

A Review of Serious Games for Programming

Michael Miljanovic, Jeremy Bradbury
Software Quality Research Lab
University of Ontario Institute of Technology
Michael.Miljanovic@uoit.ca

November 8th, 2018

Serious Programming Games

RE CODE

RESET LEVEL SWITCH TO JAVA C#

X	EXPECTED RESULT	YOUR RESULT	DESCRIPTION
✗	0	1	Mismatch
✓	-1	0	

@ Well done so far. Look at numbers on line 4 to capture the code.

```

void main()
{
    //this should be commented out since it isn't c++ code!
    int my_number = 2;

    string my_word = "hello there!";

    if (my_number <= 1){
        my_number = 0;
    }
    else{
        my_number = 10;
    }
}
    
```

Tasks:
ACTIVATE the beacons in the right order.
COMMENT out the first that is not C++ code ✓

Available Tools:
[Icons: eraser, copy, paste, undo, redo, run, stop]

Commenter: 999

Time remaining: 962 seconds

TUTORIAL

MAIN METHOD

FUNCTION 1

FUNCTION 2

SYSTEM CLOCK

BREAK

CMDS 4 CALLS 14 CLEAR COMMANDS

◀ MENU WALKTHROUGH ?

code

Original Code Clear Code

```

right

ensure /gidget/:position = /puppy/:position
    
```

world

gidget

- energy: 100
- grabbed: []
- image: "default"
- labeled: true
- layer: 1
- name: "gidget"
- position: [1, 1]
- rotation: 0
- scale: 1
- transparency: 1

It looks like the goal of this level is to move myself to the /puppy/. Use the buttons on the bottom-left to see what my code does, and click on the top-left white panel to start editing!

