Coding Phase – ICS4U0 – Unit 6 Final Project –Self-Evaluation

Coder Name: _____ Game Name: _____

Some coding metrics:

Total Lines of XML code: ____ Total Lines of Java Code: ____ Total Cards: ____ Total methods: ____ Objects: ____ Extra Features: ____

Application: A/C/D. Program Constructs Predicted Mark: . . . . . %

Curriculum	Criteria: Use Count to Find Level	R	1	2	3	4	4+
A1. Data Types and Expressions	❑ File Read: Line? . . . . . ❑ 2D array: How many? . . . . . ❑ File Write: Line? . . . . . ❑ Stack: How many? . . . . . ❑ 1D array: How many? . . . . . ❑ Queue: How many? . . . . . ❑ String Functions: ❑compareTo, ❑equals, ❑indexOf ❑ getText ❑ Type conversion: ❑(int) ❑(Button) Android findViewByID, Line: . . . . . ❑Integer.parseInt, ❑Double.parseDouble	-	1	2 -	4	5 +	6 +
A2. Modular Programs	❑ Intents and multiple screens used ❑ Widgets/Views used (encapsulation, sets, gets) ❑ Panels/ViewGroups used (method overloading) *MainActivity Methods (not in Objects or ADTS):* ❑ Own methods (not onClick): How many? . . . . . Name? . . . . . ❑ Own parameters (not View view): How many? . . . . . Name? . . . . . ❑ Own return types (not void): How many? . . . . . Name?	-	1	2	3	4	5
C1. OOP & Modular Design	❑ Object: Name: . . . . . ❑ Another Object: Name: . . . . . ❑ Yet Another Object: Name: . . . . . ❑ ADT of Objects: Name: . . . . . *Encapsulation* ❑ Constructors (default and editable) in own object: Line: . . . . . ❑ Accessors, including toString, in own object: Line: . . . . . ❑ Mutators in own object: Line: . . . . . ❑ CompareTo, equals in own object: Line: . . . . . ❑ Facilitator: What?	-	3	6	7	8	9 +
A3. Algorithms/ C2. Algorithm Analysis	❑ Push, Enqueue: onClick? . . . . . ❑ Search (Win): Method? . . . . . ❑ Pop, Dequeue: onClick? . . . . . ❑ Sort Algorithm: Method? . . . . . ❑ ShowCard: onClick? . . . . . ❑ Recursion: Method? . . . . . ❑ Score Keeping: Method? . . . . . ❑ Complex: Method? . . . . . ❑ Reset button: Screen? . . . . . ❑ Other? . . . . . ❑ Save/Open: onClick? . . . . . ❑ Other?	-	2	4	5	6	7
D3. Emerging Tech/ D4. Exploring Computer Science	❑ Emerging Tech: App Creation ❑ XML Parsing: Name of an XML file? . . . . . ❑ Image Processing: Complex photoshop or formatting: What? . . . . . ❑ Dictionary File: Method? . . . . . ❑ Logic: Complex calculations: Method? . . . . . ❑ Recursion: complex code that Calls itself: Method? . . . . . ❑ Artificial Intelligence: Method? . . . . . ❑ Extra feature: ❑New widget/view: Name all: . . . . . , ❑Timer, ❑Animation, ❑Other:	-	2 -	3	4 +	5	6

Communication: A. Code Style Predicted Mark: . . . . . %

Curriculum	Criteria: Use Count to Find Level	R	1	2	3	4	4+
A4. Code Maintenance	❑ 3 Title comments ❑ Comments before every method ❑ Comments before major loops and ifs ❑ No huge sections of commented out code ❑ No huge sections of random blanks ❑ Comments continue right to the end ❑ Code is indented properly (just use the menu option!!) ❑ Levels / new games do not require new screens ❑ No method is more than 30 lines	-	3	4	5	6	8
A4. Instructions/ User Help	❑ Multiple Screens, can move between them ❑ Look of screens consistent across screens. ❑ One colour scheme on all screens. What is it? . . . . . ❑ Help screen has instructions ❑ Help screen has more than one picture ❑ Help screen has more than one screen snapshot of the actual game ❑ Help screen has high quality instructions, more than a few lines. ❑ Game has a lot of error handling built in ❑ Game has toasts, dialog boxes or on-screen feedback ❑ Turns or score are clearly indicated on the screen	-	3	4	5	6	8

Thinking: B. Software Development Life Cycle Predicted Mark: . . . . . %

Curriculum	Criteria: Use Count to Find Level	R	1	2	3	4	4+
B1. Project Plan & Design	❑ Analysis Google Form Submitted ❑ UML is Submitted + Card printout ❑ UML is detailed ❑ Memory diagram is submitted ❑ Memory diagram is detailed ❑ Screen design is submitted ❑ Screen design has dimensions for grid squares ❑ Screen design is very detailed ❑ Instructions printout submitted ❑ Instructions very detailed, text and pictures are outlined.	-	2	5	7	8	10
B1. Follows Timelines/ B2. Meets Goals	❑ Card pictures created on Schedule – June 5 ❑ Card Object Up on Schedule – June 6 ❑ Game Screen Up on Schedule – June 7 ❑ Push/Pop Complete on Schedule – June 7 ❑ Score on Schedule – June 11 ❑ Game working before Code Freeze – June 12 ❑ Extra Features Added. What?	-	2	3	4	6	7
B1. Testing	❑ Code ready for Alpha testing - June 7 ❑ Alpha tested someone else's ❑ You wrote good Alpha test comments ❑ Someone else Alpha tested yours ❑ Code ready for Beta testing – June 11 ❑ Beta tested someone else's ❑ You wrote good Beta test comments ❑ Someone else Beta tested yours	-	2	3	5	6	8
B1. Close Project	❑ On schedule – June 14 ❑ Logo on App ❑ apk file works or is working on Gorski's phone ❑ App is named well on Gorski's phone. ❑ Word File has correct name: LastLastLastGame.docx ❑ Word File saved to correct location ❑ Word File has screen snapshots submitted ❑ Word File has clear, resized, cropped screenshots submitted ❑ Word file has all code cut and pasted into it – XML & Java ❑ Line count calculated correctly	-	3	5	6	8	10
B2. Reflection	❑ Program Constructs check-brick done ❑ Style check-brick done ❑ SDLC check-brick done ❑ Predicted mark for each check-brick done. That's 3 predicted grades. ❑ Individual evaluations are completed for each group member. ❑ Written questions answered on forms. ❑ Written questions answered well; not a one word or short answer. ❑ Reflection (individuals + this sheet) submitted by due date – June 14	-	2	4	5	6	7

List your extra-features:
