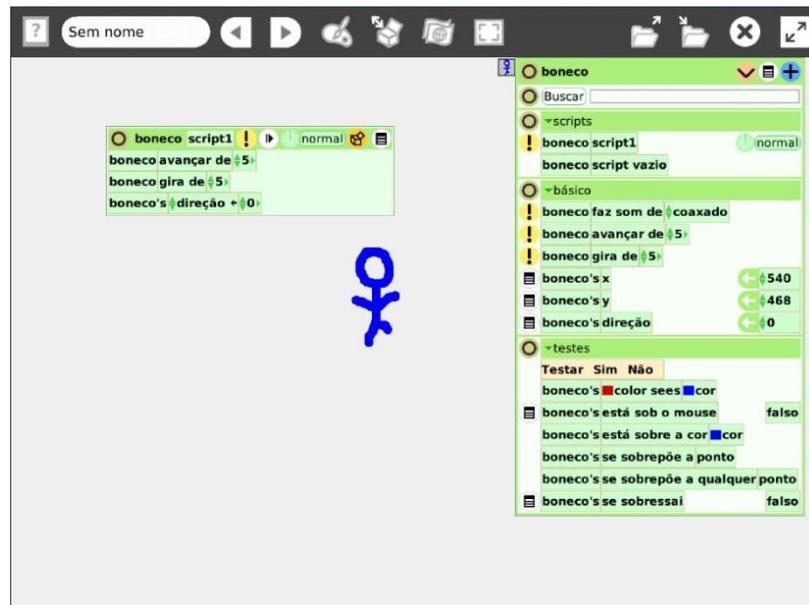


Figura 7. Criando Scripts com vários comandos

É somente com a criação de *scripts* que podemos fornecer ao objeto comandos e torná-los automáticos (figura 7, acima), sendo que, para fazê-lo, não é necessário mais do que o “arraste” de um comando para o “mundo”. Essa ação basta para que um *script* seja automaticamente criado, contendo o comando arrastado, e para que se possa, através de um clique, fazer com que esse comando seja executado diversas vezes por segundo. Um exemplo básico seria criar um *script* com o comando “avançar 5” (ou cinco pixels, unidade de distância do programa). Se executado, o *script* será acionado um número de vezes por segundo, e então teremos o resultado de movimento do objeto avançando na direção apontada.

Múltiplos *scripts* podem ser criados para um único objeto, e múltiplos comandos podem ser colocados dentro desse *script*. Da mesma forma, podemos colocar comandos referentes a um objeto X em um *script* de um objeto Y, e assim por diante. Essa forma de disposição e organização da linguagem de programação faz com que efeitos de movimento e simulação sejam possíveis com poucas ações do usuário. Da mesma forma, objetos diferentes podem enviar e receber mensagens entre si, através dos comandos colocados em seus *scripts*.

O conceito de “herança” (inheritance) é, então, constatado da seguinte forma: quando criamos uma cópia de um objeto qualquer, esse novo objeto “herda” propriedades, atributos, e *scripts* do original. Esse mecanismo – conceitualmente básico da LOO – na medida em que é compreendido, é também um facilitador na criação de simulações de problemas dinâmicos, já que se pode, a partir da definição do comportamento de um

objeto, ter logo o resultado de muitos objetos, idênticos ao primeiro, em funcionamento na tela do computador. Através dessa característica, também são abertas muitas possibilidades de relação e de coordenações entre as partes (objetos) e o todo (ambiente, micromundo) de um determinado projeto.

As relações entre objetos podem ser feitas de diversas maneiras, sendo a troca de mensagens entre os objetos outro conceito básico da linguagem. A programação dessas relações – ou trocas – é muito diversa e ocorre em vários níveis diferentes. Inserir o comando de um objeto no *script* de outro pode já ser considerado uma comunicação. Há, também, comandos com mensagens específicas de um objeto para outro, ou outros.

A criação de proposições condicionais merece ser aqui destacada: criamos uma relação entre objetos diferentes com uma condição. Assim, por exemplo, dois objetos de cores diferentes se relacionam se criamos a condição “se a cor do objeto X 'vê' a cor do objeto Y, então (...); se NÃO, então (...)”. As relações possíveis a partir da criação de condições como essa são praticamente inesgotáveis. Destacamos, também, que um evento a ser satisfeito – ou não – pela condição criada não está imediatamente visível ao sujeito: para se apropriar de tal procedimento, é necessário ultrapassar estratégias de ensaio e erro, para efetuar antecipações sobre as ações possíveis.

As características acima comentadas tornam o Etoys um interessante dispositivo de pesquisa quando foco consiste nos processos de aprendizagem. Observamos que se trata de uma linguagem ao mesmo tempo simples e robusta, que pode operar com distintos modos de ação e compreensão, com quem com ela interaja. Tal aspecto da linguagem não é dado ao acaso: o Etoys, assim como outras linguagens de programação que são desenvolvidas para uso na educação, têm incorporadas a característica de “chão baixo” e “teto alto” (low floor; high ceiling). Por “chão baixo”, entende-se a facilidade ou simplicidade para começar um projeto; por “teto alto”, a possibilidade de dar mais complexidade ao projeto no decorrer do tempo. Resnick (2009), retomando as ideias de Papert (1992), ainda acrescenta a necessidade do ambiente ter “paredes largas” (large walls), no sentido de suportar muitos tipos de projetos diferentes e, então, usuários com “interesses e estilos de aprendizagem diferentes” (2009, p. 63).

A breve apresentação acima realizada é apenas um esboço de entrada no ambiente de programação, com a intenção de familiarizar o leitor e apresentar as funcionalidades básicas e a organização do Etoys. Ao mostrar as primeiras possibilidades de ações e interações que decorrem da exploração do ambiente, tentamos apresentar brevemente a concepção que orienta o programa e que o diferencia de outras linguagens.

Nosso interesse para a investigação está em observar os mecanismos cognitivos postos em ação na atividade dos estudantes em interação com uma linguagem desenhada com tais características e, especificamente, na produção de jogos utilizando o Etoys.

2.4 Aprendizagem e programação de jogos digitais

A caracterização acima realizada da Linguagem Orientada a Objetos, materializada no Etoys, trouxe evidências de sua riqueza para ser tomada como um dispositivo para a análise dos processos e movimentos da aprendizagem durante a atividade de programação de jogos.

Em uma perspectiva construcionista, a atividade de programar um computador pode ser vista como uma alfabetização digital, que apresenta características distintas de outras modalidades de alfabetização (Gee, 2007). A “alfabetização”, para Papert (1994, p.17), teria um sentido mais amplo do que a decodificação de símbolos: tornar-se alfabetizado significa “pensar de uma forma diferente do que anteriormente, enxergar o mundo de outra maneira”. Essa nova forma de enxergar o mundo, comporta, também, o conjunto de práticas sociais envolvidas nos modos de pensar, avaliar, criticar e, sobretudo, construir algo de valor em um determinado grupo de pessoas. Desse modo, alfabetizar-se é mudar a si mesmo e ao mundo.

Em uma cultura digital, ser alfabetizado estaria ligado à *fluência* de uso com as tecnologias digitais. Para pensar o que seria essa fluência digital, Resnick (2002) fala na possibilidade de construir “coisas significativas” com as tecnologias digitais: para além daquele usuário que navega na web, que ouve músicas, que vê vídeos ou que joga jogos, aquele que pode produzir suas músicas, produzir e editar seus vídeos, desenvolver seus jogos, etc. A alfabetização traria a condição de cidadania em habitar determinado mundo. Ao compreender e operar com as regras constitutivas de determinado mundo é possível criar.

Ao propor a programação de jogos em Etoys, como recurso para estudar processos de aprendizagem pensamos ir além de uma posição de sujeito que decodifica uma linguagem: o computador é um objeto-para-pensar-com, sendo o Etoys um sistema complexo – um mundo – com o qual o sujeito vai interagir. Alfabetizar-se é inserir-se e compreender as regras desse sistema, tendo assim capacidade de criar.

A programação de um jogo abre espaço para discussões em níveis diferentes: ela comporta a atividade de desenho (design) ou projeto do jogo, a constituição de uma narrativa e de jogabilidade pela integração de mídias interativas (BRENNAN E RESNICK, 2012) com o código de programação do Etoys. Há, também, a atividade de jogar o próprio jogo, ou o jogo criado por outrem. Então temos: a criação (design), a programação, e a atividade de jogar o produto digital. Acreditamos que, nos três níveis, há processos de aprendizagem envolvidos: quais processos seriam? Tentaremos, nesta seção, buscar subsídios teóricos para discutir esses aspectos.

Ao reunir contribuições de autores que consideram as três dimensões mencionadas buscamos relacionar discussões do campo dos videogames (Games Studies), da programação de computadores e suas relações com aprendizagem.

Com o intuito de criar meios de compreensão sobre a produção de projetos em Scratch ⁴, Brennan e Resnick (2012) buscam desenvolver, fundamentados no construcionismo de Papert, um framework para tornar possível o estudo do desenvolvimento do que chama de “pensamento computacional”. Os autores consideram que programar um jogo, uma história animada, um cartão postal interativo, etc. são “atividades baseadas em design”. Nesse tipo de atividade, estariam envolvidas três dimensões principais, que estão incorporadas na própria atividade: “conceitos computacionais” (que aqui chamaremos de conceitos de programação), “práticas computacionais” e “perspectivas computacionais”.

Apresentaremos, resumidamente, algumas ideias gerais sobre as atividades de design e programação de Brennan e Resnick (2012) em seu framework, tomando de empréstimo alguns conceitos específicos que envolvem o caso da programação de jogos em Etoys.

Ao programar um jogo o usuário terá que entrar em contato com um conjunto de conceitos de programação mais ou menos comuns a várias linguagens. Brennan e Resnick (2012), para o caso da atividade de design em Scratch, identificaram um conjunto de sete conceitos: sequência, loops, programação paralela, eventos, condicionais, operadores, e uso de dados. Como *sequência*, entende uma série de passos ou

⁴O Scratch é um ambiente computacional desenvolvido por Resnick e seu grupo Lifelong KinderGarden no MIT. Criado na mesma linguagem orientada a objetos que o Etoys, o Squeak, trata-se de uma tentativa de tornar a programação acessível a todos os usuários, sobretudo aqueles sem experiência prévia. O Scratch é desenvolvido sob os mesmos princípios epistemológicos que o Etoys, que influencia e serve de inspiração para nova linguagem. Ao criar e desenvolver o Scratch, no entanto, tentou-se ir mais além, tornando o “chão mais baixo e as paredes mais largas” (RESNICK, 2009). Algumas aproximações e diferenciações serão problematizadas na sequência do trabalho.

instruções para serem executadas pelo computador; sendo o *looping* o mecanismo para executar a sequência um certo número de vezes. Um *evento* constitui um artifício que é a “causa” de outro (por exemplo, um botão que dispara um script, o aparecimento de um texto ou imagem, etc.). *Paralelismo, ou programação paralela*, são programas, ou sequência de programas acontecendo simultaneamente, como é o caso de vários objetos movimentando-se em um jogo, por exemplo. *Proposições condicionais* que, como veremos, podem estar diretamente ligadas ao evento, consistem na definição de um acontecimento sob determinadas condições (se um objeto está em tal posição, então ele aparece, se não ele desaparece, etc.). Os *operadores* são atributos matemáticos contidos e disponíveis nos próprios comandos: podemos, por exemplo, dar a um objeto o comando de avançar 50, multiplicados por 5, ou subtraído por 10, etc., assim como outras funções matemáticas mais complexas. O uso de *dados* se refere ao armazenamento ou a atualização de certos valores nos objetos, tendo os exemplos mais usuais no uso de variáveis ou listas (BRENNAN E RESNICK, 2012).

Os conceitos apresentados, no entanto, não são suficientes por si só para explicar o processo em uma atividade de design e programação de um jogo. Brennan e Resnick (2012) atentam para o que chamam de “práticas de pensamento computacional” (p.6). Essas práticas dizem respeito ao que pode ser observado do processo de pensar e aprender enquanto se programa: se os conceitos de programação seriam “o que” se aprende, as práticas seriam o “como” se aprende. A partir da observação de jovens programadores de Scratch, quatro categorias de estratégias, ou práticas, foram estabelecidas: ser incremental e iterativo; testagem e debugging; reuso e remixagem; abstração e modularização.

Desenhar um projeto de programação não é um processo sequencial e direto, em que primeiro traçamos um plano geral de design e então implementamos um código, passo a passo. Conforme Brennan e Resnick (2012), o design e a implementação de um projeto em Scratch envolve uma adaptação mais ou menos constante em que a solução para os problemas é obtida a partir da experimentação de partes do processo, que modificam o plano geral. Os autores descrevem como que ciclos de aprendizagem que envolvem: imaginar e construir, experimentar, testar, imaginar novamente, etc. A esse “ciclo”, que repete e ao mesmo tempo incrementa o projeto, os autores chamam “iteração”. Contidas nesses ciclos, observou como fundamental as práticas de teste e debugging: ao criar um script para determinado objeto, além da prática de tentativa e erro, também a antecipação do resultado daqueles comandos a serem dados ao objeto (em resumo:

identificar o problema, leitura dos scripts, experimentação, nova tentativa de compor o script).

Ainda como fundamental estão as práticas de reuso e remixagem. Dentro desse grupo de práticas, poderíamos incluir tanto o uso de uma ideia de projeto para ser recriada (como por exemplo, tentar reproduzir um jogo já conhecido), como códigos a serem transportados e remixados dentro de um projeto (nesse ponto, o autor dá especial importância a possibilidade de acesso a comunidades de programadores online, por exemplo). Em uma quarta categoria, se inclui o que o autor chama de abstração e modularização: no processo de desenhar um projeto e ao programá-lo, há a construção (em imaginação ou em ação) de algo maior pela coordenação de partes menores. Na construção de um jogo com múltiplos personagens e comportamentos diferentes, por exemplo, o usuário separa as “categorias” de comportamento, ou de personagens por sua programação, e com esses “módulos” pode pensar, testar, com partes com as diferentes partes do projeto.

Outra dimensão da atividade baseada em design, trazida por Brennan e Resnick (2012), é o que os autores chamam de “perspectivas” da atividade de programar. Relembrando o conceito de fluência digital, ponderam que as experiências com mídias interativas que nos são possibilitadas são geralmente como consumidores: navegando na internet, assistindo vídeos, lendo textos, etc. Salientam então que, ao desenvolver um projeto autoral, programando ou integrando mídias, há três possibilidades fundamentais que tornam essa atividade diferente: a criação e expressão de ideias próprias; a conexão, ou o acesso à comunidades de aprendizagem e as interações por ele possibilitadas; o uso crítico, ou a possibilidade de se questionar sobre a tecnologia digital que está constantemente frequente, mas que normalmente é incompreensível ao sujeito.

Em nosso trabalho de criação e desenvolvimento de jogos em Etoys, além da aprendizagem de recursos da programação, outras perspectivas se abrem: a do criador, ou designer, e a do jogador, que pode jogar o game criado por ele mesmo ou por um colega. Assim, pensamos ser importante trazer à discussão algumas características apontadas na literatura sobre as várias dimensões de aprendizagem exploradas não só por um jogador de videogames, como também de um designer de jogos. Nosso intuito é também considerar a possibilidade de imersão no Etoys, assim como se, na atividade de programar um jogo, podemos também observar aprendizagens relativas a como um jogo funciona.

2.5 Aprendizagem e Videogames

Para esse trabalho é interessante discutir as ideias acima com as de autores no campo dos estudos dos videogames (GEE, 2004, SQUIRE, 2011) apontando associações possíveis entre seus argumentos. Uma delas é que, assim como comentamos para os Etoys, os videogames propiciam uma interação com sistemas complexos. Conforme GEE (2007), o contato imersivo com videogames inclui possibilidades de fazer inter-relações entre diferentes mídias (imagens, palavras, ações, símbolos, artefatos, etc.) integradas em um sistema, o que estimularia o desenvolvimento de noções, “intuição de como sistemas funcionam” (SQUIRE, 2011).

Alinhados às ideias de Resnick e, portanto, com o construcionismo, algumas pesquisas no campo dos videogames têm estimulado seu uso como uma potente interface para a educação e para o desenvolvimento da fluência digital. Alguns trabalhos dessa vertente têm dado especial atenção para a atividade de construção e programação de jogos digitais e as diferentes formas de aprendizagem nela envolvidas (MACKLIN e SHARP, 2012; KAFAI e PEPLER, 2012).

Compreendendo ambientes de programação (como o Scratch ou o Etoys) como sistemas semióticos complexos, Kafai e Pepler (2012) sustentam que aprender a construir um jogo (design) pode ser um meio valioso para engajar jovens em práticas que simulam a operação entre objetos conectados por regras em interações recorrentes. Aproximando o conceito de fluência digital às práticas ligadas ao trabalho com games, os autores propõem o termo “gaming fluencies”, estando contida nessa fluência não só o game design mas, também, além os aspectos “criativo, crítico, e técnico de trabalho com as novas mídias” (p.355). Destacamos, aqui, a preocupação dos autores em criar objetos de aprendizagem que relacionem as atividades possibilitadas por sistemas como o Etoys: a criação, a programação, e o jogo.

Pensamos que essas características também possam ser interessantes para o nosso caso, no qual os estudantes são convidados a criar seus jogos. Como base para essa discussão retomo os pontos levantados por Maraschin e Baum (2013).

O primeiro ponto comentado no referido artigo consiste em pensar os videogames como experiências projetadas. Os autores perguntam o motivo da existência da proposição de que jogar videogames configura uma experiência e não somente a realização de ações previamente determinadas pelo jogo? Em sua resposta remetem a

Squire (2006) que propõe que os videogames não sejam entendidos como códigos estáticos que devem ser lidos ou interpretados, como se existisse uma “essência do jogo”, um único (e correto) modo como ele deve ser jogado. Os games, segundo o autor citado oferecem uma experiência projetada na qual o importante é compreender os modos de habitar esses espaços, os mecanismos utilizados para criar sentidos e como esses sentidos e experiências emergem em diferentes contextos do jogo. Deste modo, os jogos se configuram como domínios semióticos (GEE, 2007) e de ação nos quais o jogo só acontece quando existir uma coerência operativa entre o jogador e a jogabilidade do game. Essa coerência se produz em uma articulação que é, ao mesmo tempo, sensório-motora e semiótica, através da qual se compartilham modos de ação e de significados (BAUM & MARASCHIN, 2013). Portanto, faz sentido pensar que os videogames possam ser interessantes para compreender a emergência de modos cognitivos em configurações nas quais a experiência se produz pelas congruências operacionais e coordenações de ações entre o sistema programado e a atividade do jogador. Acreditamos que essa análise poderia ser ampliada no momento que os estudantes não são somente convidados a jogar, mas a produzir seus jogos pela qual devem situar-se também nas possíveis experiências dos jogadores.

Para Gee (2007) a experiência nos videogames emerge a partir de um casamento entre as regras e os elementos do jogo. O programador/jogador deve criar um mundo considerando uma sinfonia visual, motora, sonora, sinestésica e de tomada de decisão. A produção dessa experiência é uma ação efetiva de cada programador/jogador; por isso, o programador deve conceber um modo desenvolver o jogo no qual o jogador possa escolher distintos trajetos lhe dando a condição dessa experiência. Para o autor acima citado, o jogo se produz em uma nova forma de arte performance entre jogadores e designer, que no caso de nosso estudo, podem ser a mesma pessoa ou os colegas de turma. Considerar a posição do jogador resulta da interação entre o planejamento do jogo e aquilo que pode ser efetivamente jogado. O jogo seria então uma co-construção entre o designer e o jogador. Nessa atividade emergente o programador deve considerar o jogador como ativo pois relee os objetivos, os caminhos, que são condicionados, embora não determinados, pelos parâmetros do jogo. Uma das dificuldades de produzir games é pensá-los como espaços participativos e de experimentação que convidam os jogadores a habitar avatares com capacidades particulares e diferenciadas podendo, em um mesmo jogo, eleger diferentes maneiras de ser/estar nesse domínio.

Na literatura existe um debate do motivo pelo qual os jovens preferem os

videogames aos jogos educativos. Para responder a esse questionamento, Squire (2011) traça algumas diferenças significativas envolvendo principalmente a ação do jogador, o conhecimento e o contexto da aprendizagem. Os jogos educativos, de modo geral, têm como principal função transmitir um conjunto de conhecimentos aos seus usuários. Mais que o desenvolvimento de habilidades específicas, esse tipo de jogos se organiza a partir da recorrência e hierarquização dos conteúdos. Existe neles uma premissa de que o conhecimento pode ser equiparado a um conjunto de fatos que se organizam geralmente dos mais concretos aos abstratos e dos mais específicos aos mais gerais. As habilidades e conhecimentos do usuário pouco interferem no processo de aprendizagem e qualquer recurso exterior ao software não é válido, eventualmente sendo considerado trapaça. O jogo educativo é exógeno, o contexto e a motivação encontram-se fora de sua operatividade/jogabilidade, geralmente ligados à autoridade escolar e/ou a avaliação. Nos videogames, entretanto, existe uma preponderância da ação, do desenvolvimento de certas habilidades que qualificam o jogador para seguir no jogo e da construção conjunta de significados. Os conteúdos são atualizados sob demanda, ou seja, na medida da necessidade da ação. A maioria dos jogos é endógena, ou seja, a motivação e o conteúdo são indissociáveis da jogabilidade e os valores se constroem no desdobrar do próprio jogo (SQUIRE, 2006).

A qualidade dos videogames simularem espaços de exploração e de ação advém se constituírem em sistemas dinâmicos complexos que operam de modo semelhante às simulações. Mas, como observamos anteriormente, uma simulação não significa uma representação de um espaço de experiência já constituído, mas uma composição entre real e ficcional. Essa característica real-ficcional possibilita aos videogames criar mundos nos quais seja possível experimentar outras formas de acoplamentos e ações, que seriam muito difíceis de ser exploradas em contextos concretos. O que também não significa dizer que essa experiência não seja real. Segundo Maturana e Varela (1995) temos tantos espaços de realidade quantos acoplamentos efetivos pudermos constituir coletivamente.

Um segundo ponto comentado por Maraschin e Baum (2013) consiste no tipo de conhecimento que é mais solicitado ao se jogar videogames, fazendo apelo a uma cognição mais operativa que declarativa:

Varela (1996) retoma a distinção feita por John Dewey, no início do século XX, entre know-how e know-what para estabelecer diferenças entre esses modos de conhecer. Os modos de conhecer que se dão pelo julgamento abstrato, por uma espécie de “saber sobre”, que se pergunta sobre as coisas (ou mesmo sobre si) e tenta explicá-las através de uma atitude reflexiva e racional produzem conhecimentos do tipo “saber o quê”, “saber-sobre” (know-what), enquanto em nossa experiência cotidiana domina a esfera do “saber-como” (know-how), ou seja, um conhecimento em

ação, corpóreo, vivido e inseparável da história e das contingências que o compõe (p. 258).

Para os autores, nossa cultura e os modos como a escola ensina acabam atribuindo valores diferenciados aos dois conhecimentos. O conhecimento incorporado, da experiência é tido como inferior por seu menor poder de abstração e de generalização. O know-how permaneceria localizado, contextualizado. Mas, em contrapartida, em situações nas quais é necessário buscar solucionar um problema concreto, o know-how pode ser mais efetivo, porque se constitui como uma disponibilidade para a ação. Gee (2007) ao comparar o modo como a escola ensina com o modo como os jovens aprendem a jogar videogames diz que se tentássemos ensinar como jogar videogames do modo escolar faríamos os estudantes aprender o jogo pelo manual e não pelo jogo em si. Segundo Varela (op. cit), é o saber em ação, encarnado, que abre para a invenção, para a criação, justamente, por sua efetividade e por fazer face frente ao inesperado do acontecimento. Seu principal mecanismo é o *breakdown*. Esse termo designa aquelas situações em que há uma espécie de quebra na continuidade cognitiva, aqueles momentos nos quais a cognição, em uma determinada ação presente e imediata, experimenta um problema e hesita quanto ao que fazer. As experiências de *breakdown* possibilitam a invenção por que, segundo Kastrup (1999), produzem um “movimento de problematização das formas cognitivas constituídas” (p.15). Entendemos que a experiência de projetar mundos nos games articula esses dois modos de cognição, pois aquele que projeta um jogo necessita pensar nas possibilidades do jogador, articular o saber-sobre com o saber-como.

Outra característica dos videogames consiste em sua ludicidade. A maioria dos games são divertidos (lúdicos) porque disponibilizam ao jogador uma posição projetiva – uma ligação entre o mundo digital (jogo) e o mundo que denominamos real (vida fora do jogo). Tal verossimilhança permite uma experiência na qual podemos vivenciar, simultaneamente, o mundo como um projeto imposto sobre nós e como um espaço no qual podemos ativamente projetar nossos desejos, valores e objetivos (GEE, 2007).

De modo geral o jogador atua através de um *avatar* que produz essa ligação entre mundos. O domínio de ação-conhecimento do jogo é, ao mesmo tempo, uma restrição e uma condição de possibilidade da sintonia entre *avatar* e jogador.

A ludicidade é efeito do processo de produção dessa sintonia. Portanto, não é algo que somente acarreta prazer ou diversão, pois em muitos momentos as dificuldades levam a frustrações momentâneas. O que explica a vontade continuar jogando é, segundo Gee (2007), muito mais a paixão pelo desafio, pela aprendizagem do que a diversão em si. Essa paixão pode ser exercitada pois nos videogames, as consequências

de um erro não são devastadoras: o jogador pode recomeçar do último *check point* salvo. Além disso, a seriação dos desafios possibilita um acoplamento gradual entre jogador e *avatar*. No caso de programar o jogo agrega-se o desafio de criação e a possibilidade de ir desvendando esse mundo de construir ferramentas para que outros as experimentem.

Segundo Squire (2011) os jogos multiplayer amplificam essas possibilidades, uma vez que os personagens, ao serem redefinidos pelos jogadores, incrementam as possibilidades de interação, incluindo modalidades culturais, complexificando o sistema de valores inicialmente projetado.

Uma característica mencionada na literatura dos games é que os mesmos agregam pessoas com interesses semelhantes, o que denominam espaços de afinidade. Para Gee (2003, 2007) Videogames com V maiúsculo incluem não somente o jogo, mas as redes sociais de interação derivadas do mesmo. O autor questiona denominar de “comunidade de prática” os coletivos que se afiliam a determinados temas na internet e que compartilham questões educacionais e de trabalho (WENGER, et.al., 2002). No caso dos games, prefere a denominação de “espaços de afinidade” (GEE, 2007 p. 87). Segundo o autor, a ideia de comunidade pode levar a sentidos de pertencimento e de relações de consensualidade que não se adequam ao conjunto das relações existentes nos espaços de trocas gerados pelos videogames. Para o mesmo autor, pensar em espaços de afinidade, permite ampliar o sentido das relações, em encontros que não seriam somente aqueles proporcionados pelos jogos multiplayer.

Ao analisar os fóruns do game “Age of Mythology”, Gee (2004) evidencia que os mesmos são formados pelo compartilhamento de ações e conteúdos mais do que por marcas identitárias ou vínculos institucionais. Gee descreve algumas características dos fóruns que possibilitam ser pensados como espaços de afinidade: (a) a participação é aberta a qualquer usuário; (b) iniciantes e veteranos compartilham o mesmo espaço; (c) os participantes tem poder de definir e transformar o ambiente; (d) conhecimento é compartilhado entre os jogadores; (e) existem muitas formas de participar e (f) a participação nos espaços de afinidade remodela o próprio o jogo mesmo, ao propiciar feedbacks aos desenvolvedores de jogos ou por explicitar modos de modificar os jogos que assim o permitem (“*modding*”).

Os espaços de afinidade se relacionam ao interesse diante de algum processo/habilidade/conteúdo. É um espaço de algo/espaço para algo. Para o autor acima citado, é possível analisar o conteúdo nesses espaços de afinidade de dois modos. De um modo direto, ou seja, de como ele está organizado, o que está presente e o que

está ausente. Mas também podemos pensar de um modo indireto, tal como nos modos como os integrantes interagem com ele. No caso dos videogames, focar o conteúdo, de um modo direto, é perguntar-se sobre o design desses jogos, enquanto que focar na via interacional é perguntar como as pessoas organizam seus pensamentos, crenças, valores ações e as interações sociais em relação aos sinais que estão disponíveis em nos jogos (GEE, 2004). Mas, o mais interessante nos videogames é que podemos interagir diretamente com os conteúdos no tempo real das estratégias do jogo. Conteúdo e interação estão juntos uma vez que o *avatar* é, ao mesmo tempo, parte e não parte do conteúdo do jogo, pois também depende de um jogador que o opere. Como dissemos anteriormente, há uma relação entre saber-sobre e saber-fazer. Essa imersão com o conteúdo do jogo é uma experiência inusitada que pode ampliar modos de diferenciados de um operar cognitivo.

O que vimos discutindo concorda com a posição de Squire & Giovanetto (2008) segundo a qual os games podem ser considerados mais que tecnologias, são meios que comportam práticas sociais. Por exemplo, nos jogos denominados de realidade expandida, o jogo provoca uma sinergia com o contexto no qual ele é jogado, o que implica que tanto os jogadores, como os protagonistas que são contatados (pessoas da comunidade que dispõem de informações para a continuidade do jogo) fazem parte dos espaços de afinidade. Nessa perspectiva, os games passam a ser um modo de exploração de mundos na cultura.

Acompanhando os modos de jogar de diferentes jogadores Squire (2011, p.34) traça uma espécie de trajetória comum pela qual passam os jogadores até chegarem de iniciantes (ou “noobs”) a especialistas (“experts”). Esse caminho é percorrido em comunidades de gamers e se desenvolve nas seguintes fases: noob (exploração de controles e temas); jogador competente (conhecimentos básicos do jogo); jogador máster (conhecimento dos sistemas envolvidos no jogo); “Tinkerer” (conhecimento de variáveis, poder de modificar as regras do jogo); designer (criador de um jogo). Como fundamental, ainda, estão as interações sociais do jogador no espaço de afinidade do qual participa. Como podemos verificar nessa progressão evidenciada pelo autor, tornar-se um criador de jogos resulta na participação de um coletivo de gamers que partilham conhecimentos, habilidades e modos de invenção. Podemos pensar que resulta em um domínio cognitivo no sentido dado por Maturana e Varela (1995).

É importante concluir essa incursão no domínio de estudo dos videogames, ressaltando que os jogos não são neutros. Os videogames são mundos construídos,

mundos que são projetados de acordo com um conjunto particular de regras que permitem certos tipos de ação e não outras. De tal modo, são espaços ideológicos. Por isso, diferentes comunidades constroem diferentes modos de participar, cooperando, competindo, o que pode eventualmente estabilizar modos de ler a tecnologia. Desta forma, uma alfabetização no domínio dos games implica também analisar os aspectos ideológicos, de valorização de determinados tipos de experiências em relação a outras possíveis. Embora nosso trabalho não foque especificamente a dimensão política dos videogames, é importante salientar sua existência, principalmente no momento em que são utilizados como tecnologias de aprendizagem. Mesmo na produção de jogos (games), como é o caso dessa pesquisa, esse aspecto pode ser objeto de reflexão para aqueles que se põem a programar.

III. OBJETIVOS E PROBLEMA DE PESQUISA

Iniciamos esta pesquisa com algumas hipóteses e expectativas em relação às experiências possíveis dos sujeitos com o Etoys nas Oficinas de Construção de jogos. Tendo fundamento em décadas de pesquisa do LEC sobre as construções de sujeitos com linguagem Logo e sendo o Etoys um “herdeiro quase direto” daquela, inicialmente nossa proposta era buscar nas condutas cognitivas dos sujeitos aspectos psicogenéticos das noções envolvidas na atividade em Etoys, particularmente na concepção do objeto digital. Na época do projeto de dissertação partimos da hipótese que através das atividades nas Oficinas de Etoys poderíamos traçar como se dava o processo psicogenético de construção da noção de objeto digital e dos sistemas de relações postos em ação em um jogo.

O desdobramento das Oficinas nos forçaram a reconfigurar essas expectativas e reconsiderar a proposta inicial pelos motivos que passamos a elencar:

Alguns aspectos e circunstâncias da realização das oficinas, nos levaram a uma reorientação na forma de análise do material recolhido. Embora seja difícil a um pesquisador mudar o rumo da pesquisa em seu andamento, tal situação faz presente em uma pesquisa intervenção. Como comenta Kastrup (2008), uma pesquisa provoca efeitos nos participantes, no próprio pesquisador, na instituição onde se realiza e no próprio rumo da pesquisa. Alguns desses efeitos serão discutidos adiante, cabendo agora iniciar pelo deslocamento ocasionado pela própria questão de pesquisa.

Um primeiro ponto a considerar é a diferença das experiências que dão lugar a uma pesquisa psicogenética – na qual pretende-se acompanhar como algum conceito ou noção se constitui – com o desafio de construir jogos. Em uma pesquisa psicogenética, busca-se analisar como determinada noção ou conceito passa a fazer parte da estrutura cognitiva, de tal modo que o sujeito pode operar com ela e construir novos conhecimentos. Nas pesquisas anteriores (Fagundes, 1986, Maraschin e Nevado, 1992) construiu-se uma indicação que para se acompanhar um processo psicogenético haveria a necessidade de um tempo maior de experimentação com objetos que poderiam desafiar um sujeito a constituir tal noção. Esse número ficou estimado em torno de um mínimo de 20 sessões.

Em nossa experiência não foi possível chegar a um número tão elevado de sessões devido a várias razões. A principal dificuldade deveu-se a circunstâncias de cotidiano da escola: greves e o programa dos conteúdos a serem trabalhados em sala de

aula que impediram ou dificultaram os estudantes de participar de um número maior de oficinas. Da mesma forma, e isso se tornará mais claro adiante, o acompanhamento da atividade dos estudantes apenas pelo pesquisador, ainda que em grande parte filmado, não foi suficiente para o necessário detalhamento das atividades dos participantes a cada momento do processo. Mas o que pareciam ser somente restrições ou dificuldades começaram a evidenciar outras possibilidades de estudo.

Desse modo, optamos por acompanhar os movimentos da aprendizagem dos participantes que se empenhavam na construção de jogos, ao invés de seguir a psicogênese de uma noção específica. Levamos em consideração, como veremos, aspectos psicogenéticos de algumas noções de programação que puderam ser mais detalhadamente acompanhados nos processos de aprendizagem (em especial os TESTES – proposições condicionais), mas a análise será feita a partir de processos de aprendizagem que emergiram da experiência das oficinas, e que suscitaram problematizações.

Diante do exposto, nosso foco passou a ser a atenção aos movimentos cognitivos ou processos de aprendizagem que ocorreram durante as oficinas. Tal deslocamento operou uma reformulação na questão e nos objetivos de pesquisa se comparados ao projeto inicial.

3.1 Problema de Pesquisa

Para redesenhar o problema de pesquisa relacionamos proposições advindas do campo de estudo dos videogames, acopladas a uma dimensão fundamental: a da produção e programação do jogo. Assim, consideramos não só os mecanismos cognitivos envolvidos na atividade de produção de um jogo em uma linguagem computacional específica, o Etoys, como também contribuições sobre os mecanismos na atividade de jogar (ressaltando que, em nosso caso, o jogo desenvolvido foi também jogado nas oficinas pelos próprios criadores ou pelos colegas).

Partimos da proposição de Squire (2005) segundo a qual jogar videogame consiste em uma sofisticada prática que envolve o reconhecimento de sinais e padrões e privilegia um conhecimento operativo em vez de um conhecimento declarativo. Essa aprendizagem se processaria através da interação de quatro processos:

- (1) aprender a “ler” o jogo como um sistema semiótico, (2) aprender e coordenar a gama de movimentos possíveis e os seus efeitos, (3) entender as regras e suas

hierarquias e (4) monitorar de modo contínuo os objetivos e subobjetivos (SQUIRE, 2005, p. 8).

Nossa principal questão consiste em avaliar se as operações que o autor atribui à atividade de jogar também seriam válidas para pensar a atividade de programar um jogo? Além disso, quais conceitos de programação estariam envolvidos na programação de um jogo em Etoys? Como esses conceitos podem estar coordenados no contexto de um jogo?

Outra questão agregada à anterior, parece ser a de que os processos definidos pelo autor não contemplam de modo destacado o design e a narratividade no jogo. Certamente se sua questão se refere mais aos modos de jogar, o jogador se insere em um design e em uma narrativa já proposta pelo jogo (embora em muitos casos, o jogar influencie o desdobramento da narrativa). Em nosso caso, ao serem convidados a programar um jogo, existiria um quinto processo: a construção de uma narrativa que se acopla à jogabilidade expressa nos pontos anteriores. Outra questão decorrente dessa é como a função jogador se acopla à função de programador. Ou seja, para programar um jogo há a necessidade de supor que o mesmo será jogado por alguém.

Desta forma, nossa questão de pesquisa passa a ser a de mapear esses processos na atividade de programação de jogos dos sujeitos convidados a participar das oficinas.

3.2 Objetivos

O principal objetivo deste trabalho consiste na análise dos processos de aprendizagem envolvidos na atividade de programação de jogos com a linguagem Etoys.

Como objetivos específicos podemos citar:

- mapear os modos pelos quais os participantes das oficinas interagem com o Etoys como um sistema semiótico, dinâmico e complexo;
- acompanhar como coordenam regras e suas hierarquias; movimentos e seus efeitos;
- flagrar como compõem objetivos e subobjetivos no jogo;
- analisar as composições entre a jogabilidade e a narrativa na composição dos jogos.
- identificar possíveis estratégias predominantes no design e programação dos

jogos

- identificar os conceitos de programação envolvidos no processo de apropriação do Etoys, assim como eles se relacionam, ou não, no contexto do jogo.

IV. MÉTODO

O método delineado para esse estudo consiste em uma pesquisa-intervenção com a realização de oficinas de programação de jogos no Etoys. Passamos, então, em um primeiro momento, à contextualização da inserção do presente estudo no contexto da pesquisa maior. A seguir, faremos a descrição do campo de pesquisa, dos participantes e da caracterização das oficinas.

4.1 O Projeto UCA e o Projeto Província de São Pedro

As atividades que dão origem a esta pesquisa estão inseridas, como comentamos, em dois Projetos, em níveis federal e estadual. O primeiro, o Projeto UCA, teve origem em 2007, quando uma leva de laptops 1000 laptop XO foram doados pela ONG OLPC ao governo brasileiro. Em sua fase Pré-Piloto, o Projeto UCA teve cinco experiências em cinco estados brasileiros. No Rio Grande do Sul, a primeira experiência ocorreu em uma escola Estadual, onde cada aluno recebeu um laptop XO. O Laboratório de Estudos Cognitivos (LEC/UFRGS) foi convidado a coordenar a experiência, com projeto de pesquisa que incluía algumas investigações conduzidas por um grupo interdisciplinar de pesquisadores do LEC em diferentes áreas de conhecimento. Já naquele ano, uma das Unidades de Análise se constituiu em experiência na atividade de programação com o Etoys, um dos softwares disponíveis nos laptops XO, na qual me encontrava inserido.

A partir das cinco primeiras experiências no ano de 2010, o Projeto UCA ganhou maiores dimensões no cenário nacional, sendo incluído pela lei nº 12.249 como política pública e passando a se chamar PROUCA (Programa UCA). No mesmo ano, foi iniciada uma etapa Piloto do PROUCA, em que 300 escolas públicas receberam um laptop por aluno (em nível do Programa, os laptops entregues não eram os XO's, mas sim os fabricados pela empresa vencedora da licitação realizada pelo governo). A fase Piloto foi encerrada em 2012, quando foi realizado um registro de preços pelo Fundo Nacional do Desenvolvimento da Educação (FNDE) e aberto uma linha de financiamento pelo governo federal, através do Banco Nacional do Desenvolvimento (BNDES) para que cada estado ou município pudesse, através de iniciativas locais, oferecer laptops a suas redes de ensino.

Ao final do ano de 2010, o LEC foi novamente convidado pela OLPC e governo

federal a coordenar uma experiência de pesquisa para a doação de novos 500 xó's. Uma nova escola estadual, agora em parceria também com o Governo do Estado do Rio grande do Sul, foi escolhida como local de pesquisa interdisciplinar. Participamos, então, através do LEC, a partir de 2011, de um grupo de pesquisadores que acompanhou o processo de inserção daquela escola na modalidade 1:1. Como uma atividade voluntária, oferecemos oficinas de programação em Etoys para duas turmas de terceiro ano do ensino fundamental, acompanhando alunos e professoras, que aderiram voluntariamente às oficinas.

Foram realizadas oficinas com as turmas inteiras (aproximadamente 30 alunos), em sala de aula, em encontros semanais de aproximadamente uma hora e meia. A riqueza dos processos desencadeados pelas atividades nas oficinas gerou a publicação de alguns trabalhos (FAGUNDES, SPERB E SCHAFFER, 2011; LONGO, 2012), além da necessidade e a motivação para a sequência desta pesquisa.

Em 2013, no âmbito estadual, inicia o Projeto Província de São Pedro, idealizado pela Secretaria de Educação do Estado do Rio Grande do Sul. O projeto tem como objetivo a inclusão digital, a “modernização tecnológica da rede estadual e a apropriação da tecnologia por alunos e professores, principalmente através da utilização de dispositivos móveis” (SEDUC/RS). As escolas estaduais do Rio Grande do Sul, antes inseridas no Programa UCA são integradas ao Projeto Província de São Pedro, tendo disponibilizados um laptop para cada um de seus alunos e professores. Pelo fato desse programa entrar na pesquisa como campo empírico os mesmos não serão objeto de análise do ponto de vista de seus objetivos e políticas. Cabe apenas mencionar que o laptop disponibilizado pelo Programa Província de São Pedro é um terceiro modelo, que não o XO, do Pré-Piloto do PROUCA ou o Classmate do Piloto, mas um netbook Positivo. Tal fato já demonstra que a continuidade do esforço governamental de inclusão digital se dá com descontinuidade das tecnologias disponibilizadas.

4.2 A Escola, a turma e os participantes

A escola participante desta pesquisa é o Instituto de Educação General Flores da Cunha, situada em uma região central de Porto Alegre. Uma das principais características da escola é ter, em seu grupo discente, alunos de diversas regiões e variadas condições econômicas e sociais. Nessa escola funcionam tanto o Ensino Médio quanto o Fundamental, sendo que as turmas de alunos dos quatro primeiros anos do Ensino

Fundamental são atendidas em um outro prédio, conhecido como “Anexo” situado no mesmo bairro do prédio principal. Foi no “Anexo” do Instituto de Educação, portanto, que realizamos as primeiras oficinas, experiência que chamamos de Piloto da Pesquisa, no ano de 2011. Cabe salientar que o Anexo está inserido, desde aquele ano, integralmente na modalidade 1:1.

A turma de alunos do terceiro ano escolar que em 2011 participou das oficinas de aprendizagem em Etoys, em 2013 passou a ser atendida no prédio principal do Instituto, agora no quinto ano letivo. Os quintos anos são os únicos a seguir inseridos na modalidade 1:1 no prédio principal. Ao longo da pesquisa, portanto, tanto alunos como professores desse ano tinham em seu cotidiano de sala de aula o laptop à disposição. Para que pudéssemos acompanhar mais detalhadamente os processos de aprendizagem de cada um dos estudantes, decidimos levar grupos menores para o Laboratório de Informática da Escola. Além da escolha do laboratório, ao invés da sala de aula, também utilizamos um computador de mesa ao invés do laptop. Ambos fatores contribuíram para dar mais condições de acompanhamento do estudo.

Por questões circunstanciais de funcionamento da escola (disponibilidade do Laboratório de Informática) e turno de trabalho, optamos por incluir, na pesquisa, alunos de apenas uma das turmas que havíamos acompanhado na experiência de 2011.

Em 2013, nesse novo momento da pesquisa, consideramos alguns critérios para a escolha dos participantes. O primeiro deles é que tivessem participado das oficinas anteriores e apresentado diferentes modos de apropriação da programação do jogo. Como não dispúnhamos de muito tempo para as oficinas e as mesmas aconteciam no horário escolar, ficava difícil retirar os estudantes várias vezes ou por um período mais longo de tempo da sala de aula. Desta forma, o fato de já terem experiência com o Etoys tornou-se um importante critério de inclusão. O outro critério é que aceitassem nosso convite. Consideramos importante a participação voluntária nas oficinas, uma vez que a curiosidade e o interesse são indicativos de uma posição ativa de aprendizagem e de construção de conhecimento.

De posse desses critérios fomos encontrar a turma. A escolha dos participantes se deu em três momentos. Em um primeiro, em Maio de 2013, fizemos o convite aos estudantes em sua sala de aula. Naquele momento, um fato que chamou bastante nossa atenção foi a diferença entre o interesse de meninos e meninas: apenas duas disseram que gostariam de participar, enquanto quase todos os meninos “levantaram a mão”. Como forma de selecionar os meninos, já que muitos estavam interessados,

perguntamos à professora regente, em um segundo momento, quem ela considerava que poderia se beneficiar com a intervenção. Assim, foi necessário agregar esse novo critério, uma vez que somente conseguiríamos atender dois grupos de três alunos a cada uma das oficinas semanais. Em um terceiro momento, após a comunicação aos pais e à direção da escola, voltamos à turma e comunicamos aos participantes, para então fazer as combinações sobre o funcionamento das oficinas. Alguns alunos demonstraram frustração com o fato de não poder participar. Por sugestão da professora regente, fizemos a combinação de que os participantes das oficinas se disponibilizariam a convidar colegas interessados e compartilhar as atividades desenvolvidas na oficina.

Participaram das oficinas 06 estudantes, quatro meninos e duas meninas, do quinto ano do Ensino Fundamental, todos com idades entre 10 e 11 anos, sendo que os encontros eram realizados em trios.

Além das combinações com a professora regente e a direção da escola, também os pais foram consultados e consentiram sobre a participação de seus filhos nas atividades da pesquisa em horário de aula normal. O projeto teve aprovação do Comitê de Ética em Pesquisa do Instituto de Psicologia da UFRGS.

4.3 O Projeto Piloto

Tomamos como experiência introdutória deste estudo 20 oficinas realizadas ao longo do ano de 2011. A descrição dessa intervenção, que faremos aqui, certamente não é suficiente para demonstrar a riqueza dos processos ocorridos nos encontros. O intuito, aqui, é o de apontar como essa experiência ajudou tanto a motivar quanto a delimitar as questões do presente trabalho.

As oficinas foram realizadas semanalmente, para as turmas inteiras (cerca de 30 alunos cada) e suas professoras. Tinham duração de uma hora e meia, com a presença de dois pesquisadores. Como objetivos, tínhamos a familiarização dos alunos e das professoras com o ambiente Etoys, através de atividades que foram se definindo conforme as próprias produções e o interesse dos envolvidos.

Os encontros foram realizados em forma de desafios, geralmente pela apresentação inicial de uma situação-problema no Etoys, para que então se fizesse a proposta de atividade. Eram estimuladas a livre criação e livre interação entre os alunos, alunos e professora, e alunos e pesquisadores. Após uma apresentação inicial, os estudantes trabalhavam individualmente ou se reuniam em grupos para criar projetos de

seu interesse.

Uma esquematização geral simplificada das experiências nas oficinas pode ser exposta da seguinte forma:

Exploração dos recursos do Etoys	Interações aluno-aluno aluno-pesquisador aluno-professor
Escolha do Projeto (individual ou em grupo) (história, animação, jogo, etc)	
Desenvolvimento do Projeto (individual ou em grupo)	

Quadro 1. Organização das atividades nas Oficinas do Projeto Piloto

Ao final daquele ano letivo, quando recolhidos os laptops dos alunos, foram coletados os arquivos produzidos em Etoys de cada computador. Tivemos, assim, acesso a cada um dos projetos desenvolvido pelos estudantes em sua versão final. Na primeira seção dos resultados, apresentaremos uma síntese da análise dos dados obtidos nesta etapa.

4.4 As Oficinas

Em 2013, foram realizadas 14 oficinas com cada grupo no laboratório de informática da escola, com periodicidade semanal e duração de aproximadamente uma hora e trinta minutos cada. Os alunos eram chamados por mim em sua sala de aula, para então irmos ao laboratório. Conforme previsto, dividimos os participantes em dois grupos (três em cada grupo), sendo que havia um dia para cada grupo durante a semana. Nas oficinas, cada estudante dispunha de um computador de mesa para trabalhar. Aos participantes também foi oferecida a possibilidade de trazerem seus laptops educacionais, o que, no entanto, aconteceu somente nos últimos encontros. A pedido dos estudantes, no decorrer das atividades, também foram fornecidos pendrives para que pudessem gravar, transportar e desenvolver seus projetos em momentos fora da oficina.

As atividades das oficinas foram registradas em: filmagem, cópia dos arquivos das versões dos jogos desenvolvidos, registros no diário de campo do pesquisador.

De uma forma geral, a estratégia utilizada em cada oficina é mostrada no quadro abaixo:

Proposta de Desafios (situação-problema, com material de suporte)	Interações
Exploração dos recursos do Etoys	estudante-estudante
Escolha do Jogo (individual ou em grupo)	estudante (s) - pesquisador
Desenvolvimento do Jogo (individual ou em grupo)	

Quadro 2. Organização das atividades nas Oficinas de Construção de Jogos

Buscamos ir adequando a estratégia de cada oficina conforme o interesse dos participantes, quando esse fosse manifesto, ou para despertá-lo, quando não manifestado. Por exemplo, na primeira oficina propomos uma livre exploração do Etoys como intuito de fazer o que “lembravam” do programa. Nessas primeiras explorações, os participantes, em sua totalidade, buscaram recriar projetos que haviam feito anteriormente. Quase todos eles, então, tentaram reproduzir jogos que desenvolveram, os quais envolviam os atributos de posição (pares ordenados “x” e “y”) dos objetos na tela, além de tentativas de criação de eventos com o uso de proposições condicionais.

Como exemplo de um desses jogos, Gui (10 anos), tenta reproduzir um projeto realizado na etapa anterior da pesquisa. Nesta construção inicial do aluno (compartilhada com um colega), diversos objetos são criados em um background de jogo, chamado de o “jogo mais difícil do mundo” (sic). Um objeto, personagem a ser controlado pelo usuário, tenta atravessar uma região habitada por outros objetos, que se constituem como “obstáculos” ao seu movimento. Gui deseja criar um evento que programe os objetos de forma que quando o personagem controlado pelo usuário “toca” em um obstáculo, ele volta à sua posição de início. Essa condição de “tocar”, na linguagem do programa, é posta pela construção de um TESTE. Embora o estudante não consiga reproduzir, de memória, as relações necessárias à lógica do TESTE sua intenção é clara (de fato, o sujeito descreve sua intenção como de criar o “modo bater e voltar”).

Na segunda oficina, propusemos aos participantes uma espécie de jogo

desenvolvido no próprio Etoys em que a brincadeira/desafio era encontrar um objeto pelas suas coordenadas x e y . Antes de explorar o desafio no computador, no entanto, fizemos uma brincadeira no quadro branco da sala, em que, desenhando em um pequeno plano cartesiano, como em uma “batalha naval”, os estudantes tinham de dar pares posições uns aos outros para que adivinhassem posições (figura abaixo).

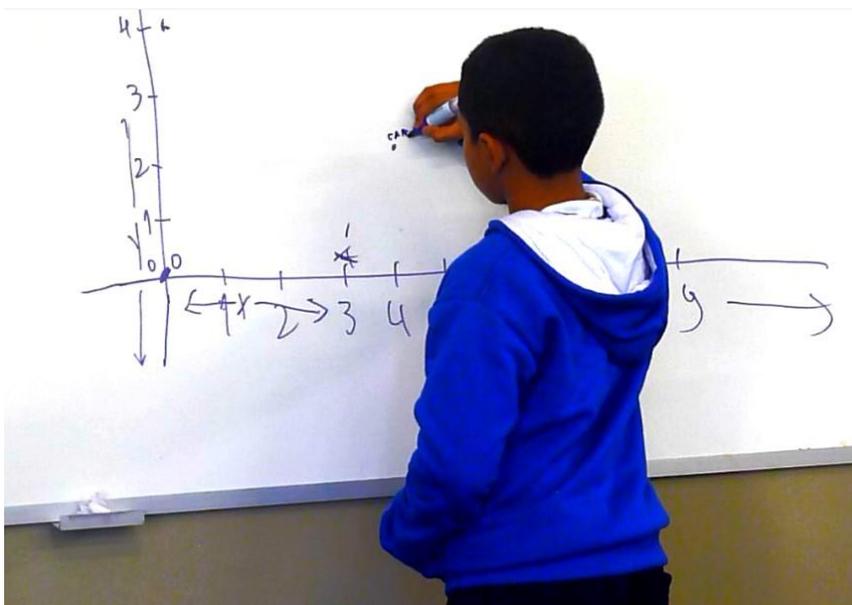


Figura 8. Brincadeira sobre plano cartesiano no quadro

O propósito do jogo, que denominamos de “Onde está o Wally”, era o de desafiar os sujeitos ao uso das coordenadas x e y para encontrar o objeto, clicando em um botão que gera uma posição aleatória pela tela. Além do sorteio da posição com o “Wally” visível no Mundo, há também a opção de fazer o “Wally” ficar invisível, tentando então adivinhar sua posição apenas pelas coordenadas.

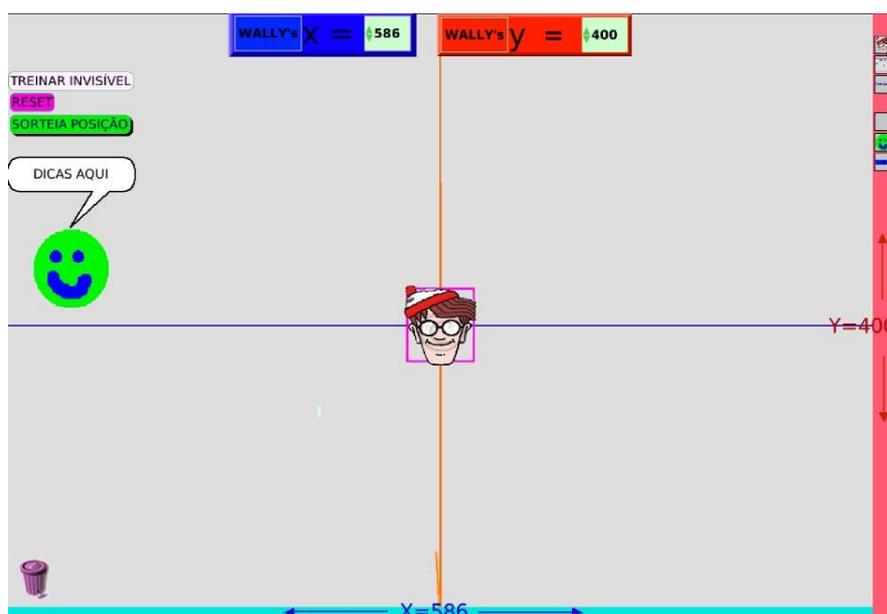


Figura 9. Jogo "Onde está Wally"

Um dos objetivos com esse jogo era proporcionar a interação com as regras do programa que determinam as posições dos objetos pela tela. A partir dessa interação observamos as reações e hipóteses dos participantes a respeito do funcionamento dos códigos postos em ação no Etoys. Assim, foi possível identificar que a interação com o "Wally" gerou uma brincadeira, entre os participantes, de fazê-lo desaparecer da tela, incrementando os valores de X ou Y. Em ambos os grupos, os sujeitos faziam o objeto mudar de lugar para então solicitar ao colega "botar o mesmo Y ou botar o mesmo X que o meu", por exemplo.

Além disso, tínhamos como objetivo apresentar de um modo lúdico e o menos diretivo possível, alguns recursos para a construção de um jogo simples, como por exemplo, as possibilidades de movimentar um objeto não só pelo arraste, mas alterando o número de suas coordenadas; a criação de botões para disparar programas criados, a possibilidade de importação de imagens para o Etoys.

