2012

Unit Plan

An introduction to Programming with Scratch and associated Algorithms

Technology

This unit of work has been designed fit for purpose to introduce students to basic computer programming. It forms part of Level One Digital Technologies Course for 2012.

Knowledge gained by students in this Unit will be used to complete Achievement Standards AS 90176 ‘Construct a basic computer program for a specified task’ and AS 91075 ‘Construct an algorithmic structure for a basic task.’

Contents

Topic Theme:

An Introduction to Computer Programming with Scratch

Focus Curriculum Area:

This Unit has been designed for Year 11 students who are completing the Level One Digital Technologies course. In a Face to Face class situation that will run over 5 – 6 weeks, the Moodle component will provide a blended learning environment, that helps support students who are absent, accommodates diversity and invites collaborative independent learning.

Participants:

I will be working with the same Year 11 ICT students from the first teaching pilot; however our group has now grown to 15. Only two students have used Scratch before and their abilities could be determined as novices. The other students are completely new to the programme. 10 students have access to a computer at home and have been encouraged to download the programme to their own computers.

Focus Strands:

Technology: Level 6

This unit of work aligns with Level 6 Technology strands, Technological Systems and Technological Practice. There are also links to other areas of the curriculum including the Arts, utilising design principles and Maths, incorporating integers and algebra.

Technological Systems

- Understand the implications of subsystems for the design and development of technological systems.

Technological Practice

- Critically analyse their own and others' past and current planning practices through the development of algorithms, to support effective use for planning tools.

Selected Competencies:

Relating to others:

Students will work together and independently to critique and examine games designed in Scratch via the online Scratch.mit.edu community and downloaded examples. Students will also be encouraged to peer assess shared examples through the discussion forum.

Using language symbols and texts:

Students will be introduced to new computational terms and encouraged to use them to help their developing understanding for programming concepts. The Scratch glossary will support understanding along with formative assessment via quizzes.

Sequences	Set of instructions in order	Debugging	Finding and correcting mistakes so a program works correctly. May involve tracking variables.
Procedures	Small chunks of instructions that do a specific job, called from the main program and returning to the main program when the action is completed. Same meaning as sub-routine	Variable	A name for a storage place in computer memory. Can be of many different types. Contains a value which can be changed
Parallelism	Control different actions to sprites simultaneously.	Initialize	Set the beginning value of a variable
Events	When key pressed and when sprite clicked are examples of event handling – responding to events triggered by the user or another part of the programme	Coordination synchronization	broadcast and when I receive can coordinate the actions of multiple sprites. Using broadcast and wait allows synchronization.
Conditionals	Decisions based on certain conditions	Integer	A whole number
Operators	Support for mathematical and logical string expressions.	Iteration	Repeating sets of steps May be counted or conditional Forever and repeat blocks
Increment	Add one(or another number) to the value of a variable	Looping	Repeating sets of steps May be counted or conditional
String	Words, phrases or characters	Nested	One control structure inside another one
Data	Storing and retrieving. Updating values.	Output	The results of running the program
Threads parallel execution	Launching two stacks at the same time creates two independent threads that execute in parallel.	Input	What the user enters
Boolean Logic	and, or, not are examples of boolean logic.	Dynamic interaction	mouse_x, mouse_y, and loudness can be used as dynamic input for real-time interaction

Selected Values:

Community and Participation

Students will be encouraged to actively engage in the class forums in Moodle to comment on each other's computer games. Students will also be encouraged to interact with the global online community either in http://scratch.mit.edu/ or YouTube.

Content Learning outcomes:

After 5 weeks students should be able to execute basic computer programming in 'Scratch' to create their own computer game, using the following computational concepts as defined by Brennan, K., & Resnick, M., (2012). They should also be able to troubleshoot their game and provide a debugging log of how they improved it.

- Sequences
- Loops
- Parallelism
- Events
- Conditionals
- Operators
- Data

To compliment their Scratch game students will need to create an Algorithm in PowerPoint that shows the key steps involved.

Focused Moodle Learning Outcomes:

After 5 weeks students should be;

- Developing more confidence to communicate via online class forums
- Able to lead a discussion by showing an example of their work
- Confident users and editors of the class glossary
- Able to confidently navigate their way to a variety of resources in Moodle
- Exploring the Scratch.mit.edu online community and possibly participating in Scratch help forums
- Confident about how to find information via the internet that is useful to develop further understanding which may lead to more independent learning from the student's aspect.

Focus Area for this Teaching Pilot:

1. Learning design

In the designing of the Moodle course that supports this unit of learning I have tried to implement 2 concepts as described by Masterman, E., Jameson, J., & Walker, S., (2009, pg.224)

(1) A focus on learning through activity rather than through the absorption of content
(2) The structuring of those activities into sequences

In my last teaching pilot I made extensive use of the 'lessons' feature in Moodle and found it to be an invaluable tool, particularly for those students who missed classes or needed to refresh their memory on certain aspects of html.

I am also planning to use conditional formatting in some areas in an attempt to scaffold the content more effectively. With considered application I hope to structure activities that will step student's through, rather than overwhelming them with many concepts at once.

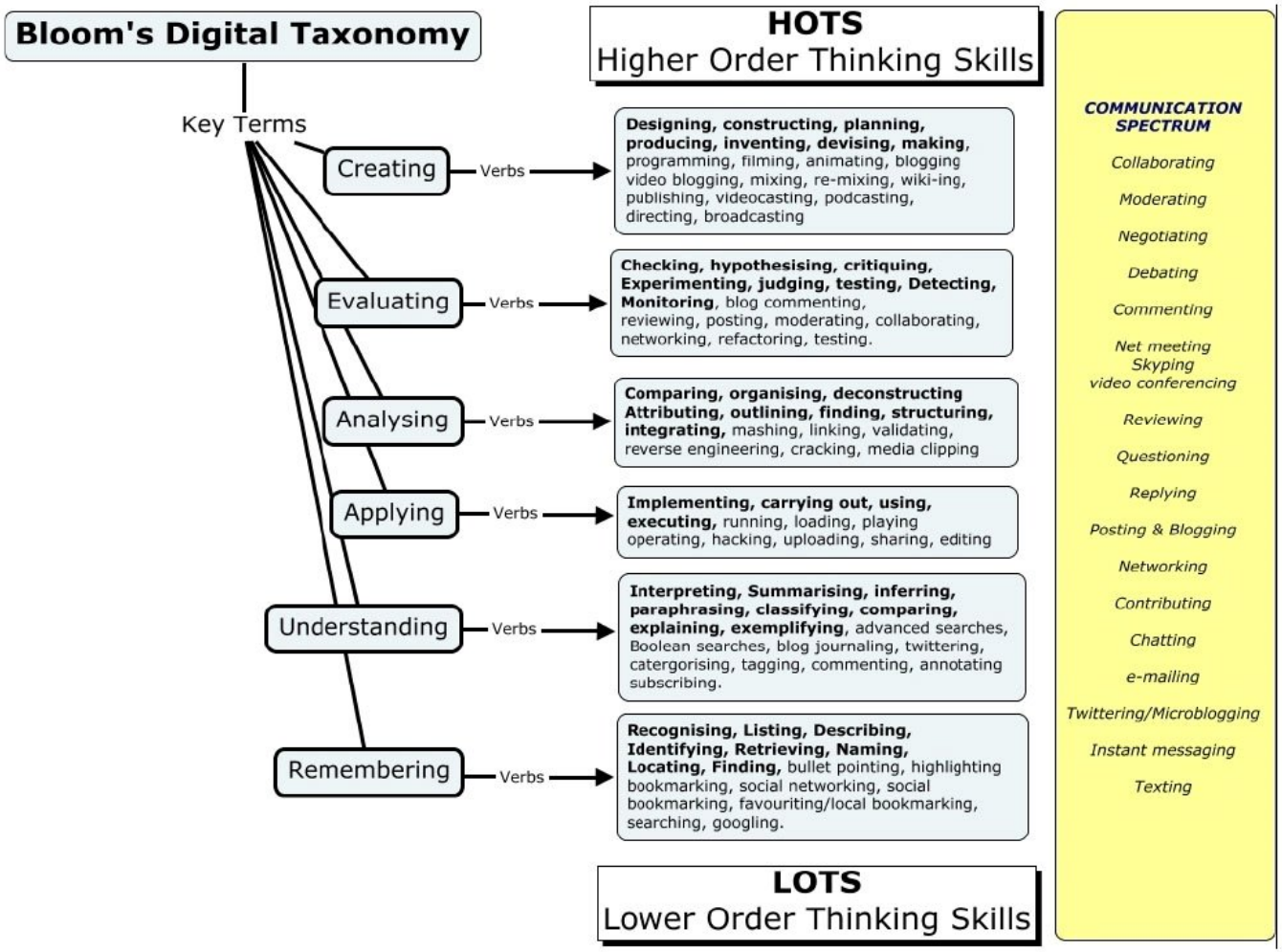
Rhode, J., (2008, pg.1) explains 'interaction is understood to be a fundamental element for quality learning environments'. Churches A.,(year and page number) with his revised Bloom's taxonomy for the digital medium says it's 'the quality of the action or process that defines the cognitive level, rather than the action or process alone'.

