Who: Sean Hansberry, Michael Crozier, Scott Schubert, Jonathan Schmitt

Title: 20 Questions

Vision: Create an entertaining game with flexible architecture that allows for extensive subject choice by the player.

Automated Test Cases: Run testProg.py to verify the program is running correctly. This test file should return True for all cases in which the program is working correctly and False if there are any runtime errors.

User Acceptance Testing:

Test Case ID:	101	Test Designed by:	Scott Schubert
Test Priority(L/M/H)	Low	Test Designed Date:	11/10/2015
Module Name:	Incorrect input	Test Executed by:	N/A
Test Title:	Recognize invalid input and reask the question until correct input received	Test Execution Date:	TBA

Description:	Test the main part of the game to respond to invalid input
Pre-conditions:	User enters "y", "n", "yes", "no", "nay", "indubitably" during the main part of the game.
Dependencies:	main.cpp, Animals.txt, Animals.cpp

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Run the program from terminal	N/A	Program begins execution	Program begins execution	Pass	
2	Enter any input besides approved input above	N/A	"Invalid response" Repeats question	"Invalid response" Repeats question	Pass	

Test Case ID:	102	Test Designed by:	N/A
Test Priority(L/M/H)	H	Test Designed Date:	11/10/15
Module Name:	Animal constructor verification	Test Executed by:	N/A
Test Title:	Create an animal object from Animals.txt	Test Execution Date:	TBA

Description:	Verify that animal objects are being created and stored in main.c from Animals.txt
Pre-conditions :	Animals.txt has data
Dependencies:	main.c, Animals.txt, Animals.cpp

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/ Fail)	Notes
1	Run main	N/A	Asks first question	Asks first question	Pass	
2	Initialize Animals	Ant,1,0,0,0,0,0 ,0,0,0,0,0,0,0, 1,0,0,1,0,0,0	Create an Animal Struct called "Ant"	Animal Struct "Ant" created	Pass	

Test Case ID:	103	Test Designed by:	N/A
Test	High	Test Designed Date:	11/11/15

Priority(L/M/H)			
Module Name:	Question Count	Test Executed by:	N/A
Test Title:	20 questions being asked	Test Execution Date:	TBA

Description:	Verify that 20 questions are being asked and that the program is not stuck in a question loop
Pre-conditions:	User enters y/n for each question
Dependencies:	main.c

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Run main	N/A	Asks first question	Asks first question	Pass	
2	Accept user input for first question	Yes	Logs yes for first question	Logs answer "Yes"	Pass	
3	Accept user input for next 19 questions	Y,Y,Y,Y,Y,Y,Y,Y,Y,Y,Y,Y,Y,Y,Y,Y,Y,Y,Y	Logs answers for each question, returns closest Animal match	Logs answers for question does not return Animal match	Fail	

VCS: https://github.com/Mcrozier5609/20-Questions