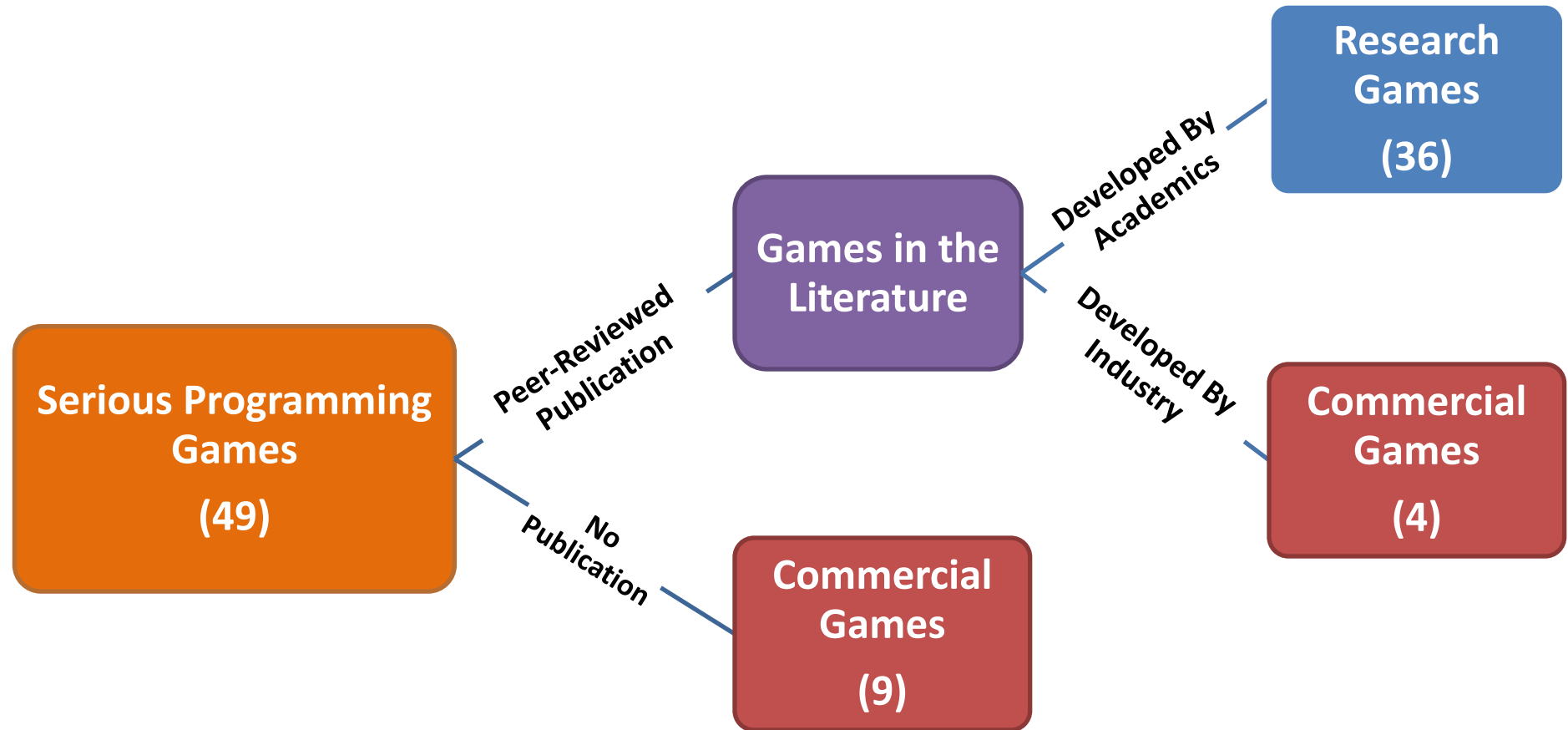
One step One line To end Stop

◀ Prev Next ▶

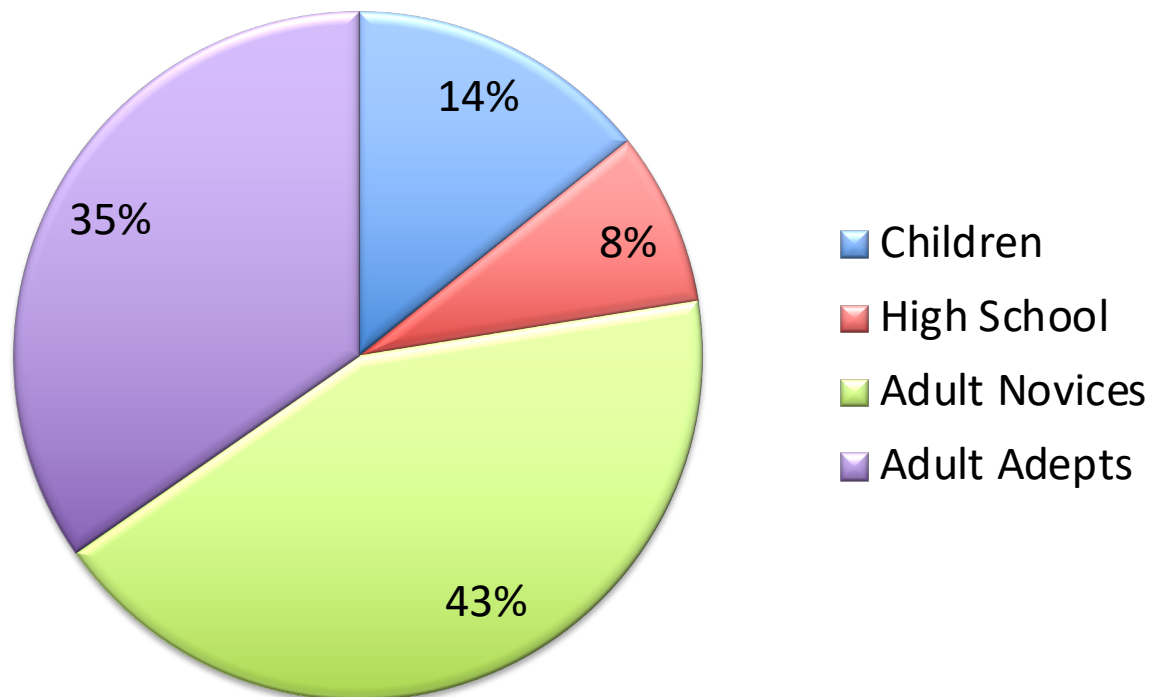
Questions

- What is the **state** of the serious programming game literature?
- What **concepts** are covered in serious programming games?
- What questions are **evaluated** for serious programming games?
 - How are these questions evaluated?