With this in mind I have decided to assign students with the task of becoming 'Scratch experts'. Their job will be to find something new to do in Scratch that we haven't covered and then to post their findings to a new discussion in the forums. They can then show off their work and lead a discussion providing further help if required and feedback to their peers.

2. Student Engagement/ Participation/Interaction

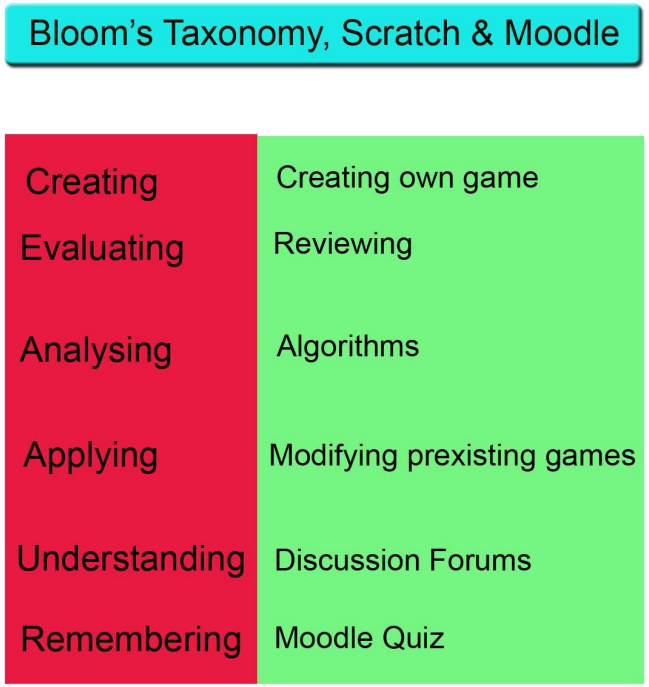
Both Scratch and Moodle are built on strong social constructivist pedagogies Antonenko, P., Toy, S. & Niederhauser, D. (pg.4), Brophy, J. (1998) states 'students will need guidance with constructivist teachers'. I feel the same can be said for constructivist online environments it terms of the importance of navigation and ease of interaction.

As mentioned earlier I hope that I have exercised a good balance of instructional activities, quizzes that provide an opportunity for formal assessment and discussion forums to invite metacognitive reflection. Interestingly with this unit of work, all students will likely have some prior knowledge to draw on in terms of what makes a good computer game as they have all played computer games before to varying degrees.


Bloom's Digital Taxonomy
Key Terms
HOTS
Higher Order Thinking Skills
Creating
Verbs
Designing, constructing, planning, producing, inventing, devising, making, programming, filming, animating, blogging video blogging, mixing, re-mixing, wiki-ing, publishing, videocasting, podcasting, directing, broadcasting
Evaluating
Verbs
Checking, hypothesising, critiquing, Experimenting, judging, testing, Detecting, Monitoring, blog commenting, reviewing, posting, moderating, collaborating, networking, refactoring, testing.
Analysing
Verbs
Comparing, organising, deconstructing Attributing, outlining, finding, structuring, integrating, mashing, linking, validating, reverse engineering, cracking, media clipping
Applying
Verbs
Implementing, carrying out, using, executing, running, loading, playing operating, hacking, uploading, sharing, editing
Understanding
Verbs
Interpreting, Summarising, inferring, paraphrasing, classifying, comparing, explaining, exemplifying, advanced searches, Boolean searches, blog journaling, twittering, catergorising, tagging, commenting, annotating subscribing.
Remembering
Verbs
Recognising, Listing, Describing, Identifying, Retrieving, Naming, Locating, Finding, bullet pointing, highlighting bookmarking, social networking, social bookmarking, favouriting/local bookmarking, searching, googling.
LOTS
Lower Order Thinking Skills
COMMUNICATION SPECTRUM
Collaborating
Moderating
Negotiating
Debating
Commenting
Net meeting
Skyping
video conferencing
Reviewing
Questioning
Replying
Posting & Blogging
Networking
Contributing
Chatting
e-mailing
Twittering/Microblogging
Instant messaging
Texting

