

# Test Plan

Alex Guerrero, Keyur Patel and Shafeeq Rabbani

October 22, 2015

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Test Items . . . . .	2
<b>2</b>	<b>Software Risk Issues</b>	<b>2</b>
<b>3</b>	<b>Features to be Tested</b>	<b>2</b>
<b>4</b>	<b>Features not to be Tested</b>	<b>2</b>
<b>5</b>	<b>Testing Types</b>	<b>2</b>
5.1	Structural Testing . . . . .	2
5.2	Functional Testing . . . . .	2
5.3	Unit Testing . . . . .	2
5.4	Static vs. Dynamic Testing . . . . .	2
5.5	Manual vs. Automatic Testing . . . . .	2
<b>6</b>	<b>Approach</b>	<b>2</b>
6.1	Test Cases for Snake.py . . . . .	2
6.2	Test Cases for Map.py . . . . .	3
6.3	Test Cases for Food.py . . . . .	3
6.4	Testing for GUI . . . . .	3

# **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**

### **5.2 Functional Testing**

### **5.3 Unit Testing**

### **5.4 Static vs. Dynamic Testing**

### **5.5 Manual vs. Automatic Testing**

## **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