Propusemos após, como sugestão de desafio, que produzissem algo com o que haviam aprendido jogando com o Wally. A partir dessa sugestão, algumas ideias surgiram. Por exemplo, Joa G. criou o objeto "louva-a-deus" que se encontrava camuflado em um background propondo que o desafio seria "achar o louva-a-deus pelo x e pelo y". Liv. cria um objeto "formiga" a ser controlado pelo usuário, pelas coordenadas x e y, para chegar de uma posição inicial até outra (ambas produções serão tratadas adiante).

Dan edita o objeto "Wally", usando as propriedades de desenho e o chama de "Walking Dead". Sua estratégia é bastante econômica, o objeto é somente editado

graficamente, conservando todas as regras e scripts como no jogo original. Com essa simples mudança, no entanto, Dan. renova o interesse pelo jogo, ficando por longos minutos jogando, diverte-se e ri do nome que deu a sua própria criação.

Na **terceira oficina**, os participantes voltam a explorar o Etoys livremente, podendo interagir entre si e propor quaisquer ideias que tivessem para um jogo a ser desenvolvido. Um movimento que chamou atenção nesse encontro, semelhante a outros estudantes, foi o de recuperar (como havia feito Dan. ao final do encontro anterior) o jogo “Onde está o Wally” e editá-lo segundo seus próprios desenhos. Os estudantes não alteram a programação subjacente ao jogo, mas demonstraram prazer em interagir com os objetos que haviam recriado. Essa é uma modalidade interessante de produzir um jogo utilizando a programação de um jogo anterior. A programação parece funcionar como as engrenagens mencionadas por Papert.

Liv, quando desafiada a pensar em um possível jogo, prontamente responde que “não tem ideias”, mas acaba por criar um início de “jogo de estacionamento” (sic). Rapidamente, com a ajuda dos colegas passa a diferenciar edição de objetos e mundo, chegando ao design de um jogo onde um carro teria que ser estacionado em uma “garagem”. Chama atenção também que ela, desde o início, manifesta a intenção de criar “fases”. Essa intenção de Liv. é também manifesta por Joa S (na primeira sessão), quanto deseja criar “diferentes níveis” para seu jogo (no caso, “o jogo mais difícil do mundo”). Tal intenção, dada a estrutura do Etoys não é algo simples de fazer, como veremos.

Ao final desse encontro, Dan e Liv realizam uma brincadeira de correr pela sala, imaginando ser os objetos que haviam criado na tela do computador.

Uma questão que emerge dessas primeiras oficinas que é sobre a própria concepção de “jogo”. Gab, por exemplo, no seu segundo encontro, inventa o que chama de “jogo de achar a estrela”. Editando o “Mundo”, ela desenha muitas estrelas, todas idênticas e as distribui pela tela. Como são desenhos decalcados no background, ou “Mundo”, não é possível arrastá-los como um objeto digital. Gab então cria um objeto “estrela”, idêntico às anteriores e propõe que seu jogo está no desafio de “achar a estrela que se move com o mouse”. Aqui vemos que o breakdown advindo da não diferenciação entre desenho e objeto digital inicial leva a uma resolução incorporando a um jogo. Ou seja, aproveita a indiferenciação inicial para convertê-la em um jogo. É como se dissesse ao jogador algo do tipo: vamos ver se consegues diferenciar o que é um desenho (das muitas estrelas) de um objeto (a estrela específica). O mesmo poderia ser pensado para Joa S e seu “louva-a-deus”.

As oficinas realizadas iam dando pistas de como disponibilizar materiais, desafios e informações para as oficinas subsequentes. Deste modo, fomos organizando materiais referentes a um ou outro recurso específico do Etoys que, supúnhamos, seria interessante para a composição de um jogo. Trabalhamos aqui com a ideia de conteúdos sob demanda (GEE, 2004). Como exemplo desses materiais, em um **quarto encontro** levamos dois tutoriais como sugestão para os estudantes. Um dos tutoriais se constituía em uma espécie de “passo a passo” com o programa e recursos para construir uma animação por quadros, um desenho animado. O outro material mostrava a composição de uma “perseguição” no Etoys (a construção de dois objetos e os scripts que os relacionavam de forma a um se movimentar na direção do outro). Iniciando a oficina e disponibilizando os materiais, tentávamos novamente seguir as interações dos estudantes frente ao jogo que desenvolviam (se desenvolviam o iniciado, ou se começavam outra atividade) e aos recursos disponibilizados.

Como um exemplo dos movimentos emergentes nesse quarto encontro podemos citar Liv. que, após momento de relutância inicial, chega à proposta de reproduzir um jogo que “seu irmão jogava”. Tratava-se de um jogo “desses de internet”, em que “tem um sorvete lá que fica fugindo e os monstros que o perseguem, em tipo um labirinto”. Havia lembrado, então, de um jogo online que conhecia e passou a tentar reproduzi-lo, criando e editando objetos. O desenvolvimento dessa proposição na seção posterior.

Joa G e Joa S, por outro lado, assim que tiveram acesso aos tutoriais passaram a tentar programar reproduzindo o que estava no tutorial: não iniciaram o desenho de um jogo ou desenvolveram um iniciado, mas agiram sobre recursos disponíveis.

A descrição acima pretendeu dar uma ideia do funcionamento das oficinas e a possibilidade de flagrar movimentos de aprendizagem. Não seguiremos apresentando as demais oficinas do mesmo modo, mas para fechar essa seção, mencionamos duas oficinas que foram realizadas com os dois trios reunidos, a fim de apresentarem e compartilharem os jogos que estavam desenvolvendo, ou que haviam desenvolvido em separado.

Nesses encontros houve a interação com pessoas fora da escola. No primeiro, foram entrevistados por pesquisadores sobre suas produções, enquanto nelas trabalhavam. O segundo, consistiu em uma oficina na qual participaram da criação de jogos com alunos de outras escolas participantes do Programa UCA⁵. Nesse encontro,

⁵ o evento foi o encerramento oficial da etapa piloto do PROUCA, realizado no colégio de Aplicação da UFRGS (CAP/UFRGS). No encontro, estiveram presentes alunos e professores de seis escolas do Rio Grande do Sul, além de uma escola do projeto Ceibal, do Uruguai.

os participantes foram convidados, em um primeiro momento, a apresentar suas produções no Etoys para outros alunos e em um segundo momento a participar desenvolver jogos com os demais estudantes.

Faz-se necessário acrescentar, para completar a descrição do método, que duas entrevistas individuais foram realizadas com cada um dos participantes das oficinas no intuito de investigar como estava se dando o processo de apropriação da linguagem e de desenvolvimento dos seus jogos. As entrevistas foram realizadas aproximadamente na metade e ao final do andamento das oficinas.

Sabendo ser impossível dar conta, em nossa análise, de toda a riqueza dos processos de cada sujeito nas oficinas, optamos pelo mapeamento de alguns dos movimentos da aprendizagem como definidos na apresentação do problema da pesquisa. A análise será feita a partir da descrição de movimentos específicos que emergiram da experiência das oficinas discutida com conceitos advindos da revisão de literatura. Trata-se então de uma análise descritivo-crítica, na qual o elemento descritivo assume a função de operador da análise.

V. APRENDENDO A PROGRAMAR UM JOGO EM ETOYS

5.1 O Projeto Piloto

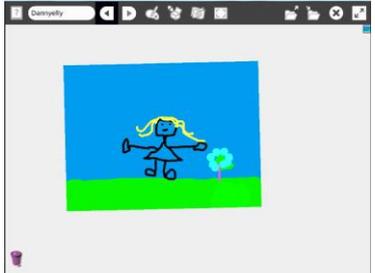
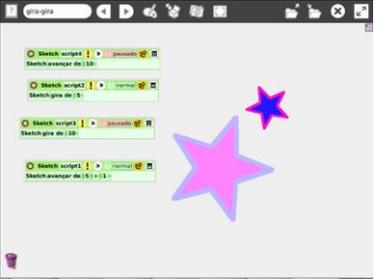
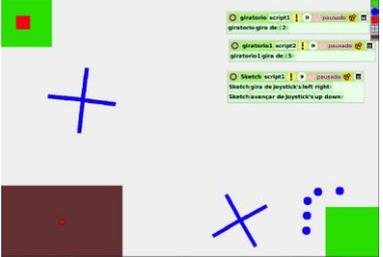
Tendo descrito o formato das oficinas do experimento piloto na sessão anterior, passamos aqui a apresentar uma análise dos modos de apropriação da linguagem de programação a partir dos projetos de jogos desenvolvidos pelos estudantes. Como acima salientado, tivemos acesso a cada um dos projetos desenvolvido pelos estudantes em sua versão final.

Através da análise das produções realizadas em 2011 por cerca dos 50 alunos do terceiro ano do ensino fundamental, foi possível chegar a caracterização de diferentes modos de apropriação do ambiente Etoys, conforme a Quadro nº 3.

Para diferenciar os projetos em modos utilizamos como critério agrupar projetos com semelhantes estratégias de produção, conforme explicitamos em cada um dos modos.

Modo I	Modo II	Modo III
<p>As trocas da criança com o ambiente centram-se sobre um nível gráfico dos objetos. Frequentemente há a indiferenciação entre o que é o “mundo”, ou o fundo, e o objeto, o que estará nele contido, ou mesmo entre os próprios objetos. A ideia de que “tudo é objeto”, está ainda distante do sujeito. Ainda não há, assim, qualquer comunicação “objetiva” entre o sujeito e o objeto que ele construiu, isto é, não há scripts que comandem o objeto. Há produção de símbolos que guardam uma ligação subjetiva com suas ideias, mas ainda não há construções que envolvam a linguagem de programação. Também ainda não há uma intenção (ao menos manifesta, no produto criado) da criança com relação aquilo que quer fazer, que deseja fazer (“projeto”). Podemos dizer que, se existe uma narrativa e uma jogabilidade, essas estão amalgamada em uma figura ainda estática.</p>	<p>Há o início da compreensão da possibilidade de criar diferentes objetos, e diferenciá-los do mundo. Observa-se, também, uma intenção inicial da criança em atribuir algum comportamento ao objeto, ou de dar-lhe “comandos”, mas essa ideia não é necessariamente representada pela programação através de comandos. Esses comandos atribuídos ao objeto, quando existem, são na maioria das vezes aleatórios, sendo que ainda não há relação entre o comportamento de diferentes objetos.</p>	<p>Há objetos criados separadamente, com clara diferenciação do mundo. Há uma intencionalidade da criança em determinar comportamentos a esse objeto, que pode ser constatada objetivamente, por comandos (em scripts) dados a esse (s) objeto (s). Há, também, início de relações entre objetos diferentes (comandos ou propriedades que se relacionam, como por exemplo um joystick que controla um personagem, etc). Essas relações, em alguns casos, já contém proposições condicionais (se...então), que: ou relacionam diferentes objetos ou determinam o comportamento de um objeto através de uma condição a ser ou não satisfeita.</p>

Quadro 3. Modos de Apropriação do Etoys a partir dos projetos desenvolvidos no Projeto Piloto

Modo I	Modo II	Modo III
<p data-bbox="244 297 619 517">“criança brincando em um jardim” - um cenário é criado, com vários elementos diferentes, como um único “objeto” estático.</p> 	<p data-bbox="643 297 1018 427">“gira-gira” - duas estrelas (um único objeto), com vários scripts criados aleatoriamente</p> 	<p data-bbox="1042 297 1425 887">“o jogo mais difícil do mundo” - objetos diferentes (“giratórios”, “quadrado vermelho”, “joystick”, “pontos de chegada” e “partida”), com comandos próprios de acordo com sua função no “jogo”. Há relação entre objetos (joystick e “quadrado vermelho”) e a clara intenção de um jogo: a chegada de um personagem (quadrado vermelho) a um objetivo, passando por obstáculos</p> 

Quadro 4. Exemplos de produção por Modo de Apropriação

Inferimos, conforme tentamos mostrar nos exemplos, que houve diferentes modos de apropriação do Etoys como um sistema semiótico que implicam uma compreensão distinta. No Modo I pode existir diferenciação entre partes e todo na expressão falada, mas ela não alcança tradução em termos da linguagem digital. Podemos supor que a narrativa “a criança está brincando no jardim” não encontra necessidade de ser expressa em uma jogabilidade. A narrativa parece bastar-se por si mesma. Em muitos outros casos os participantes somente se dão conta que estão diante de um objeto único (embora composto por diferentes desenhos) quanto tentam movimentar um dos desenhos. Aqui pode ocorrer um breakdown no sentido de considerar que o que define um objeto digital não é somente seu desenho, mas também a programação implícita no mesmo. Programação e desenho dão existência a um objeto digital.

Na produção caracterizada como exemplo de apropriação do Modo II, vemos já uma distinção em relação ao que caracteriza um objeto digital. Os objetos aparecem

separados do fundo (background, ou “Mundo” - que também é um objeto, embora ainda não concebido como tal) e com uma operatividade própria dada pelos comandos; mas os scripts são criados ainda aleatoriamente: na produção acima, o “Gira-Gira”, por exemplo, vários scripts dão o mesmo comando ao objeto. Podemos pensar que já existe uma primeira apropriação semântica, ou seja, que um objeto é um desenho e uma operatividade, mas a sintaxe que define essa função é ainda parcial ou globalmente atribuída.

Já no que definimos como Modo III de apropriação, os objetos são criados separadamente e com funções específicas que os colocam em um contexto operativo, de jogo. Há também relações entre esses objetos: podemos dizer que a jogabilidade e a narrativa existem já incorporadas a linguagem e passíveis de serem acionadas pelo programa. Progressivamente existe uma coordenação entre narratividade-design e operatividade. Certamente ainda os níveis de complexidade não são muito grandes, mas podemos dizer que existe uma fluência digital inicial.

Essa primeira mostra de resultados, embora limitada, uma vez que não foi possível acompanhar o processo de produção dos participantes, já revela que uma alfabetização digital apresenta dificuldades. Os diferentes modos de apropriação expostos através da análise dos projetos mostram uma crescente diferenciação na apropriação do Etoys como um sistema que integra objetos digitais e os relaciona em um “background” operativo. Através dos modos apresentados, fica claro que há movimentos de aprendizagem da linguagem do Etoys, embora ainda sem a riqueza da análise do processo. Assim, essas primeiras experiências nos motivaram a seguir questionando sobre os movimentos da aprendizagem, os quais distinguimos nas operações destacadas na formulação do problema da pesquisa.

No percurso da experiência piloto, outro aspecto a ser destacado foi o interesse dos estudantes na construção de jogos, que se fez presente desde o princípio. Durante um dos encontros, quando já tinham alguma familiaridade com os recursos disponíveis no Etoys, um estudante teve a ideia de reproduzir um jogo online⁶ que muitos alunos jogavam. A partir dessa primeira iniciativa, muitos colegas o seguiram tentando incluir a produção de jogos em seus projetos no Etoys.

A atividade de criação de jogos parece ter dado novos significados à própria atividade de programação, possibilitando, inicialmente, algumas diferenciações no uso dos recursos, que apareceram integrados em um espaço de jogo. Tal resultado, aliado à

⁶“O jogo mais difícil do mundo”, disponível em <http://www.o-jogo-mais-dificil-do-mundo.com>

discussão teórica anteriormente feita, nos estimularam a seguir investindo na programação de jogos como um interessante dispositivo de estudos dos movimentos e processos de aprendizagem.

5.2 Aprendizagem e Programação de Jogos

Passamos agora a discussão dos movimentos e processos de aprendizagem flagrados nas oficinas desenvolvidas em 2013. Para mapear tais movimentos havíamos definido quando da apresentação do problema e dos objetivos alguns indicadores de análise: 1) movimentos em direção à alfabetização digital em Etoys; 2) coordenação dos movimentos entre os objetos e seus efeitos; 3) organização das regras e suas hierarquias; 4) monitoramento dos objetivos e subobjetivos e 5) regulação entre jogabilidade e narrativa.

Os tópicos de análise acima arrolados são discutidos a partir de episódios coletados em diferentes oficinas que procuram mapear esses movimentos da aprendizagem. Ressaltamos que na discussão não tomaremos esses cinco indicadores separadamente, uma vez que um projeto de jogos implica um conjunto de ações articuladas. Tomemos, por exemplo, o caso de Joa G., que desenvolve um jogo que chama de “perseguição”, onde dois carros competem em um cenário de corrida. No processo de produção desse jogo pode-se destacar a estratégia usada pelo estudante para criar o design e a narrativa de seu projeto, mas também existe a apropriação dos conceitos de programação e como eles se relacionam ou não no jogo, a composição das regras de funcionamento do próprio jogo, etc. Assim, um mesmo projeto pode incluir mais de um dos indicadores acima mencionados.

Apesar dessa ressalva, organizamos essa discussão em quatro tópicos que passamos a descrever:

5.2.1 Primeiros passos com o Etoys

As primeiras análises advindas do projeto piloto já trouxeram evidências de que o processo de apropriação do Etoys caracteriza-se como uma alfabetização. Estabelecer o movimento dessa alfabetização não é uma tarefa fácil, pois ainda praticamente não há, na literatura, pesquisas que tenham tomado como questão tal processo nessa linguagem

de programação. Se considerarmos a lista que toma experiências de programação em Etoys como objeto de discussão (<http://forum.world.st/Why-is-Scratch-more-popular-than-Etoys-td3787377.html>) existem muitas controvérsias e relatos de dificuldades do uso dessa linguagem principalmente por crianças e adolescentes. Com o intuito de mapear movimentos em direção a essa alfabetização retomaremos algumas características peculiares do Etoys como sistema e como linguagem, para mostrar e interpretar exemplos de como os estudantes compreenderam, e em que nível, sua organização.

O Etoys é uma linguagem visual de programação onde o elemento icônico é predominante e condensa em si diversos comandos de programação. Se examinarmos mais detalhadamente essa característica, veremos que – ao mesmo tempo em que possibilita ao usuário alcançar resultados rápidos e com poucas ações -, dificulta a construção passo a passo de cada etapa de um programa. Tomamos como exemplo a criação de um script básico, comparando Logo e o Etoys: se desejamos criar um programa que ponha um objeto em um looping de determinado movimento: um quadrado, por exemplo. Na linguagem LOGO teríamos que escrever um procedimento denominado de QUADRADO, intercalando comandos de movimento do objeto para desenhar um lado do quadrado e o giro de 90 graus, fazendo-o repetir 4 vezes para completar a figura. No Etoys, basta o arrasto do comando “avançar de ___” para o mundo e cria-se um script que, para um usuário parece funcionar de modo automático uma vez que a programação não se encontra visível comando a comando. Teremos – também de forma “automática” - um “looping” de oito vezes por segundo de execução. Para obter o mesmo quadrado e o objeto repetindo o movimento, basta trazer o comando de giro 90 para dentro do script.