3. Assessment

I have chosen Churches, A. (2009) revised Bloom's taxonomy as a complimentary assessment tool because of its relevancy to digital tools currently used. This has been a good starting point not only for developing formative assessment but also for assisting in the planning and order of lessons. The following diagram is an adaptation of the possible learning processes in relation to this unit of work.



Due to a lack of time in the last pilot, I omitted using the 'questions' component within the 'lessons'. This time around I am endeavouring to incorporate formative testing along the way via questions and quizzes to see if it helps to deepen student's understanding, as suggested Bird, V. (2012, pg. 441) To compliment the quizzes I will intermittently use some 'choice' activities to further evaluate their understanding of the process.

Course Map

Week One: NB: The time allowed for this component will be dependent on student's prior knowledge and their ability to comprehend the new knowledge.			
Achievement Objectives Learning Intentions	**Learning Experiences**	**Assessment of learning intentions based on Bloom's Digital Taxonomy (Highlighted in green)**	**Resources to support learning**
Technological Knowledge	**Introduction to Sequencing and Algorithms**	**Formative Assessment**	**Scratch - Remembering:**
Students will be introduced to the Scratch interface, the Control and Motion buttons. Building a knowledge base to understand the logical process of **sequence** programming Algorithms/ Flow Charts	A key element in understanding programming is to be able to execute a set of instructions to produce a specified behaviour, just like you have a recipe to bake a cake (Brennan & Resnick, 2012, pg.3) **To understand how to Sequence students will learn about;** • The Scratch Interface • Understanding the *x* and *y* Axis • The Blocks Palette • The Control and Motion Blocks • Adding and editing Sprites • Adding and editing Stages **Introduction to Algorithms** • How to write an algorithm that explains your computer game • What are the key symbols • Identify the start, decision, action components within an algorithm.	**Remembering:** Formative Quiz inside Lesson one to evaluate learning acquired. **Revision homework** Study the Scratch concepts and reference guide pages 16 and 10 for a test in Moodle the following week. Evaluation of NZ versus Australia game in discussion forum **Naming:** Students can name and organise sprites, costumes and stage Feedback from homework – Task: Download and install Scratch on your home computer. Practice working with the Control and Motion buttons	Understanding how to manipulate the Controls and Motion buttons to create a sequence to bring about a desired behaviour. Students download the Scratch programme to their own computer (informative letter to parents) Scratch Concepts Guide Scratch Referencing Guide NZ Versus Australia Game **Moodle - Remembering:** 1. Lesson One – Introduction to the Scratch interface 2. Lesson Two – Adding Costumes. Stages and Sprites

Week Two

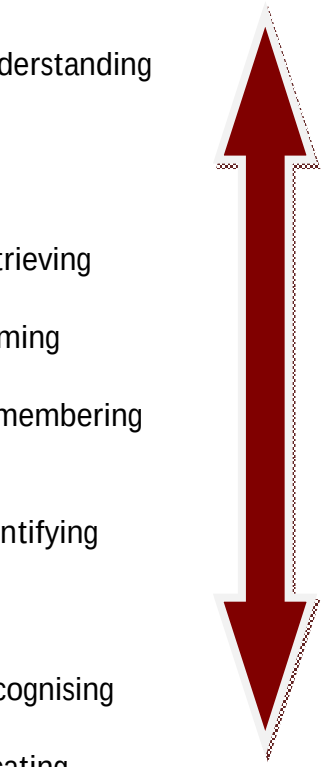
Achievement Objectives Learning Intentions	Learning Experiences	Assessment of learning intentions based on Bloom's Digital Taxonomy (Highlighted in green)	Resources to support learning
Technological Systems	**Creating Events and executing Variables**	**Formative Assessment**	**Resources**
Students will be introduced to the potential of the subsystems present in the Open Source programme Scratch	Students will be introduced to the concept **Events,** an essential component of interactive media (Brennan & Resnick, 2012, pg.4) Students will learn how to make an action as a result of an event or other action. **To understand how to create an 'event' students will learn about;** • The sensing blocks • Global Variables/ Local Variables • The looks block • Boolean logic Student's will add a Sprite to the evaluated game and learn how to write an algorithm for that Sprite. Students will examine an example algorithm and corresponding game	Observations of students participation gathered through Moodle Activity Logs **Locating** Students are able to execute a variable for a predefined purpose (visible in game) **Identifying** Students are able to identify the right blocks to interact with the variable **Remembering:** Quiz and Choice feedback Questions embedded in Moodle lesson **Understanding** Based on prior knowledge from playing computer games Homework – Look at the Scratch website play one of the communities Scratch games and evaluate	**Scratch** Understanding how to create events with the Controls and Motion buttons Scratch Reference Guide http://info.scratch.mit.edu/sites/infoscratch.media.mit.edu/files/file/ScratchReferenceGuide14.pdf **Moodle** 1. Lesson Three – Drawing with Scratch 2. Text answer questions within the lesson 3. Quiz 4. Choice 5. Question Bank **Websites** Scratch.mit.edu Introduce students to the Scratch website.

