

A303	Level 4	Level 3	Level 2	Level 1	PyGame #
communication		Comments - useful succinct, (block OK), not sparse, block at top (modifications, start date, programmer name, problem description, Notes on anything special or unusual, citations)	lacking explanation, inline comments, or hard to read, or spelling mistakes on output or Programmer missing or modifications	Any two or more details missing or Citations/borrowed work not acknowledged OR more than 25% of code borrowed	
Knowledge	Excellent User instructions that are clear and can be accessed at any time or Additional options, like premature quit overloading or similar more advanced coding	Readability - code is easy to follow, enough white space user interface, clear language, no grammar or spelling errors Effectively used classes and inheritance	Adequate 2 or 3 grammar or spelling errors used but not to best advantage or created cumbersome usage	Needs improvement User interface not clear, poor language and or spelling and grammar errors hard coded words and or excessive repetition	
Knowledge	colouring to add to understanding, input one colour/output another	Flowchart - organized, proper spacing, correct symbols, proper language	Poor spacing, some incorrect symbols, typing errors,	Poor organization, many symbols used incorrectly for the concept attempting to convey, arrows in the wrong section or incorrect pointer	
Thinking & Inquiry	Description at top to clarify	IPO - proper chart (headings), steps, proper terminology	Unclear steps or incorrect terminology	missing information in columns or steps missing	
Thinking & Inquiry	Screen sizes to user systems. Make it interactive. Colour or design helps user know what stage they are at.	Graphics are appropriate and professional looking for Yatze Blommas - word list appropriate components - Standard screen size 1280 x 1024	Small graphics, features not intuitive	No graphics, just console that user would not necessarily have. Lost count on number of guesses	
		Trace statements - variable name, steps and changes (Table/chart)	Variable names don't match or missed a step	more confusing, complicate the existing error, or do not recognize it.	
Application	Could be class based Highly efficient, best possible, perhaps found a new function or way to complete the task	All functions (including user) are efficient and work as expected Code efficiency - nested if's better into function or class.	Functions work, but confusing or inefficient Nested if's or lack of proper loops - does not take advantage of built in functions	Hard coded where a user defined function or built in would suffice	
Flowchart/IPO/Trace Table (Application)	colouring to add to understanding, input one colour/output another	Flowchart - organized, proper spacing, correct symbols, proper language	Poor spacing, some incorrect symbols, typing errors,	No functions or classes poor organizations, many symbols used incorrectly for the concept attempting to convey, arrows in the wrong section or incorrect pointer.	

90 MS Die Inherit 6020 ropper Calls ✓ One Round / Game PyGame
 Flow Colour - ✓ Trace - Version: IPO ✓
 Block @ Top Received Sure for unwanted to s/b | scored 11 / 100