Com essa comparação gostaríamos de marcar uma característica importante do Etoys: a programação (ações, códigos, operações) é “incorporada” em um bloco (“tile”) de programação e que não visível de modo direto ao usuário. Essa característica diminui as possibilidades tanto de construção comando a comando como de acompanhamento da *gênese* de cada uma das noções de programação. Apesar disso, aumentam as possibilidades de criação de um conjunto de ações relativamente complexo. O exemplo a seguir detalha a possibilidade de um planejamento complexo, mesmo quando não exista um domínio comando a comando.

Na quarta Oficina, Dan. (10 anos) tem a ideia de fazer “uma corrida entre dois ou três carros” (que chama de “velozes e furiosos”). Após criar os carros, suas tentativas passam a ser de coordenar os movimentos de cada um deles para que a corrida ficasse “parelha” (ou seja, para que os carros chegassem em tempos parecidos, simulando uma

competição). Assim, experimenta diferentes valores com os comandos avançar de ___. Noto que Dan tem alguma dificuldade ao experimentar as “velocidades”, pois cada vez que clicava no “disparar”, os objetos se movimentavam em uma velocidade tal que lhe trazia problemas para a visualização do próprio movimento. Pergunto-lhe, então, o que fazia aquele ícone do “reloginho” que apertava?

D- Sai correndo! (o objeto)

B- Ah, sim! Porque ele sai correndo, que o computador tá dizendo?

D- (...)

B- esse relóginho o comando muitas vezes, fica repetindo, repetindo, sabe quantas vezes ele repete?

Abro a opção de ticks para mostrar a dan.

B- quantas vezes que ele repete? Tá escrito ali...

D- Ah, não sabia dessa aí!

D- 100 ou outro..

B- 8 vezes por...

D- 8 segundos!

B- 8 vezes por segundo!

Abaixo, apresentamos a figura da atividade que originou o diálogo exposto.



Figura 10. Uso de looping no Etoys no Projeto “Velozes e Furiosos” (Dan)

O projeto de Dan. pode ser discutido sob diversos aspectos, mas nosso interesse é pensar como a incorporação, ou a condensação de funções nos ícones da linguagem Etoys são interpretadas pelos participantes. No caso em questão, o ícone é uma espécie de relógio, que determina o estado do script (pausado, em espera, rodando). Se acionado, inicia um looping em que realiza o conjunto dos comandos programados oito vezes por segundo. O estudante não tem o acesso direto à programação desse looping: tudo se passa como se tivesse que fazer o caminho inverso e deduzir a operação do ícone após sua execução através da leitura do script. Cabe salientar, nesse sentido, que foi o diálogo

iniciado pelo pesquisador que tornou a função do ícone um “problema” para Dan. Uma nova alfabetização mostra a necessidade de coordenar ações não somente com a linguagem, mas com pessoas que participam desse domínio semiótico e que possam, portanto, convidar a agir nesse domínio, compartilhando seus códigos e regras, sua sintaxe e sua semântica.

Como mostraram as pesquisas em LOGO (FAGUNDES, 1986; MARASCHIN, 1989; MARASCHIN E NEVADO, 1992), um caminho importante percorrido pelos sujeitos para o domínio do sistema estava na atribuição de significados aos comandos dados à tartaruga, produzidos na leitura da programação feita. Assim, por exemplo, o sujeito poderia constatar que, dependendo do parâmetro numérico atribuído ao comando de deslocamento da tartaruga, a quantidade percorrida pelo objeto se modificaria, deixando diferentes comprimentos de traço na tela. Os comandos ou programas, quando criados, ficavam registrados e se transformavam em objeto de análise para o sujeito que estava programando. Constituíam-se em uma reflexão sobre ação que transcorria comando a comando. A programação em Etoys provoca outros modos de alfabetização. O que aqui importa não é tanto uma leitura comando a comando de tudo o que foi executado, mas produzir relações parte-todo, parte-parte e controlar seus efeitos. Blocos de programação são fornecidos ao programador que os utiliza como se fossem módulos que podem se compor com outros módulos. Por isso dizemos que o Etoys, diferentemente do Logo, modifica a alfabetização, “encurtando o caminho” de acesso a sistemas complexos, ao possibilitar o programador trabalhar com conjuntos de procedimentos não necessitando programar comando a comando.

Essa característica do Etoys trouxe desafios constantes para os participantes iniciantes: a necessidade de lidar com muitas noções ao mesmo tempo e de relacioná-las. Durante as oficinas, tivemos muitos exemplos nos quais os sujeitos tiveram dificuldades em atribuir significações àquela ação que se desenrolava na tela, embora pudessem relacionar um conjunto de ícones entre si. Tal característica abriu a possibilidade de acompanhar a aprendizagem de objetos complexos considerando a relação entre seus elementos muito mais do que acompanhar o desenvolvimento de noções básicas, passo a passo.

Outro ponto da linguagem a ser destacado é a distribuição dos scripts pela tela, quando da composição da programação de diversos objetos. Em muitas situações essa especificidade da interface do Etoys parece ter trazido desafios aos participantes. Nesse caso, houve a necessidade de um esforço “extra”, por parte do sujeito, no processo de

apropriação, no sentido de ir além da própria linguagem e sua sintaxe. O sujeito, mesmo atribuindo significado aos comandos da linguagem e tendo objetivos específicos naquele programa, teve de ir além, e lidar com arbitrariedades que fogem à linguagem e trazem dificuldades. A figura abaixo exemplifica uma situação em que essas dificuldades aparecem:

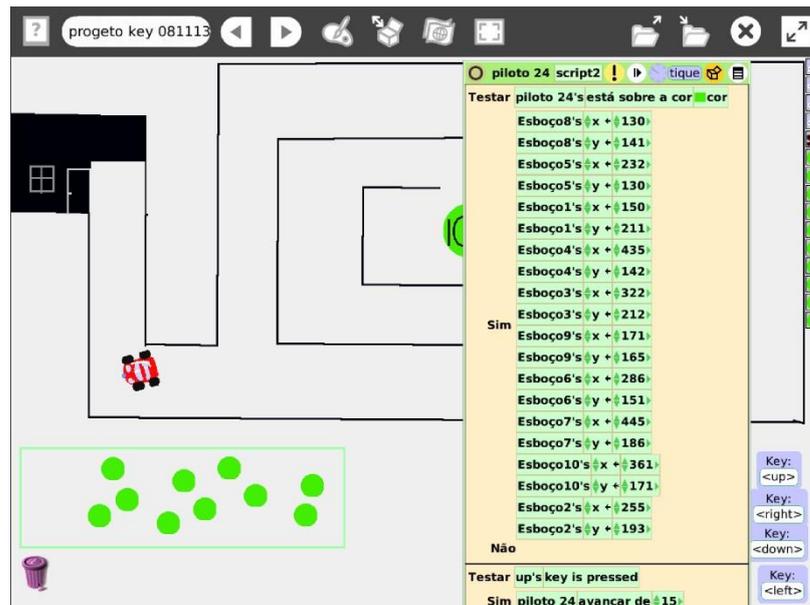


Figura 11. Dificuldades na organização dos scripts

No jogo acima, criado por Dan, há relação entre diversos objetos (o carro, o controle do carro, que é programado para ser feito com as setas do teclado, etc.). Nos centraremos aqui, no entanto, em um aspecto que ilustra o desafio descrito acima. Uma das intenções de Dan, em seu jogo, é fazer com que o jogador (pelo controle do objeto carro) coleccione as “moedas” (que aparecem como círculos verdes na tela) enquanto realiza a corrida.

Para tanto, Dan cria um teste que controla a condição de encontro do carro com a moeda (ou se um “vê” a cor de outro). Dentre as várias escolhas possíveis, a opção de Dan é por criar dentro de um único script diversos testes, tanto o de controle de posição das moedas, quanto o de controle do carro pelas teclas (na tela, podemos ver a determinação de cada uma das moedas, que nomeia “bolinhas”, e logo abaixo o início de controle do carro, que não pode ser contido na tela). Nesse momento, resulta em grande esforço para Dan coordenar as diversas noções envolvidas na programação com a necessidade de, ainda, fazer com que os scripts de comandos caibam e fiquem visíveis

na tela na área limitada para a escrita.

No exemplo mostrado, podemos observar a criação de um jogo, a relação entre muitos objetos, e ainda a criação de novas relações por Dan (não só um carrinho comandado por controle, mas um carrinho que coleta moedas, etc.). Consideramos que este processo mostra com precisão algumas das características da linguagem que vimos discutindo: se, por um lado, possibilita a criação livre e com autoria e a composição de relações entre objetos, por outro, o propósito de ser o mais aberto possível traz consigo dificuldades ao sujeito que escapam a própria linguagem, ou às noções de programação envolvidas.

Os exemplos anteriores evidenciam como a linguagem desafia a realizar uma composição entre muitos comandos em um único script, ou a organização de muitos scripts no monitor. Tal condição demonstra a necessidade de atenção para a organização dessas composições. Essa organização inicialmente não obedece necessariamente a uma regra única. Podemos dizer que, de modo geral, os iniciantes produzem relações entre os scripts a partir de uma organização “motora” ou “perceptiva”, porque, por exemplo, os scripts, de fato, “não vão mesmo caber na tela”, como expressou Dan.

Ainda, os casos nos quais o mecanismo do jogo projetado pelo sujeito (com intenção e necessidade por vezes claras) simplesmente não eram intuitivos: além do caso do “reloginho”, temos as próprias categorias de muitos comandos, muitas possibilidades ao mesmo tempo separadas apenas pela gramática, sem uma outra distinção mais intuitiva (como, por exemplo, por formas ou cores). Toda essa organização necessária foi dificilmente compreendida pelos sujeitos desde os momentos iniciais, na retomada dos projetos, até o final das oficinas. Por exemplo, o fato da “atrapalhação” de muitos scripts juntos, ou muitos comandos dentro de um mesmo script pode gerar uma necessidade de organização “lógica”, de redefinição desses scripts com menos comandos e de, por exemplo, uma nomeação clara que os diferencie.

A necessidade de organizar scripts desafia o planejamento e a constante organização do ambiente em que trabalha. Isso só pode ser atingido com um grau de interesse e de prática de programação muito grande. A partir desses exemplos, muito frequentes nas interações observadas, podemos nos questionar sobre o quanto o Etoys, pelas características apontadas, como que torna difícil a imersão necessária no caminho para a criação e execução de novas ideias dos iniciantes.

As produções e as atividades de cada participante com o Etoys, durante a construção de seus jogos, evidenciam diferenças em suas velocidades de apropriação.

Os participantes que apresentaram uma maior fluência foram os que conseguiram coordenar os recursos do programa com suas próprias ações e intenções de programação. Para esses participantes, os jogos ganharam em complexidade em correspondência a sua compreensão das relações entre parte-todo.

5.2.2 Coordenação entre os objetos e seus efeitos (design, narrativa e programação)

Como discutimos anteriormente, um dos objetivos do Etoys é possibilitar ao usuário o “acesso” à programação simultânea de objetos e à interação entre eles. Assim, rapidamente pode-se não só criar muitos objetos no “mundo”, como também editar esse mundo como o cenário em que esses objetos estão. Constatamos, durante as oficinas, que essa característica facilitou, de fato, a construção de um “design” possível para o jogo. Por outro lado, parece ter criado – nesse mesmo processo - uma espécie de intervalo entre a composição desse design e a construção da programação que corresponderia à operatividade do jogo, que desse conta da coordenação entre os objetos e seus efeitos.

Selecionamos para ilustrar esse item o caso de Liv (10 anos). Liv frequentou apenas 8 oficinas. Ao longo dos encontros, alternou momentos de entusiasmo, sobretudo no compartilhamento de ideias com os colegas quando compunha o desenho de um novo jogo, com momentos de frustração quando teve de entrar em contato com a programação dos objetos que criava. Dentre os muitos motivos que podem estar relacionados com sua desistência, nosso interesse se volta para a interação de Liv com as características específicas de programação com Etoys relacionadas as suas tentativas de compor suas intenções expressas na criação de seus projetos.

Tal como Liv, um dos movimentos notadamente presente na prática dos sujeitos, frente ao desafio de criar um jogo, foi a recriação em Etoys de um game previamente conhecido. Assim, a estudante tinha já uma clara ideia do design, do funcionamento do jogo, do comportamento dos personagens e de algumas relações entre eles. Liv desenvolve o projeto denominado “O Jogo do Sorvete”. Quando perguntada sobre a origem da ideia, diz:

“é que tem um jogo que meu irmão joga! É duas gelatina e tem uns monstro lá que pega as gelatina, que pega as banana e uns negócio.

B- Isso é de computador?

L- É, é no “friv.” (site, espécie de portal de jogos online, disponível na internet e muito popular entre as crianças)

D- Ah, “friv” eu conheço!

L- Vou mostrar, tem internet aqui?

(constatamos que não havia conexão naquele dia...)”

Na figura abaixo, vemos o jogo ao qual Liv se refere:



Figura 12. O jogo online “Bad Ice Cream”

No diálogo abaixo, Liv explica o funcionamento do jogo:

L- Não, é que esse é o sorvete, é que é tipo um labirinto, que tu tem que pegar as bananas, os coisas lá que... as fruta. E tem os monstros, um dois ou três que perseguem...

B- Ah, que perseguem o sorvete, o sorvete é o herói, então...

L- É... por aí...

B- Como se fosse.

L- É, como se fosse...

L- (desenhando) quadrado, quadrado, quadrado (como que narrando a ação)
(...)

Liv havia criado três monstros (como objetos irmãos!), e me explica o jogo.

L- Mas eles têm que ser mais lentos, e esse tem que tentar fugir!

Liv brinca com o “sorvete”, movimentando-o pela tela.

L- entendeu?

B- Sim...

L- e ele tem que sabe comer... algumas coisas!

B- então tu me falou pelo menos três coisas diferentes que o teu programa tem que ter.

L- É (rindo).

L- ele (monstro) tem que ter um sensor que pegue ele! E esse aqui tem que saber comer

e ser mais rápido do que eles!

B- por qual vamos começar, vamos começar pelo movimento deles?

L- Mas tem como fazer eles perseguir esse?

B- é isso que tá ali ó. (me referindo ao tutorial)

(...)

L- Bruno, será que dá pra fazer uma estradinha, o painel onde ele vai correr?

B- O mundo?

L- É, o mundo dele!

(...)

Liv explorava o tutorial dentro do etoys (um gato que perseguia um rato)

L- mas se ele encostar nele, ele morre! Entendeu? Senão não tem graça, né?!

B- Tá certo, mas vamos passo por passo?

L- Tá...

B- Primeiro vamos... Tu consegue descobrir, será? O que tu quer fazer é parecido com o que tá aí, não é? (no "tutorial de perseguição" oferecido)

L- É a mesma coisa!

B- E tu consegue descobrir por ai?

L- Não!

(...)

O resultado da atividade de Liv, ao final da oficina, pode ser visto na figura abaixo:

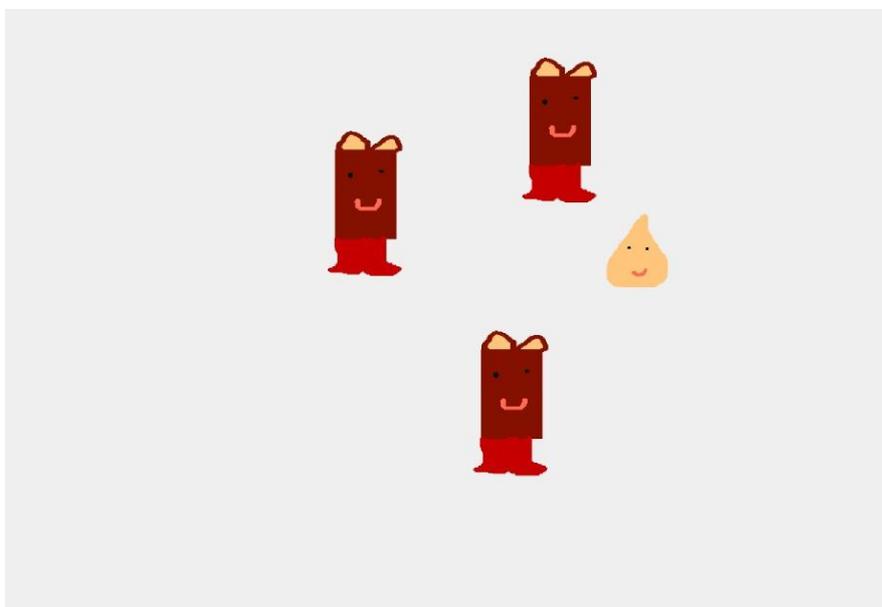


Figura 13. O "jogo do sorvete", recriado por Liv.

Será importante lembrar que o relato acima é um exemplo extraído da quarta oficina e se configura como um processo com alguma recorrência nas atividades dos participantes. Foram inúmeras as vezes em que um jogo foi planejado e um design criado pelos estudantes, mas havendo muitas variações em como esse desenho inicial foi ou não dotado de operatividade, em nível de programação.

No exemplo acima, podemos vislumbrar o nascimento de um design de um jogo.

Liv, em poucos minutos, cria seus objetos e expressa as relações e coordenações entre eles. A estudante fica satisfeita com o resultado da sua criação (*L- ó, ficou bem bonitinho!*). Sua produção inicial possibilita inferir diversos conhecimentos já adquiridos sobre as propriedades gráficas do Etoys (tais como: o uso de ferramentas de edição, a criação de objetos separados por função, a herança de objetos irmãos). Liv também expressa as relações e coordenações advindas da experiência com o jogo já conhecido. A partir desse momento, ela se dá conta da diferença entre produzir um design do jogo - e sua narratividade - e dotá-lo operatividade através da programação.

Expressos verbalmente por Liv, as funções de cada objeto/personagem poderiam ser assim esquematizadas:

- A) os comportamentos do sorvete
 - fuga do monstro
 - o sorvete deve **saber comer (se vê uma fruta...então...)**
 - é controlado pelo jogador (relações do controle com o objeto)

- B) o comportamento do (s) monstro (s)
 - perseguição (relação) ao sorvete
 - deve ser **mais lento** (relação) que o sorvete

Adicionamos à descrição do caso, ainda, o seguinte trecho de diálogo com Liv:

Liv havia criado um script para o seu “sorvete”

L- porque ele tá louco desse jeito? (a)

L- ò, que que eu fiz aqui. Tá!

(...)

L- depois, ele vai ser controlado por alguém, entendeu? (b)

O design e a narrativa constituem-se em componentes do jogo que precisam ser implementados como um programa. A jogabilidade ou operatividade do jogo não se produzem apenas a partir do design. As últimas falas de Liv, no diálogo acima apontam para a) o conflito entre a intenção de como programar o objeto e sua resposta frente ao script criado; e b) a regulação entre design e programação ainda estão por se constituir.

Liv, entretanto, não deu sequência ao seu “jogo do sorvete”. Após algumas tentativas, ela decide deixar de participar das oficinas. Na entrevista posterior com a estudante, a mesma explicitou dificuldades que encontrou no prosseguimento, ou seja, na programação do mesmo.