Identification



Audience



Serious Programming Game Content

Algorithms &
Design
(e.g. Problem
Solving)

Fundamental
Programming
Concepts
(e.g.
Conditionals
& iteratives)

Fundamental
Data
Structures
(e.g. Arrays)

Development
Methods
(e.g.
Debugging
strategies)

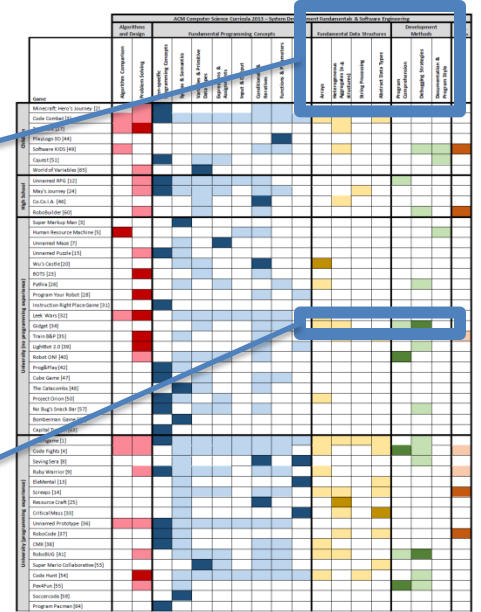
Software
Design
(e.g. Object-
oriented
design)

Software Development Fundamentals

Software
Engineering

Results (Content)

Fundamental Data Structures				Development Methods		
Arrays	Heterogeneous Aggregates (e.g. structures)	String Processing	Abstract Data Types	Program Comprehension	Debugging Strategies	Documentation & Program Style



Results (Evaluation)

Did the users have positive feelings about the game?	Research Questions			
	Did the users have positive feelings about the game?	Was the game accessible?	Were users engaged while playing the game?	Was there a learning effect from playing the game?
Was the game accessible?				
Were users engaged while playing the game?				
Was there a learning effect from playing the game?				

Train B&P [35]			✓	◆
----------------	--	--	---	---

Game	Research Questions				Method of Evaluation				
	Did the users have positive feelings about the game?	Was the game accessible?	Were users engaged while playing the game?	Was there a learning effect from playing the game?	Internal Feedback	Survey/Questionnaire	Formal Interview	API Tests	Game Play Statistics
Children									
TeenTalk [27]									
PlayLogo 3D [44]									
Software KIDS [49]									
Capson [51]									
High School									
Unnamed RPG [12]									
May's Journey [24]									
Co.Co.L.A. [46]									
RoboBuilder [60]									
University for programming experience									
Unnamed Maze [7]									
Unnamed Puzzle [15]									
Wu's Castle [20]									
BOTS [23]									
Pythia [26]									
Program Your Robot [28]									
Instruction Right Place Game [31]									
Gidgit [34]									
Train B&P [35]									
LightBot 2.0 [39]									
Robot OMI [40]									
Prag&Play [42]									
The Capson [48]									
Project Orion [50]									
No Bug's Snack Bar [57]									
Capital Tycoon [63]									
Saving Sea [6]									
EleMontal [13]									
Resource Craft [15]									
Critical Mass [33]									
University for programming experience									
Unnamed Prototype [36]									
RoboCode [37]									
CMS [38]									
RoboBAG [41]									
Code Hunt [54]									
PuzzleFun [55]									
Succorcode [59]									
Program Pacman [64]									

Results (Evaluation)

Informal Feedback	Method of Evaluation					
Survey/Questionnaire	Informal Feedback	Survey/Questionnaire	Formal Interview	Skill Tests	Game Play Statistics	Expert Feedback
Formal Interview						
Skill Tests						
Game Play Statistics						
Expert Feedback						

