Test Plan

Alex Guerrero, Keyur Patel and Shafeeq Rabbani

October 23, 2015

Contents

1 Introduction

The test plan is designed to identify the types of tests to perform and helps explain how tests will be performed.

1.1 Test Items

The different items to be tested includes:

- A: The functions and methods of each class of the Model (backend)
- B: The game board against the functional requirements of the product
- C: The graphical interface that implements the Model

2 Software Risk Issues

3 Features to be Tested

4 Features not to be Tested

5 Testing Types

Testing can be broken up into different types, which each have their own role in the testing the product. These test types should be utilized to comprehensively evaluate the quality of the product.

5.1 Structural Testing

Structural testing is also known as white box testing. Structural tests are derived from the program's internal structure. It focuses on the nonfunctional requirements of the product. This type of testing shows errors that occur during the implementation by focusing on abnormal and extreme cases the product could encounter.

5.2 Functional Testing

Functional testing is also known as black box testing. Functional tests are derived from the functional requirements of the program. It focuses less on how the program works and more on the output of the system. These tests are focused on test cases where the product receives expected information.

5.3 Static vs. Dynamic Testing

Static testing simulate the dynamic environment and does not focus on code exectution. This testing involves code walkthroughs and requirements walkthroughs. Static testing is used prevalently in the design stage. In contrast, dynamic testing needs code to be executed.

Dynamic testing involves test cases to be run and checked against expected outcomes. A technique to save time during dynamic testing is to choose representative test cases.

5.4 Manual vs. Automatic Testing

Manual testing is done by people. It involves code walkthroughs and inspection.

Automatic testing can usually be conducted by computers. The tools used to assist with automatic are unit testing tools for the respective programming language. Automatic testing relies on people for testing more qualitative aspects like GUI.

6 Approach

6.1 Test Cases for Snake.py

Table 1: Snake.py

Method	Input	Expected Outcome
constructor	none	first 20 points of the snake are generated
changeDir	dir=-1	if current direction is 2 or -2, it will be updated to -1

6.2 Test Cases for Map.py

6.3 Test Cases for Food.py

6.4 Testing for GUI

References