As dificuldades até aqui apontadas podem em um primeiro momento ser atribuídas

aos participantes e ao próprio pesquisador como, por exemplo, a falta de um conhecimento específico sobre a linguagem de programação, ou então a ações específicas (na apresentação do ambiente Etoys) que facilitassem o processo de apropriação por parte dos estudantes. Essas questões são significativas e voltaremos a elas adiante. No entanto, a análise da já referida lista de discussão (<http://forum.world.st/Why-is-Scratch-more-popular-than-Etoys-td3787377.html>) revela que as dificuldades aqui enfrentadas e colocadas como questões nos processos de aprendizagem dos estudantes têm sido objeto central de debate entre os próprios desenvolvedores da linguagem.

Segundo seu principal criador, Alan Kay, o Etoys é propositalmente “aberto”. A ideia de um “mundo vazio” deveria favorecer não só a criação manual de objetos pelo estudante, considerando seu próprio desenho, como também a criação de ferramentas próprias. Isso significa que se trata de estimular a criação e o desenvolvimento de recursos pelo próprio usuário. Para Kay, esse é o verdadeiro propósito para a construção de conhecimento em um ambiente de programação para estudantes. A tentativa do autor é tornar o “computador” uma espécie de argila para a modelagem da mais ampla gama de objetos possíveis. Pensamos que a partir de nossas observações podemos pensar alguns apontamentos.

Diversos desenvolvedores e usuários experientes apontam para a dificuldades que observamos não só em Liv., mas em inúmeras vezes e em todos os casos: há muitos itens e funções disponíveis a todo momento e não estão claros os motivos pelos quais estão ali. Tornando ao caso de Liv, poderíamos traduzir assim: após ter criado seu design e seus objetos: por onde começar a programar as ações necessárias para seu jogo funcionar?

Os participantes da lista discussão parecem convergir, então, para a ideia de que o Etoys como linguagem favorece a profundidade (High Ceiling), mas torna o processo de aprendizagem mais lento e obscuro do que seria necessário. A preocupação fundamental desses participantes da lista torna-se, então, o quanto o Etoys motiva seus usuários a continuarem tentando. Tal como veremos adiante, os participantes que permaneceram explorando um tempo suficiente o programa conseguiram desenvolver projetos interessantes, com boa apropriação da programação. O caso de Liv ilustra como as características particulares do Etoys estimularam primeiras interações e composição de design, mas também não favoreceram ultrapassar o intervalo entre essas primeiras composições e a sequência da criação, que seria a operatividade do jogo.

Os diversos autores que vimos estudando, no campo dos videogames e da criação de jogos, apontam para as práticas que envolvem a entrada neste mundo. Enquanto os primeiros apontam para a participação e trocas em comunidades, os chamados espaços de afinidade; os segundos advertem para as perspectivas de programação como fundamentais nos processos de criação, autoria, e fluência digital (BRENNAN E RESNICK, 2012). Nos processos de apropriação apontados por ambos grupos de autores, o objeto (videogame, ou projeto de programação) se torna parte da realidade do estudante em seu mundo, no sentido de que seu interesse é espontâneo, as trocas em colaboração e cooperação, remixagem e compartilhamento algo fundamental.

Em nossa experiência, podemos evidenciar alguns casos nos quais o Etoys tornou-se uma ferramenta para “pensar com”, como propunha Papert. Uma limitação de nosso estudo foi o fato de que os sujeitos não tiveram a oportunidade de acesso a outros projetos que não os seus próprios, produzidos nas oficinas. Tal como a consulta a literatura, em uma alfabetização na língua, a alfabetização digital também implica em uma imersão na “literatura” dos videogames, por exemplo. Voltando ao caso de Liv, supomos que a possibilidade de abrir o projeto de outra criança, por exemplo, de poder acessar e editar algo semelhante ao que queria fazer, sem o modelo de um adulto, poderia ser decisivo na continuidade de sua apropriação. Uma vez que seu modelo toma como base um jogo comercial, ou seja, uma literatura já em outro nível.

5.2.3 TESTES como marcas da organização de regras e suas hierarquias

A partir da atividade das oficinas, vimos crescer a importância, para a programação dos jogos, do uso dos TESTES. A constatação dessa importância não é uma novidade, uma vez que já tínhamos visto em Papert (1985) a importância do “QUANDO” em programação paralela, e em nosso piloto, o quanto o uso de condicionais possibilita criar regras do jogo e demarcar escolhas abertas ao suposto jogador. Entra assim o “leitor” no processo semiótico do jogo. No desenvolvimento das oficinas foi se tornando gradativamente claro aos participantes que, para progredir em seus projetos em direção a programação, fazia-se necessário o uso de TESTES. A maior parte das relações entre objetos pensadas pelos sujeitos envolveria pelo menos o uso de uma condição. Se retomarmos o exemplo de Liv, exposto na sessão anterior, vimos que tanto para programar as ações do monstro como do sorvete, necessitaria empregar o uso de

condicionais.

O mecanismo de teste foi usado em situações diversas e com alguma variedade de objetivos. Durante as oficinas, pudemos flagrar diferentes modos de compreensão dos estudantes ao lidar com as proposições condicionais. Conforme veremos, a maior ou menor fluência no uso das condições esteve diretamente ligada com a complexificação do jogo, não só no estabelecimento de regras e relações entre objetos como na criação de objetivos e subobjetivos, que trataremos adiante.

Conforme o emprego das condições proposicionais nos jogos dos estudantes, bem como suas manifestações observadas nas interações, chegamos a diferentes modos de compreensão dos TESTES:

- a) O TESTE não é visto como possibilidade (na programação)
- b) O TESTE é possível, mas só em determinada condição de ação
- c) O TESTE pode compor múltiplas ações

- a) O TESTE não é visto como possibilidade (na programação)

Consideramos, aqui, também as situações como a de Liv., na seção anterior, em que uma ou mais condições são concebidas como possibilidade, na intenção ou na verbalização da criança, mas não é experimentada na programação (assim, por exemplo, o jogo do sorvete, em que diferentes eventos são concebidos, mas não ainda postos em prática como tentativa).

Em que pese termos mostrado algumas possibilidades de uso dos TESTES a todos os estudantes, e que eles mesmos tenham tido acesso aos jogos uns dos outros, o mecanismo do SE nem sempre foi observável e então não incluído na programação. Abaixo, mostramos alguns exemplos de produções desse modo:

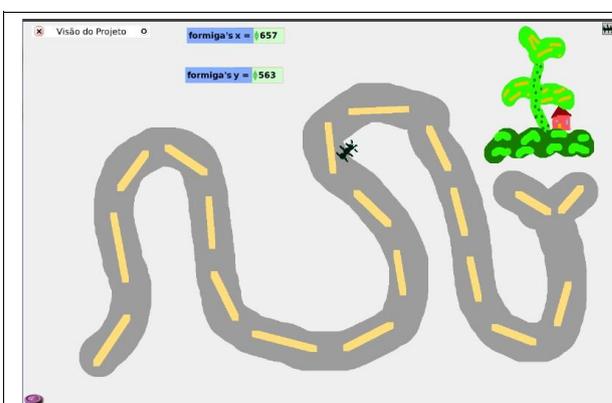


Figura 14. o “jogo da formiga”



Fig 15. o “jogo do louva-a-deus”

No “jogo da formiga” (Liv) temos como objetivo levar o objeto pelo caminho até o

objetivo final, de sua “casa”. Não há, no entanto, programação correspondente à regra do jogo: nada acontece objetivamente com a formiga se ela, ou o jogador, não cumpre seu caminho. O controle do objeto, elemento da jogabilidade, se dá pela manipulação dos atributos de x e y, presentes na tela. Podemos dizer que há já elementos de uma narratividade e jogabilidade, mas ainda não coordenadas à operatividade do jogo. O “jogo do louva-a-deus” (Joa G.) tem características bastante semelhantes, mas em sua “narrativa” há o elemento de descobri-lo, movendo-lhe por seu par X e Y até a zona de cor diferente do objeto. Em ambos os casos, há concepção, pelo criador, de um “leitor” - um jogador-que interagirá com sua produção. De fato, no encontro em que criou o “jogo do louva-a-deus”, Joa G. prontamente nos desafiou a encontrar seu objeto camuflado “adivinhandando pelo X e Y”. Apontamos, ainda, um terceiro exemplo:



Figura 16. o "jogo da velha"

No “jogo da velha”, Gab recria um jogo já conhecido (o que consideramos, por si só, uma remixagem). Para tanto, utiliza uma ferramenta disponível no Etoys chamada “botão criador” (na parte esquerda da tela, clicando nas formas de xis e círculo criamos objetos, automaticamente). Essa criação de Gab, assim como os dois exemplos anteriores, denotam preocupação dos autores com elementos da arte, do desenho do jogo (por exemplo, as cores e os detalhes criados no mundo), ainda que esses elementos não estejam relacionados ou coordenados na programação. Nesse exemplo último, mesmo que não haja a entrada no código, as regras do jogo e seus mecanismos são postos pelo conhecimento anterior do jogador. Logo depois que Gab termina seu jogo, Dan. se mostra muito interessado, e os dois passam grande parte do encontro jogando.

Essa interação é marcada, ao final, com a sugestão de Dan. para a criação de um mecanismo em que *“criasse comandos para uma seta, pra ver se “ele” (refere-se ao computador) consegue botar tudo isso no lixo (objetos que ficam pela tela após terminado o jogo), ou guardar”* (extraído da décima segunda oficina). Ainda que acatasse momentaneamente a sugestão do colega, Gab não segue com a tentativa de criar a “automatização” de seu jogo. No entanto, é interessante apontar que esse tipo de interação foi se tornando mais frequente ao longo das oficinas.

b) O TESTE é possível, mas só em determinada condição de ação

As possibilidades de criação de condições, no Etoys, são inúmeras. Podemos, para uma condição, criar tantas ações quantas forem desejadas. Assim, por exemplo, o evento de um objeto “tocar” o outro pode gerar uma série de diferentes ações: fazer um som, mostrar um texto, trocar a posição, etc., simultaneamente. De modo geral, entretanto, os sujeitos, ao término das oficinas, não chegaram a explorar a composição de possibilidades. O TESTE é reproduzido em sua forma já conhecida, não havendo invenção de novos possíveis no jogo.

Um dos exemplos recorrentes nesse caso foram situações nas quais o mecanismo de composição do TESTE é lembrado, mas as atribuições para cada uma de suas “linhas” são esquecidas pelo sujeito. Nesses casos, a memorização do procedimento possibilitou que os sujeitos criassem um determinado TESTE, mas a compreensão ainda inicial da proposição condicional e, portanto, da possibilidade de relacionar diferentes proposições e compor diferentes eventos, impossibilitou a criação (ou invenção) de novas condições. Para tornar esse aspecto do processo mais claro, apresentaremos algumas tentativas de Joa V.

Na oficina em que lhe foi disponibilizado o jogo “Onde está o Wally”, Joa V. recria uma versão do “jogo mais difícil do mundo” (que havia feito no Projeto Piloto), em que um personagem (nesse caso, uma “aranha”, ao centro da imagem) deve ir de um ponto a outro na tela passando por certos obstáculos. Em sua composição, conforme a figura abaixo, o estudante como que mescla elementos tanto do jogo recém jogado quanto de seu já conhecido projeto.

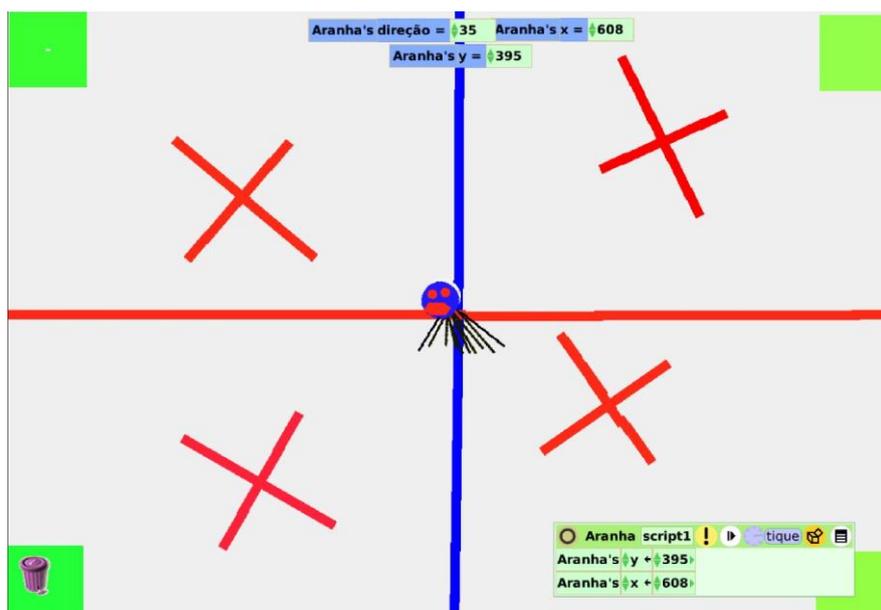


Fig. 17 -o jogo “aventuras da aranha”

Em sua tentativa, ainda que as regras e o funcionamento do jogo estejam manifestas em sua intenção, Joa V. cria um SCRIPT em que não há uma condição, ou seja, há apenas os comandos de posição (x,y) para seu objeto “aranha”. Como resultado, tem o seu objeto estático no centro da tela. Esse aparente “erro” (levando em conta seus objetivos iniciais), entretanto, não gera desequilíbrio imediato para Joa V, que se diverte em arrastar o objeto para todos os lados da tela e em seguida vê-lo voltar ao centro instantaneamente.

Em um encontro posterior, no entanto, trabalhando em conjunto com seus colegas Gui e Joa G, Joa V. vê a necessidade de adequar seu jogo: “ô Joa G, me ajuda aqui, eu quero fazer esse ‘cara’ aqui voltar ”SÓ” quando bater, daí voltar pra cá”. Como resultado, o desenvolvimento de seu projeto resulta em um jogo com a composição de um TESTE e o controle sobre o objeto pelo joystick, conforme mostramos abaixo.

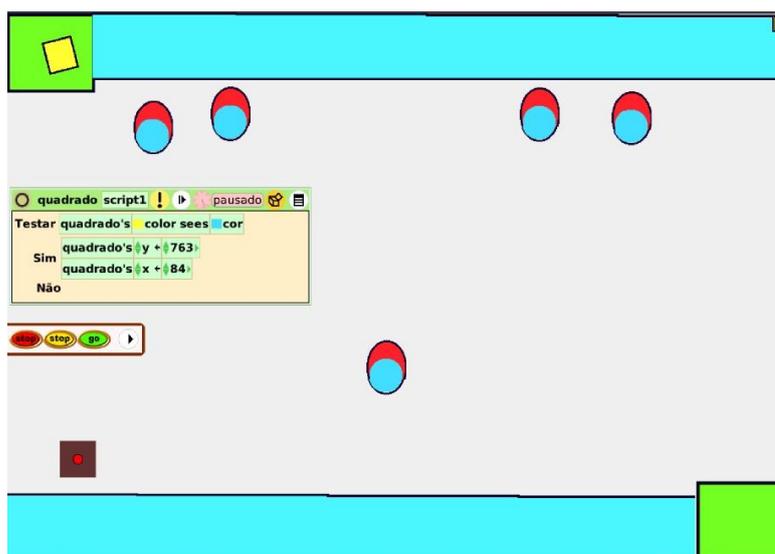


Figura 18. "As aventuras de Joa"

Na fala de Joa V, mostrada acima, vemos manifestada a criação de uma condição (se tocar no obstáculo, volta à posição inicial), criada pela necessidade de compor uma regra (derrota, se tocar no obstáculo) em seu jogo. A construção de um TESTE, então, corresponde, na programação, a essa regra e dá ao jogo um elemento novo, tanto de regra quanto de jogabilidade.

Se tomarmos minuciosamente o conhecimento envolvido no TESTE construído por Joa V, veremos que há: a lógica da proposição condicional subjacente ao algoritmo (se...então); o domínio sobre os comandos específicos da programação (quadrado "vê a cor"; estrutura da "caixa do teste", etc.) e o mecanismo de compô-los; a coordenação de um objeto em um plano cartesiano pelos atributos x e y.

A integração de todos esses aspectos para a criação de um novo TESTE, então, se mostrou um desafio muito grande para nossos participantes. No caso de Joa V., por exemplo, o jogo acima mostrado foi como que um "limite" atingido pelo estudante, não houve criação de novas regras ou relações entre objetos por TESTES em seus projetos. Em suas outras criações, o uso do TESTE esteve sempre ligado ao mesmo objetivo e na mesma condição: tornar o objeto, quando do encontro de outro, para a posição inicial.

Em alguns projetos, entretanto, pudemos observar a criação de TESTES em diferentes situações, que passamos a analisar a seguir.

c) O TESTE pode compor múltiplas condições de ação

Consideramos, aqui, dois tipos de apropriações: casos em que mesmos TESTES criados foram usados em situações diferentes, para a criação de novas regras no jogo, e casos em que TESTES foram criados com novas possibilidades de ações, ainda que para regras já existentes.



Figura 19. o jogo "Super Naves"

Tomamos como primeiro exemplo o caso de um jogo criado por Joa G, mostrado na figura abaixo:

Joa G denomina seu jogo como um "jogo de perseguição". Nele, o jogador controla, pelas setas do teclado, uma nave que deve fugir de outra nave "inimiga", em um cenário de espaço sideral. A composição de TESTES aparece, aqui, determinando regras diferentes dentro jogo: se alcançado pela nave, o herói volta para sua posição inicial (canto superior direito da tela), mas se ele se chocar contra o sol, por exemplo, irá para a "oficina" (canto superior esquerdo da tela). Ou seja, TESTES semelhantes (que condicionam a posição do objeto pelo evento de "ver um outro") são usados para compor regras diferentes. Analisando esta composição, vemos que uma mesma noção de programação é condicionada por um aspecto específico do jogo em si, elemento tanto de regra, por que especifica o comportamento dos componentes quanto de narrativa, porque aumentam as possibilidades dentro do jogo.

Dentro do mesmo exemplo apresentado, entretanto, uma outra novidade se faz presente: a inclusão de múltiplas ações para uma condição. Na figura abaixo, mostramos

na criação Joa G. dois novos aspectos.



Figura 20. TESTES compostos no jogo “Super Naves”

Na imagem da esquerda, podemos ver diferentes ações coordenadas por uma mesma condição. Joa G agregou, ao movimento controlado pelas teclas, o som de um motor para a nave. Assim, a cada vez que o jogador movimenta o objeto, pressionando o teclado, um som também é ouvido. A imagem à direita mostra uma descoberta do estudante, em que ele muda o gráfico do objeto (programado em uma animação quadro a quadro) na condição de se chocar contra o sol.

Consideramos que, nesse exemplo, o avanço de Joa G. na compreensão da lógica dos TESTES possibilitou uma integração de diferentes elementos em um sistema de jogo, com o aparecimento de novas composições. Ao mesmo tempo, esses avanços na programação estão diretamente relacionados com a preocupação, por assim dizer, de Joa G com aspectos próprios do jogo.

Consideramos que a entrada imaginada de um jogador pelo programador do jogo se mostrou, em todos os casos, fundamental para a composição de uma hierarquia e organização entre as regras criadas. De maneira geral, como pudemos ver, os participantes desejaram aumentar o grau de controle sobre os objetos, por exemplo, tornando o jogo mais “jogável”.

Esse aspecto foi essencial nos avanços possíveis sobre as noções de programação. Em diversos casos, entre eles o exemplo acima apresentado, a necessidade de aumentar a jogabilidade e o desafio levou ao esforço dos participantes em aprender procedimentos complexos.

5.2.4 Monitoramento dos objetivos e subobjetivos

Dos seis sujeitos participantes, três deles chegaram a criar jogos que, além do objetivo principal, contemplavam subobjetivos. Os subobjetivos quebram a linearidade do jogo implicando uma complexificação da jogabilidade como da própria narrativa. Consideramos, aqui, os projetos em que esses objetivos e subobjetivos foram concretizados também em nível de programação, como mostraremos a seguir.

Para ilustrar os processos que envolvem esse monitoramento, optamos por seguir o exemplo de um dos jogos criados por Joa G. (sobretudo, neste caso, por termos muitas versões de um mesmo projeto). O estudante, ao longo das oficinas, cria um jogo em que objetivos e subobjetivos, além de elementos de design, são acrescentados a cada versão. Trata-se de um jogo de perseguição em uma paisagem de pista de corrida, em que um jogador controla um carro até a marca da chegada. A figura abaixo representa a primeira versão do projeto:

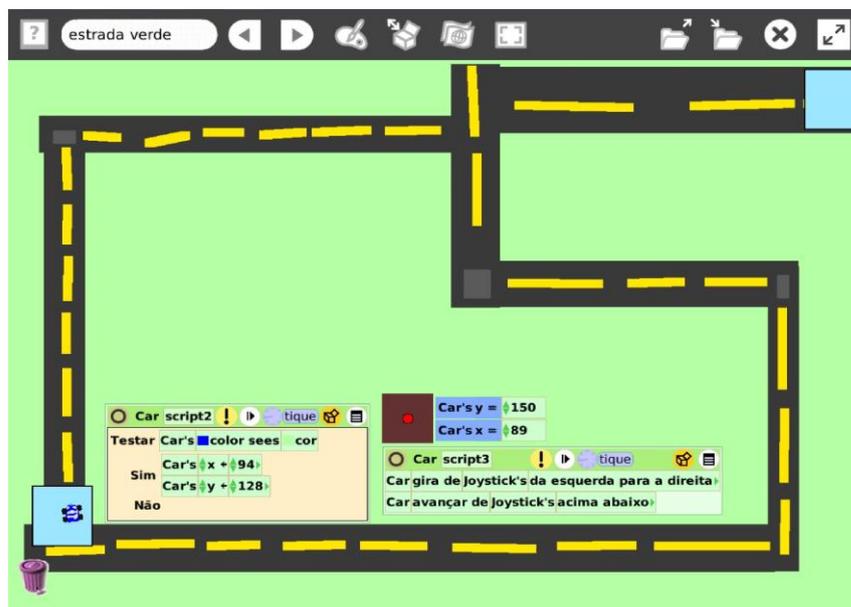


Figura 21. O jogo "estrada verde"

A primeira versão de seu jogo é criada na terceira oficina, após encontro em que haviam jogado o "Onde está o Wally". Nessa primeira composição, um carro deve ser controlado por um jogador (pelo joystick) de uma posição a outra, seguindo por uma pista que sai do canto inferior esquerdo e chegando ao canto superior direito da tela. O TESTE construído coordena o desafio do jogador: se sair da pista (ou, se "vê a cor" verde, ele volta a sua posição inicial). Temos, então, o objetivo principal do jogo: chegar até o final

da pista, com o desafio de voltar ao início se perdermos o controle do carro.

No processo de criação da primeira versão, Joa G. põe em prática alguns mecanismos de programação já conhecidos (além, é claro, das propriedades gráficas do programa): o TESTE envolvendo pares ordenados e a programação de um joystick. Ao explorar sua criação inicial, Joa G. passa por um processo em que algumas novas necessidades são criadas. O diálogo abaixo ilustra como nesse processo entrelaçam-se algumas noções de programação com objetivos na dinâmica do jogo desenvolvido.

(Diálogo extraído do diário de campo, 3ª oficina)

J- O bruno, vem cá um pouquinho

Joa G. tenta fazer um teste para seu carrinho na pista (sua tentativa é de criar a condição “se vê a cor”, vai para a posição X e Y). mas ele não seleciona o atributo do lugar certo no visualizador, o que resulta em não conseguir encaixar a proposição na caixa do teste.

J- Ah, não tá dando! Porque não tá entrando aqui?!!! (se refere ao atributo de posição que deseja colocar na linha correspondente da caixa de TESTE)

B- Pra colocar como uma ação aí, tens que pegar lá da setinha verde.

(consegue rapidamente)

J- Ah, é que eu me esqueci, faz tanto tempo! (já havia feito o mesmo teste em outros projetos na experiência piloto)

(...)

Joa havia criado dois scripts: teste e joystick que controlasse o carrinho.

Para o segundo, resulta em:

Carro gira (joystick acima baixo)

Carro avança (joystick esquerda e direita)

Dessa forma, quando vai experimentar o controle do carrinho, não consegue o resultado esperado (porque os movimentos ficaram ao contrário de sua expectativa inicial).

(Joa está experimentando seu joystick, visivelmente com dificuldades de “dirigir seu carro”)

B- O que deu aí, Joa?

J- Ah, não sei!

Liv- Que show!

J- Não sei que que deu aqui!

B- Que tá acontecendo, Joa?

(Joa vai até o script e faz ação de eliminá-lo)

B- Porque tu tá deletando esse aí?

J- Porque eu coloquei no lugar errado!

B- No lugar errado?

(não deleta o script, mas os comandos que havia dado. Mantém o script sem os comandos)

(aqui, Joa G. recordava do mecanismo de criação do controle de joystick, mas não lembrava ao certo em que linha de comando deveria colocar. Por tentativa e erro, chega

ao resultado esperado, mas não fica satisfeito com o controle do joystick)

Através do diálogo exposto, vemos como Joa G. começa a ter uma nova necessidade (melhorar o controle sobre o objeto), a partir da experimentação, nesse caso jogando, sobre o seu jogo. De fato, a atividade de jogar o próprio jogo foi constante nas oficinas: as práticas de testagem e debugging (BRENNAN E RESNICK, 2012), essenciais à atividade do programador, estiveram relacionadas diretamente ao jogarem suas próprias produções. Consideramos que essa prática como que abriu, para a criança, também a perspectiva do jogador, contribuindo para o interesse do sujeito em incrementar o seu projeto.

Abaixo apresentamos a sequência das versões seguintes do jogo desenvolvido por Joa G.

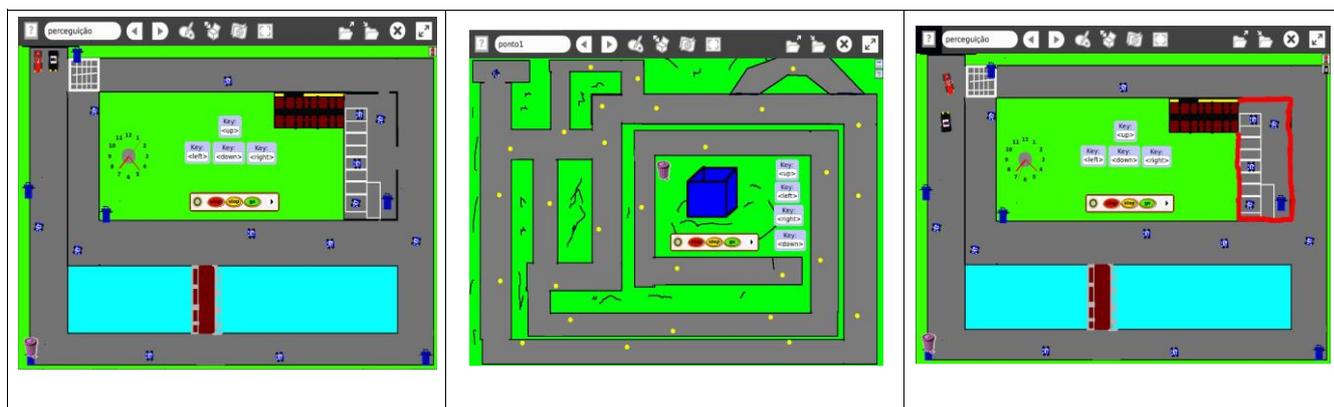


Figura 22. Sequência das versões do jogo “perseguição”

Inicialmente, há no jogo um único objetivo, que é o de controlar o carro de um ponto a outro. No desenvolvimento das versões do jogo, um novo objetivo é criado como condição para o primeiro: - chegar ao local final, fugindo de um inimigo (imagem mais à esquerda). Mas, na versão final, há também o aparecimento de outros subobjetivos: coletar pontos pelo caminho; estacionar no local correto. Dentro do jogo, há a necessidade do monitoramento do objetivo maior (chegar até o local final), com os subobjetivos: chegar até o final coletando pontos e estacionar no local certo.

Temos, assim, uma multiplicação de relações entre objetivos e subobjetivos que acrescentam complexidade em pelo menos dois níveis: ao jogo (escolhas do jogador, possibilidades de ação, narrativa e design), e à programação construída pelo sujeito. Chegando à última versão apresentada, o sujeito estabelece relações entre os múltiplos objetos criados e dispostos pela tela.

Em meio ao desenvolvimento de seu jogo, Joa G cria como subobjetivo a coleta de “moedas” pelo jogador. Através do diálogo abaixo, mostramos o momento de origem

dessa criação.

(extraído da 9ª oficina)

Dan explica para Joa G a ideia que teve de fazer um mecanismo de pontuação em seu jogo:

D- um estacionamento lá que eu fiz, daí vai tá as bolinha pra pegar e eu vou fazer um baú aqui em cima, daí quando um carro encosta nela, ela vai ir lá naquele baú. (Dan. explica o subobjetivo criado: as “bolinhas” são objetos que estão na pista e devem ser coletados pelo jogador que controla o carro. Uma vez que o jogador encontra, a bolinha vai para outro espaço (“Baú”) criado na tela)

Joa G- tu te inspirou no Castle, né?

Dan - aham.

B- no que?

Os dois- castle⁷!

Joa G- é que eu descobri um site uma coisa de jogo lá na internet. Sabe quando tu vai no google do xo, tem ali 'activis' (activities), daí aparece um monte de jogo, daí tu baixava . Era muito legal esse jogo. (Joa G. tentava sobrepor um objeto ao outro, na sua tela)

B- posso te mostrar uma “manha”?

J- uhum

B- tu quer que essa (objeto) fique sempre no fundo né?

Joa- sim, é só vir aqui e mandar ficar atrás lá (opção existente em qualquer objeto, e nunca apresentada aos sujeitos)

B- quem que o jogador dirige aí?

J- a “sky”!

(...)

B- (conversando agora com Dan.) depois quero ver contigo como tu pensou em fazer os pontos, né?

D- sim! (...) eu acho que já sei como passar de fase! Bah, agora eu to bem ligado, sabe o livro? Usa o livro e faz a cidade, aí tu faz a próxima e já tem um carrinho ali. Posso usar outro computador?

B- pode, tu que sabe...

No diálogo acima apresentado há vários pontos de interesse para analisar como se deu o processo de composição de objetivos e subobjetivos dentro de um mesmo jogo.

Assim como em uma composição escrita, na qual existe uma intertextualidade, podemos aqui dizer que um jogo se inspira em outros jogos. Aparece assim uma espécie de intertextualidade na qual uma busca em sites se torna importante, de tal modo como

⁷“Castle”: jogo que os dois estudantes costumavam jogar dois anos antes, em seus XO's (laptops disponíveis na época). No jogo em questão, o jogador tem como missão resgatar objetos escondidos pela tela; quando isso acontece, os objetos vão para a sua mochila, que fica em um canto inferior da tela

vamos a uma biblioteca selecionar textos para inspirar novos textos, os participantes buscam em sites de jogos sua inspiração para novos jogos.

O diálogo travado entre os dois estudantes lembra, também, certos aspectos do que Gee (2004) chama um “espaço de afinidade” (2007, p.87): a criação e o desenvolvimento dos jogos trouxe à tona relações – interações – de colaboração entre os participantes. No exemplo acima, Joa G. se destacava no grupo pela maior fluência na programação, mas é uma ideia dada por seu colega “menos experiente” que transforma e remodela sua criação. Há um conhecimento construído e compartilhado pelos participantes (o jogo conhecido pelos dois) que os levam a novas composições. É como se, no espaço da Oficina, os estudantes (como designers) estivessem criando o seu próprio “espaço de afinidade”.

IV. CONCLUSÕES

Após o percurso realizado, retomamos o problema e os objetivos da pesquisa para avaliar até que ponto os alcançamos para, em seguida, refletirmos sobre quais as principais implicações decorrentes do trabalho.

Partimos por questionar se as operações que Squire (2005) atribuía à atividade de jogar videogames também seriam válidas para pensar a atividade de programar um jogo em Etoys? Os quatro processos que um jogador aprende ao jogar videogames, segundo o autor, também se revelaram interessantes no momento que nos colocamos diante da tarefa de analisar a criação de jogos. A primeira resposta seria sim. Mas todos esses processos necessitaram, de vários modos, ser ultrapassados.

Os participantes da oficina tiveram a necessidade de aprender a ler o Etoys como um sistema semiótico, cuja semântica e sintaxe possuem especificidades, como pudemos observar na descrição da linguagem bem como na comparação com a Linguagem Logo e o Scratch. Mas, para além de um processo de leitura, nossos participantes foram convocados a um processo de escrita nesse novo domínio, o qual traz de modo correlato a questão do leitor, ou melhor, do possível jogador do jogo projetado. Conceber um jogo é, para além de ler, criar um sistema. Embora o jogo não seja concebido de modo imediato como um sistema complexo, os participantes encararam essa necessidade, alguns desistindo e outros persistindo.

Lembrando a assertiva de Papert (1985) e suas engrenagens, alguns de nossos participantes realmente pareceram “se apaixonar” pelo trabalho com o Etoys que, nesses casos, transformou-se em um objeto-para-pensar-com.

Também concordamos que o propósito de criar um jogo coloca a necessidade de coordenar os movimentos e seus efeitos. Mas não são movimentos e efeitos que pré-existem. No caso, os participantes necessitaram idealizar tais ações e relações. Para isso, fizeram uso da experiência anterior com outros jogos. Isso é interessante, pois novamente coloca a questão da programação em analogia à alfabetização. Na composição de novos textos recorreremos a textos anteriores, assim como na proposição de jogos, esses programadores iniciantes recorreram a sua experiência anterior com jogos.

Novamente pensamos que na atividade de programação é necessário ir além do entendimento das regras e de suas hierarquias. Primeiramente porque estamos tratando

com dois sistemas inter-relacionados. Um deles, como referimos, consiste no próprio Etoys, que embora seus idealizadores o proponham como uma espécie de argila que pode assumir muitas formas, não podemos ignorar que existem regras e hierarquias também em Etoys. O segundo sistema, que se constrói dentro do primeiro, é o jogo propriamente dito. Assim, estamos trabalhando com regras e hierarquias entre dois sistemas coordenados, o que complexifica muito a coordenação apontada por Squire (2005).

A mesma reflexão pode ser estendida quando tratamos monitoramento contínuo de objetivos e subobjetivos do jogo. No caso, não se trata somente de um monitoramento, mas da invenção dos mesmos. O que novamente complexifica os movimentos e processos de aprendizagem. Vimos que para os participantes conseguirem implementar objetivos e subobjetivos houve a necessidade de outras aprendizagens, tal como uso de condicionais.

Diante do exposto, podemos chegar a uma primeira conclusão de nosso trabalho - criar videogames coloca em movimento e complexifica os processos propostos por Squire. Mas, de certo modo, isso poderia já ser previsível. O importante no caso é indagar que outros movimentos de aprendizagem emergem quando convidamos não programadores a criar jogos e não somente a jogá-los? Essa indagação segue a sequência de questões que nos propomos quanto definimos o problema da pesquisa que perguntavam sobre quais conceitos de programação estariam envolvidos na programação de um jogo em Etoys? Como esses conceitos podem estar coordenados no contexto de um jogo?

Já na definição do problema anunciamos alguns desses conceitos tais como a regulação entre design, narratividade e jogabilidade. Evidenciamos que essa regulação pressupõe a distinção entre essas noções e seus modos de ação. Pudemos ver como essa regulação se opera progressivamente, existindo, inicialmente uma indistinção entre desenho na tela do computador e um objeto digital. Ou mesmo a noção que o próprio background no qual os objetos digitais se movimentam e se afetam mutuamente é também um objeto e não um mero cenário. A essa indistinção nomeamos de problema “dinamismo” x “paisagem” e se mostrou constantemente presente. Assim, não raras vezes o design de um jogo surgia rapidamente na tela do computador, mas mais como um desenho, ou paisagem, do que como um espaço de jogo “operativo”. É claro que há vários intermediários entre um e outro, o que procuramos mostrar através dos exemplos construídos nas oficinas.

A paisagem como elemento da narrativa não se distingue de início da

operatividade e, por esse mesmo motivo, não se articula com a jogabilidade. Configuram como se fossem um conjunto totalizante. Sobre o favorecimento da paisagem em detrimento operatividade, pensamos que contribui para isso a própria forma como o ambiente é apresentado: um “canvas” aberto e pronto para ser habitado pelos desenhos que o usuário pode rapidamente fazer. O Etoys, de certa forma, parece induzir o usuário para esse tipo de pensamento.

Quanto aos limites e dificuldades da linguagem Etoys

No que concerne aos limites e possibilidades do Etoys como uma linguagem, pensamos ter contribuído para o entendimento de que o processo de apropriação do ambiente colocou em jogo, para os estudantes, diversas práticas e perspectivas levantadas pela literatura como essenciais para a entrada no mundo da criação e autoria com tecnologias digitais.

Consideramos que o Etoys mostrou ser um interessante dispositivo de pesquisa e, para além disso, caracterizou-se como um domínio de conhecimento nos quais os participantes tiveram oportunidade de se confrontar com objetos digitais, regras, hierarquias e monitoramento de seus próprios objetivos criados.