Week Three: Students are away from regular classes this week for exams so will be set homework			
Achievement Objectives Learning Intentions	**Learning Experiences**	**Assessment of learning intentions based on Bloom's Digital Taxonomy (Highlighted in green)**	**Resources to support learning**
Technological Systems	**Parallelism, Loops, Conditionals**	**Formative Assessment**	**Resources**
Students will begin to understand the importance of organising systems and sub systems in order to execute controls for a desired function	Students will be introduced to the concept of Parallelism which will enable them to exercise control different actions to their sprites simultaneously. Students will learn how to programme two or more sprites at the same time to do different actions. Students will also learn how to make one sprite perform more than one action at the same time. Students will research how to execute a command in Scratch not yet covered in class. Students will then implement that concept and show their work in class forums in the hope that they become a class expert.	Observations of students participation gathered through Moodle Activity Logs Retrieving/ Understanding Students need to upload their game to Moodle and lead a discussion with their peers. (Brennan K., & Resnick, M., (2012, pg.23) Conversations about their work engage young people in a meta-cognitive activity, encouraging them to think about their thinking **Identifying** Students will contribute to the Programming glossary	**Scratch** Students to research exemplar games and discuss **Moodle** Forum Discussions Scratch Glossary Choice **Websites** Youtube Scratch.mit.edu **Worksheets** Scratch Concepts http://learninggameslab.org/documents/program-concepts-day-1.pdf

Week Four and Five:			
Achievement Objectives Learning Intentions	**Learning Experiences**	**Assessment of learning intentions based on Bloom's Digital Taxonomy (Highlighted in green)**	**Resources to support learning**
Technological Practice	**Creating and Evaluating**	**Formative Assessment**	**Resources**
Students will analyse their own and their peer's algorithm in terms of planning and design for their computer game Student's will demonstrate their understanding for planning practices and justify their planning to see the development of an outcome through to completion.	Students will learn how to create a user friendly interface for their game Students will learn how to create an introduction, set of instructions and a 'game over' stage. Understanding how to implement a variable for player input Students will be practising how to write their own algorithm to accompany their Scratch game using key computing terms Debugging, students will troubleshoot script or algorithm problems to exercise efficiency of code Debugging, students will troubleshoot script or algorithm problems to exercise efficiency of code Students will create, explain and evaluate their own game	Student's will upload their game to Moodle for feedback on how the game plays Code is commented with explanations for actions Retrieving/ Understanding Student's will upload their algorithms to Moodle for peer feedback in a Moodle Workshop	**Moodle** Discussion Forum Workshop Glossary of Terms

Rubric for Formative Assessment based on Churches, A. (2009)

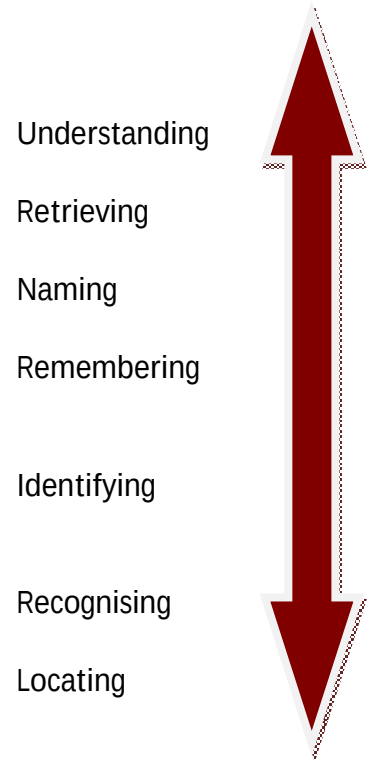
Level	Scratch Game Design with associated Algorithms
4	• Students can create their own game from the beginning • Students can mix and match code from other games • Students can organise scripts by manipulating the blocks to a predetermined outcome • Students can model their game through the discussion forums and help their peers understand • Students can write an algorithm for the whole game along with a mega-algorithm as an overview of the whole programme
3	• Students are evaluating and modifying games to their liking • Students are adding comments to their code to explain • Students are able to write an algorithm for individual sprites • Variable inputs for name, lives, score
2	• Students are able to name Sprites and copy scripts to other Sprites. • Students are able to edit Sprites and background. • Students are able to edit and clean up scripts. • Students are copying scripts from other games to create the same game
1	• Students are able to differentiate between the different types of blocks in the blocks palette and their actions. • Students are able to identify different types of control blocks and develop understanding for how blocks should be used • Students are able to import Sprites and backgrounds. • Students can identify different components of algorithm but do not know how to organise them • Students have not organised scripts, sprites or costumes fit for purpose



Rubric for Formative Assessment based on Churches, A. (2009)

I have created the following rubric from the Summative Assessment AS 91075 by BoiC Head of Department Johnston, B. 2012 and adapted it to Churches (2009) revised Bloom's Digital Taxonomy.

Level	Scratch Game Design with associated Algorithms
4	• Comprehensive checking is evident; the student has modified the algorithm and justified their modifications • The student has completed the algorithmic structure for the task efficiently, using nested structures (including some nested loops and/or nested conditions if required for the task purpose) and complex logical expressions to reduce the length of the algorithm • The algorithm is expressed concisely, simply and clearly. There are no unnecessary statements. • Testing table created and used
3	• Checking is evident; the student has modified the algorithm in response to identified logic flaws • Used iterative structures nested inside other iterative structures to express the solution effectively
2	• Students lists all the inputs that the player will put into the programme, including control of movements and specific data • Variable names are meaningful and well chosen •
1	• Appropriate keywords is used in pseudocode • Student clearly outlines the steps of the game so the algorithm can be easily interpreted • Variables have been used



Evaluation:

I was really happy with the Google Docs Form last time around so I plan to replicate the use of this format again. This time however, I would like to structure my questions to provide more insight to student's learning experience, rather than the design of Moodle's environment. I'm particularly interested to see what level of meta-cognitive thinking student's show in their feedback, which of course will rest on the structuring and authoring of questions. Some ideas I'm exploring are;

- Did any of the activities in Moodle prompt students to engage in a reflection of their own thinking?
- Has there been a shift with any students in terms of their confidence to learn independently and find answers to their solutions to problems?
- What was the level of difficulty experienced within the content?
- Could the content have been explained with more clarity?
- What could have been done differently to improve student's understanding?

References:

Antonenko, P., Toy, S. & Niederhauser, D
Modular Object-Oriented Dynamic Learning Environment: What Open Source Has To Offer

Bird, V. (2012) Southampton Solent University
Using online resources to improve Student outcomes in a level 4 Construction technology unit
Sourced: 24 August 2012 from http://www.icicte.org/Proceedings2012/Papers/11-3-Bird.pdf

Brennan, K., & Resnick, M., AERA 2012
New frameworks for studying and assessing thedevelopment of computational thinking
MIT Media Lab

Churches A., (2009)
Bloom's Digital Taxonomy
Sourced: 24 August 2012 from http://edorigami.wikispaces.com

Masterman, E., Jameson, J., & Walker, S., (2009, pg.224)

Capturing teachers' experience of learning design through case studies
Distance Education; Aug 2009;30,2;ProQuest Vol.30.No.2, August 2009, 223 – 238

Table Glossary References:

Brennan, K., & Resnick, M., AERA 2012
New frameworks for studying and assessing the development of computational thinking
MIT Media Lab

Scratch Concepts
http://learninggameslab.org/documents/program-concepts-day-1.pdf