SmallSEOTools

PLAGIARISM SCAN REPORT

Words	992	Date	April 21,2019
Characters	6333	Exclude Url	

0%	100%	0	35
Plagiarism	Unique	Plagiarized Sentences	Unique Sentences

Content Checked For Plagiarism

Chapter 5 Implementation and Testing 5.1 Implementation Approaches Key press logic code: if event.type == pygame.KEYDOWN: if event.key == pygame.K_SPACE: stop_all() press_space.play() message_display('you have pressed SPACE key',20,dw/2,dh/2,blue) e=e+1 elif event.key == pygame.K_UP: stop_all() press_up.play() message_display('you have pressed UP key',20,dw/2,dh/2,blue) a=a+1 elif event.key == pygame.K_DOWN: stop_all() press_down.play() message_display('you have pressed DOWN key',20,dw/2,dh/2,blue) b=b+1 elif event.key == pygame.K_LEFT: stop_all() press_left.play() message_display('you have pressed LEFT key',20,dw/2,dh/2,blue) c=c+1 elif event.key == pygame.K_RIGHT: stop_all() press_right.play() message_display('you have pressed RIGHT key',20,dw/2,dh/2,blue) d=d+1 elif event.key != pygame.K_UP or pygame.K_LEFT or pygame.K_DOWN or pygame.K_RIGHT or pygame.K_SPACE: stop_all() press_wrong.play() message_display('you have pressed WRONG key',20,dw/2,dh/2,blue) else: game_loop() Menu Screen code: def menu(): stop_all() menu_options.play() gameDisplay.fill(white) message_display('WHITE CANE',33,dw/2,dh/10,red) message_display('** MENU **',20,dw/2,dh/10*2,green) message_display('for HOME press SPACE',20,dw/2,dh/10*3,blue) message_display('to play again press UP key',20,dw/2,dh/10*4,blue) message_display('to know about developers DOWN key',20,dw/2,dh/10*5,blue) while True: for event in pygame.event.get(): if event.type == pygame.QUIT: pygame.quit() quit() if event.type == pygame.KEYDOWN: if event.key == pygame.K_SPACE: intro() if event.key == pygame.K_UP: game_loop() if event.key == pygame.K_DOWN: developer() else: menu() 5.2 Coding Details and Coding Efficiency 5.2.1 Code Efficiency Our project contains not more than 1k line of code. In our project we have imported approximate 100 number of voice files. Here we are using the cocomo model which is based on regression model that is based on line of code. PM = 3.0*(SIZE)*1.2 Thus the required equation is 3.0*(1)*1.2= 3.6kloc 5.2.2 Coding Details The application uses simple logic and it has more than 70 files which include backup, code, image, audio, icon, font, etc. We have used loops in coding we have defined the different functions so that we can avoid re writing the codes. We have to run only single code to run the application so we do not required any kind of arrangement so we have sorted files according to their name and types. As you can see the arrangement of the files. Fig 5.2.1 Code & Files Structure 5.3 Testing Approach Test case 1 - To verify the structure of the application and how it supports our device. [PASS] Test case 2 - To check module 1 is working correctly or not . [PASS] Test case 3 - To verify whether module 1 is not crashing or running without any error. [PASS] Test case 4 - To check module 2 is working correctly or not. [PASS] Test case 5 - To verify whether module 2 is not crashing or running without any error. [PASS] Test case 6 - To check module 3 is working correctly or not. [PASS] Test case 7- To verify whether module 3 is not crashing or running without any error. [PASS] Test case 8 - To check module 4 is working correctly or not . [PASS] Test case 9 - To verify whether module 4 is not crashing or running without any error. [PASS] Test case 10 – To check whether event handling is working properly or not. [PASS] Test case 11 – To check whether used image is fitting in size or not. [PASS] Test case 12- To check the click event for game module event 5.3.1 For Right key Key press o/p(event) Pass/Fail/no result Right key Move to right (70px) Pass Other key press Nothing happened Fail Nothing pressed Waiting state No result 5.3.2 For LEFT key Key press o/p(event) Pass/Fail/no result LEFT key Move to left (70px) Pass Other key press Nothing happened Fail Nothing pressed Waiting state No result 5.3.3 For UP key Key press o/p(event) Pass/Fail/no result UP key Increase in speed(0.2%) Pass Other key

press Nothing happened Fail Nothing pressed Waiting state No result 5.3.4 For DOWN key Key press o/p(event) Pass/Fail/no result DOWN key Decrease in speed(0.2%) Pass Other key press Nothing happened Fail Nothing pressed Waiting state No result 5.3.5 For SPACE key Key press o/p(event) Pass/Fail/no result SPACE key Pause the game Pass Other key press Nothing happened Fail Nothing pressed Waiting state No result 5.3.1 Integration and System Testing In this chapter we are going to see the possible way we can develop the program through using

System Testing In this chapter we are going to see the possible way we can develop the program through using better infrastructure and we are going to see that how can we upgrade our project to the next level. 5.3.2 Introduction In software testing we have to build the test level on from the previous level so it is important that so that we can do testing in correct order without lacking any steps, after completion of level we Can go to the next level for testing. 5.3.3 Integration Integration and system testing is a type of software testing, this type of testing is done just before releasing the product. Software testing follows very ideal set of test from which we can make sure that the project that is to be released is perfect and error less. Integration and system testing is basically done within the set of people/group who has one work to do that is to test only in the system development life cycle. As the name suggest it deals with small component not with whole software. There are three main things which is to be checked for running the test successfully the first is a test plan second is a test cases and the third one is test data is produced to effectively test that the integration is successful. The test data is one of the part of integration testing. 5.3.4 Integration Test Plan After the implementation of product testing this things to be consider. 1) A strategy to use when testing the integrated modules and how the tests will be conducted. 2) What will be tested for example software features. 3) What is the time scale and time management? 4) Responsibilities, e.g. personnel. 5) Testing pass and fail condition. 6) Risk involved 7) Approval from all important people involved. 5.3.5 Critical Path Method (CPM) We can take following mention things as input. 1. A priority chart whit schedule. 2. Estimates of task time management. 3. A written work of each task to a programmer. 4. Set of order of module flow each programmer.

Sources	Similarity