Quanto às limitações do programa, os desafios enfrentados pelos nossos estudantes em seus projetos contribuíram para a compreensão das características específicas da linguagem e da interface Etoys. Como pudemos ver, frente a nossa proposta de construção de jogos, o ambiente favoreceu a rápida composição de um design, ou paisagem, mas sua estrutura e apresentação têm como efeito a criação de um intervalo entre esse primeiro desenho e sua operatividade: a entrada na linguagem de programação. Da mesma forma, a intenção de ser uma ferramenta poderosa de criação (o que, de fato, é) traz muitas possibilidades simultâneas que dificultam o acesso ao programador iniciante. Essas possibilidades são muitas vezes pouco intuitivas, sendo a diferenciação entre elas dada apenas pela gramática (como no caso dos scripts e suas categorias). Em nosso caso, vimos como alguns sujeitos apresentaram dificuldades em transpor esses desafios. Tal ultrapassagem acarreta em muitas horas de programação e um grau de interesse elevados.

Muitas questões levantadas por nosso trabalho, como vimos, se constituem em preocupação dos próprios desenvolvedores da linguagem e influenciaram novos desenvolvimentos, sobretudo o Scratch (especificamente, o Scratch compõe características de várias linguagens, principalmente do Logo e do Etoys, ou seja, linguagens procedural e orientada a objetos).

Quanto a atividade de programar jogos como um recurso para aprender

Colocando a atividade de nossos sujeitos em perspectiva, retomamos a questão: o que aprenderam ao construir jogos em Etoys? Pensamos que uma resposta a essa questão começa a ser encontrada parcialmente na atividade de construção do jogo, mas vai além dessa atividade. Mais do que a interação com o Etoys ou a apropriação da linguagem, destacamos algumas situações em que as atitudes dos estudantes indicam a dimensão de autoria e criatividade na atividade de programar. Uma delas é demonstrada no exemplo abaixo:

“Joa V e Gui, depois das primeiras oficinas, em que conseguiram reproduzir com sucesso seus “jogo mais difícil do mundo”, repetem uma mesma atividade novamente a cada início de encontro: juntam-se em frente a um computador e, juntos, jogam o jogo construído por um deles. Os dois visivelmente se divertem, rindo, enquanto tentam chegar ao final do desafio.”

(Exemplo extraído da quarta oficina)

Para além do “que” os estudantes estariam aprendendo na programação, consideramos que a possibilidade de construir jogos convocou os participantes para um exercício de autoria, ao colocar-se na posição do próprio designer ou desenvolvedor do jogo. Assim, além de noções de programação, consideramos que os estudantes puderam performar o que é “ser” um criador (nas palavras de Papert, um construtor). Uma dessas manifestações, a nosso ver, se dá pelo contínuo prazer que os estudantes tiveram em brincar com seus próprios jogos, depois de algum objetivo atingido.

Também nesse sentido, devemos considerar algumas observações realizadas sobre a postura dos estudantes ao longo de nosso acompanhamento. Por exemplo, em meio as oficinas os sujeitos solicitaram pendrives para organizar os arquivos de seus projetos e poder transitar com eles. Alguns deles, a partir daí, passaram a trazer projetos desenvolvidos fora das oficinas, assim como levar as modificações que haviam feito durante os encontros.

Um outro exemplo de exercício de autoria deu-se quando a professora regente relatou a realização de uma tarefa com toda a turma utilizando o Etoys para a construção de polígonos. Segundo seu relato, nossos participantes tiveram uma atuação de monitores e referência frente aos colegas, ajudando-os nos problemas que surgiam na atividade.

Essa atitude também foi observada em um encontro que tivemos em uma escola participante do Programa UCA, que envolvia professores e alunos de outras cidades. No

evento, a atitude da menina Gab. (uma de nossos participantes) causou surpresa. Durante as oficinas, Gab dificilmente expressava suas ideias ou desejo de progredir em um projeto. No referido encontro, com alunos de outras escolas que não tinham conhecimento em Etoys, no entanto, Gab., assumiu uma postura de “autoridade” frente aos iniciantes (em que professores estavam incluídos), passando a ensinar noções aprendidas e, chegando à composição de um jogo de “labirinto” que ainda não havia criado.

Perspectivas para novas pesquisas

Acreditamos que uma possível limitação deste estudo tenha sido a falta de um apoio nas oficinas de um facilitador da informática. O conhecimento mais apurado de programação poderia se constituir em uma alavanca ao desenvolvimento dos projetos, uma vez que esse pesquisador tomou mais a posição de acompanhar o trabalho e aprender com os participantes.

Da mesma forma, a experiência no acompanhamento institucional, assim como a consulta à literatura, mostra que embora as tecnologias digitais estejam cada vez mais disponíveis, a sua apropriação como objeto de aprendizagem pela escola é ainda um desafio.

Cabe salientar também um movimento recente de “aprender a programar” que tem se disseminado pelo campo da cultura digital e escolar. Na Inglaterra⁸, por exemplo, a programação será uma disciplina obrigatória no currículo nacional ainda neste ano. Se pensarmos em termos de inserção no mercado, etc., a possibilidade de poder criar, produzir com as tecnologias digitais tende a ser uma necessidade crescente, uma vez que cada vez mais as tarefas profissionais se valem de sistemas informáticos. Quando os trabalhadores ficam alijados de um entendimento sobre a programação permanecem na posição de usuários, dependentes dos programadores desses sistemas.

Outro dado que merece atenção em futuras investigações é como a questão de gênero perpassa o mundo digital. Em nosso estudo tais diferenças também estiveram presentes. Foi flagrante como parecia mais espontâneo e desafiador aos meninos a atividade de programar. Essa é uma discussão que necessita ser enfrentada e estudada pois a entrada de novas tecnologias não afasta práticas de distinção e de hierarquização de gênero. Essa discussão põe em pauta como as expectativas de gênero constituem lugares sociais definidos e podem limitar o sonho de Papert com a criação da linguagem

⁸ essa inclusão já foi anunciada pelo governo britânico (<https://www.gov.uk/government/organisations/department-for-education>), e iniciará em setembro de 2014.

de programação para estudantes.

Para finalizar, vivenciamos o quanto as oficinas se constituíram como um espaço de produção de “desenvolvedores” de jogos, ou seja, os participantes construíam seus jogos, falavam sobre seus jogos, e jogavam seus jogos: um conhecimento produzido e criado por eles. Com isso, cremos importante fomentar e disseminar espaços de produção nos quais a programação seja um dos objetos de afinidade que possa congrega diferentes segmentos escolares e, talvez, problematizar essa própria segmentação. Observamos a alegria dos participantes em produzir algo que funcione e com o qual possam brincar e convidar colegas a também fazê-lo.

REFERÊNCIAS

- ALMEIDA, M. E. B. e PRADO, M. E. B. B. (2011) “Indicadores para a formação de educadores para a integração do laptop na escola”. In ALMEIDA, M. E. B. e PRADO, M. E. B. B. (Org.). “O computador portátil na escola: mudanças e desafios nos processos de ensino e aprendizagem”, São Paulo, Avercamp.
- BARANAUSKAS, M. Procedimento, função, objeto ou lógica? Linguagens de Programação vistas por seus paradigmas. Artigo. Campinas, 1993
- BATTRO, A, DENHAM, P. (2007) Hacia una inteligencia digital. Buenos Aires, Academia Nacional de Educacion.
- Brennan e Resnick (2012) New frameworks for studying and assessing the development of computational thinking. Artigo apresentado no annual American Educational Research Association meeting, Vancouver, BC, Canada. AERA, 2012
- CGI, 2012 Pesquisa sobre o uso das tecnologias de informação e comunicação no Brasil [livro eletrônico]: TIC Domicílios e Empresas, 2012 São Paulo: Comitê Gestor da Internet no Brasil, 2013.
- FAGUNDES, L. C. e MARASCHIN, C. (1992) “A linguagem LOGO como instrumento terapêutico das dificuldades de aprendizagem: possibilidades e limites”, In Psicologia: Reflexão e Crítica, Porto Alegre, v. 5, n. 1.
- FAGUNDES, L. C., MAÇADA, D. e SATO, L. (1999) “Aprendizes do futuro: as inovações começaram”, Coleção Informática para a Mudança na Educação, Brasília, SEED, MEC, PROINFO.
- FAGUNDES, L. C. A Psicogênese das condutas cognitivas da criança em interação com o mundo do computador. Tese de Doutorado. São Paulo: IP/USP, 1986.
- FERRETTI, Cláudio. Processo cognitivo e objetos interativos de aprendizagem: a construção do equilíbrio físico. Porto Alegre: UFRGS, 2007. Tese (Doutorado em Informática na Educação), Programa de Pós-Graduação em Informática na Educação, Universidade Federal do Rio Grande do Sul, 2007
- GEE, J. Situated language and learning: A critique of traditional schooling. London: Routledge, 2004
- GEE, J (2007). What Video Games have to teach us about learning and literacy. New York: Pallgrav Macmillan, 2007
- GEE, J (2007). Good videogames + good learning: collected essays on videogames, learning and literacy. New York: Peter Lang Publishing, 2007.
- HOFFMANN, D. Aprender matemática: tornar-se sujeito da sociedade em rede. Porto Alegre: UFRGS, 2006. Dissertação (Mestrado em Psicologia Social e Institucional), Programa de Pós-Graduação em Psicologia Social e Institucional, Universidade Federal do Rio Grande do Sul, 2006.
- HOFFMANN, D. Modalidade 1:1: tecnologia individual possibilitando redes de fluência digital. Tese (Doutorado). Universidade Federal do Rio Grande do sul. Centro Interdisciplinar de Novas tecnologias na Educação. Programa de Pós Graduação em

Informática na Educação, 2011. Porto Alegre, RS, Brasil.

KAFLAI, Y e PEPLER, K. Developing Gaming Fluencies with Scratch: Realizing Game Design as an Artistic Process. In: Games, Learning and Society. Learning and Meaning in the Digital Age, (p. 355-380), 2012

KASTRUP, V. A invenção de si e do mundo: uma introdução do tempo e do coletivo no estudo da cognição. Campinas: 1999.

KASTRUP, Virgínia. O método da cartografia e os quatro níveis da pesquisa-intervenção In: CASTRO, Luica Rabello & BESSET, Vera Lucia (Orgs.). Pesquisa-intervenção na infância e juventude. Rio de Janeiro: Trarepa/FAPERJ, 2008, p. 465-489.

KAY, A. (2007). Children Learning by Doing Squeak Etoys on the OLPC XO (VPRI Research Note RN-2007-006-a). Glendale, CA: Viewpoints Research Institute. Disponível em: http://www.vpri.org/pdf/rn2007006a_olpc.pdf. Acessado em: 05/10/2012

KAY, A. (2011) "Squeak Etoys Authoring & Media", <http://www.squeakland.org/resources/articles/article.jsp?id=1008>.

KAY, A. Computer Software. 1984. Disponível em: http://www.vpri.org/pdf/tr1984001_comp_soft.pdf. Acessado em: 9/10/2012

KAY, A. 2013 Vídeo-Aula intitulado "Powerful Ideas" do curso "Learning Creative Learning", disponível em: <http://ilk.media.mit.edu/courses/video.php?provider=youtube&vid=yNlyDr8dem8>. Acessado em: 4/03/2013

KIST, Sílvia Oliveira. Um laptop por criança: implicações para as práticas de leitura e escrita. Porto Alegre: UFRGS, 2008. Dissertação (Mestrado em Educação), Programa de Pós-Graduação em Educação, Universidade Federal do Rio Grande do Sul, 2008.

JUUL, Jesper (2011). Half-real. *Video games between real rules and fictional worlds*. Cambridge: MIT Press.

LIRA, Antonio da Fonseca de. O processo da construção do conceito matemático de limite pelo aprendiz com utilização de objetos digitais. Porto Alegre: UFRGS, 2008. Tese (Doutorado em Informática na Educação), Programa de Pós-Graduação em Informática na Educação, Universidade Federal do Rio Grande do Sul, 2008.

LINDNER, Edson Luiz. Uma arquitetura pedagógica apoiada em tecnologias da informação e comunicação: processos de aprendizagem em química no ensino médio. Porto Alegre: UFRGS, 2009. Tese (Doutorado em Informática na Educação), Programa de Pós-Graduação em Informática na Educação, Universidade Federal do Rio Grande do Sul, 2009

Lista de Discussão dos desenvolvedores do Etoys (2011) <http://forum.world.st/Why-is-Scratch-more-popular-than-Etoys-td3787377.html>. Acessado em Setembro de 2013.

LONGO, J. A aprendizagem por projeto e a pesquisa psicanalítica. Dissertação de

Mestrado. Porto Alegre: Universidade Federal do Rio Grande do Sul. Instituto de Psicologia. Programa de Pós-Graduação em Psicologia Social e Institucional, 2012.

LOPES, D. A exploração de modelos e os níveis de abstração nas construções criativas com robótica educacional. Porto Alegre: UFRGS, 2008. Tese (Doutorado em Informática na Educação), Programa de Pós-Graduação em Informática na Educação, Universidade Federal do Rio Grande do Sul, 2008

LOPES, D. Civilização e Progresso: lógicas de permanência e de acúmulo em jogo de simulação. Porto Alegre: UFRGS, 2000. Dissertação (Mestrado em Psicologia Social e Institucional), Programa de Pós-Graduação em Psicologia Social e Institucional, Universidade Federal do Rio Grande do Sul, 2000.

MACKLIN, C. e SHARP, J.. "Freakin Hard': Game Design and Issue Literacy". In: Games, Learning and Society. Learning and Meaning in the Digital Age, (p. 381-402), 2012

MARASCHIN, C. Processos cognitivos envolvidos na atividade de crianças de 4 a 6 anos com a linguagem Logo de programação. Dissertação de Mestrado. Porto Alegre: Universidade Federal do Rio Grande do Sul, Faculdade de Educação, 1986.

Maraschin, C & Baum, Carlos (2013). Videogames como objetos interessantes ao estudo da cognição. Reflexão e Ação. v. 21, n. 2.

MATURANA R., Humberto; VARELA G., Francisco. A Árvore do conhecimento: as bases biológicas do entendimento humano. Campinas, SP. Editorial Psy II, 1995.

NEVADO, R. A e MARASCHIN, C. O Desenvolvimento Cognitivo de Crianças com Necessidades Especiais em Interação com o Ambiente LOGO. In Psicologia: Reflexão e Crítica, Porto Alegre, v. 5, n. 1.

NEVADO, R. A. (1992) "A utilização do microcomputador na pré-escola como prevenção de distúrbios de aprendizagem", In Psicologia: Reflexão e Crítica, Porto Alegre, v.5, n.1.

NICOLELIS, M. (2011). Muito Além do nosso Eu. São Paulo: Companhia das Letras.

PAPERT, S. (1985) LOGO, Computadores e Educação. São Paulo, editora Brasiliense.

PAPERT, S. (1994) A máquina das crianças: repensando a escola na era da informática. Porto Alegre: Artes Médicas.

PAPERT, S. Situating Constructionism. In: Constructionism, editado por I. Harel e S. Papert. Norwood, NJ: Ablex Publishing, 1991.

PIAGET, J. (1990) "A formação do símbolo na criança: imitação, jogo e sonho, imagem e representação", Rio de Janeiro, Zahar Editores.

PIAGET, J. *A representação do mundo na criança*. Rio de Janeiro: Record, 1975.

PIAGET, J. A Teoria de Jean Piaget, in CARMICHAEL Manual de Psicologia da Criança. Organizado por Paul Mussen. São Paulo: Editora da USP, 1977, vol. IV, p. 71 a 116.

RESNICK, M. (2002) "Rethinking Learning in the Digital Age". MIT MediaLab, <http://ilk.media.mit.edu/papers/rethinkport.doc>.

RESNICK, M., e WILENSKY, U. (1997). Diving into Complexity: Developing Probabilistic

Decentralized Thinking through Role-Playing Activities. *Journal of the Learning Sciences*, vol. 7, no. 2, pp. 153-172. disponível em: <http://web.media.mit.edu/~mres/papers/starpeople-final.pdf>. Acessado em: 10/09/2012

SEDUC-RS. Site da Secretaria Estadual de Educação do Rio Grande do Sul. Disponível em: http://www.educacao.rs.gov.br/pse/html/proj_provincia.jsp. Acessado em Março de 2014.

Squire, K.D. (2005). Educating the fighter. *On the Horizon* 13(2), 75-88.

SQUIRE, K. Designed Cultures. In: *Games, Learning and Society. Learning and Meaning in the Digital Age*, (p. 10-31) 2012

SQUIRE, K. (2011). *Video Games and Learning: Teaching and Participatory Culture in the Digital Age*. New York, NY: Teachers College Press. ISBN: 0807751987, 2011

SQUIRE, K. From information to experience. Place-based augmented reality games as model for learning in a globally networked society. *Teachers College Record*, n.10, 2565-2602, 2010. Disponível em: <http://website.education.wisc.edu/kdsquire/tenure-files/01-TCR-squire-edits.pdf>. Acessado em Julho de 2013.

KURT D. SQUIRE, LEVI GIOVANETTO (2008) The Higher Education of Gaming, *E-Learning and Digital Media*, 5(1), 2-28. Disponível em: <http://dx.doi.org/10.2304/elea.2008.5.1.2>. Acessado em: Agosto de 2013.

SCHAFER, P. SPERB, B. E FAGUNDES, L. Squeak Etoys na modalidade 1 para 1: programação e autoria multimídia no desenvolvimento da conceitualização. *Anais do XXII SBIE - XVII WIE 2011*.

SCHÄFER, P. O percurso das enunciações em projetos de aprendizagem na modalidade 1:1. Porto Alegre: UFRGS, 2008. Dissertação (Mestrado em Psicologia Social e Institucional), Programa de Pós-Graduação em Psicologia Social e Institucional, Universidade Federal do Rio Grande do Sul, 2008.

VARELA, Francisco. *Quel savoir pour l'etique?* Paris: La Découverte, 1996.

VEEN, W.; VRAKING, B. (2009) *Homo Zappiens: educando na era digital*. Trad. de Vinícius Figueira. Porto Alegre: Artmed.

VOELCKER, M. Autoria, cooperação e aprendizagem em comunidade virtual construída e protagonizada por educadores e aprendizes de telecentros : uma possibilidade a partir da pedagogia de projetos de aprendizagem integrada ao ambiente AMADIS. Porto Alegre: UFRGS, 2006. Dissertação (Mestrado em Psicologia Social e Institucional), Programa de Pós-Graduação em Psicologia Social e Institucional, Universidade Federal do Rio Grande do Sul, 2006.

WENGER, E.; McDERMOTT, R; SNYDER, W. M. *Cultivating communities of practice*. Cambridge, MA: Harvard Business School Press, 2002. Disponível em <http://cpcoaching.it/wp-content/uploads/2012/05/WengerCPC.pdf>. Acessado em Agosto de 2013

WILENSKY, U e RESNICK, M. (1999). Thinking in Levels: A Dynamic Systems Approach to Making Sense of the World. *Journal of Science Education and Technology*, vol. 8, no. 1, pp. 3-19. Disponível em: <http://web.media.mit.edu/~mres/papers/levels.pdf>. Acessado em: 08/09/2012