Train B&P [35]		●			●	
----------------	--	---	--	--	---	--

Game	Research Questions					Method of Evaluation					
	Did they have positive feelings about the game?	Was the game accessible?	Were users engaged while playing the game?	Was there a learning effect from playing the game?		Informal Feedback	Survey/Questionnaire	Formal Interview	Skill Tests	Game Play Statistics	Expert Feedback
Children											
TeenTalk [27]											
PlayLogo 3D [44]											
Software R105 [49]											
Cipact [51]											
High School											
Unnamed RPG [12]											
May's Journey [24]											
Co-CoLA [46]											
RoboBuilder [60]											
University (Programming Experience)											
Unnamed Maze [7]											
Unnamed Puzzle [15]											
Wu's Castle [20]											
BOTS [23]											
Pythia [26]											
Program Your Robot [28]											
Instruction Right Place Game [31]											
Gidget [34]											
Train B&P [35]											
Lighthouse 2.0 [39]											
Robot OH! [40]											
ProgePlay [42]											
The Catacombs [48]											
Project Orion [50]											
No Bug's Snack [52]											
Capitales [53]											
University (Programming Experience)											
Building Sea [8]											
ElleMontal [13]											
Resource Craft [25]											
Critical Mass [33]											
Unnamed Prototype [36]											
RoboCode [37]											
CMX [38]											
RoboBUG [41]											
Code Hunt [54]											
PuzzleFun [55]											
Succorcode [59]											
Program Pacman [64]											

Challenges

- It is difficult to conduct cross-game **comparisons**
- Few games received a comprehensive evaluation
- Testing **practices** for **commercially** developed games are not known

Evaluation of Serious Programming Games

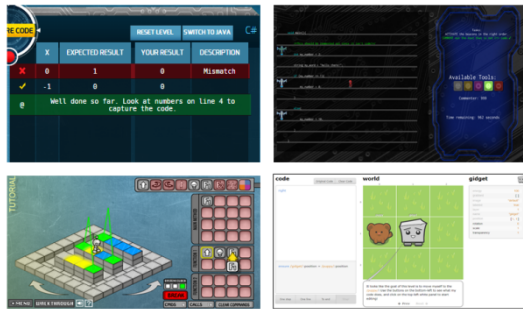
- **Learning efficacy** of serious programming games is not well understood
 - Lack of standardized evaluation methodology
 - Few studies include empirical data for reproducibility
- Issue of **availability** limits the opportunity for further evaluation
 - Only **half** of the games in the survey can actually be played or downloaded online!

Conclusion

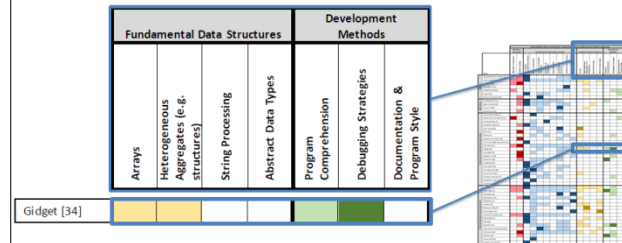
- It is unclear which **curricula gaps** can be bridged by serious programming games
- Best practices for **accessible and inclusive design** of serious programming games should be adopted
- No clear **best practices for evaluation** have been established
- There is a need for **third-party evaluations** of serious programming games

Review

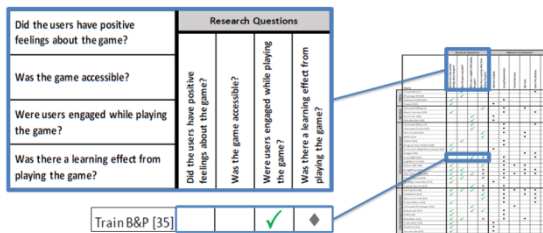
Serious Programming Games



Results (Content)



Results (Evaluation)



Conclusion

- It is unclear which **curricula gaps** can be bridged by programming games
- Best practices for **accessible and inclusive design** of programming games should be adopted
- No clear **best practices for evaluation** have been established
- There is a need for **third-party evaluations** of programming games

A Review of Serious Games for Programming

Michael Miljanovic, Jeremy Bradbury
Software Quality Research Lab
University of Ontario Institute of Technology
Michael.Miljanovic@uoit.ca

November 8th, 2018