Testing skeleton

Game unit tests:

```
func test_create_player():
    assert_eq("Player 1", player.create_player("Player 1"))

func test_select_card():
    assert_eq(true, player.select_card("Apple"), "Should be true")

func test_card_pair_match():
    assert_eq(player.select_card("Apple"), player.select_card("Apple"))

func test_card_pair_match():
    assert_eq(player.select_card("Apple"), player.select_card("Apple"))

func test_tie_game():
    ...

func test_game_winner():
    ...

func test_get_card_description():
    ...

func test_card_grid_generation():
    ...
```

Payment system unit tests:

```
describe('makePayment()', function() {

      . . .

});

describe('invalidCredentials()', function() {

      . . .

});



describe('getPaymentAmount()', function() {

      . . .

});

describe('confirmationEmail()', function() {

      . . .

});

describe('selectCharityOrganization()', function() {

      . . .

});
```

Web interface unit tests:

```
<!-- Ensure game is properly linked to main website -->

<a href="game.html" class="list-group-item list-group-item-action
bg-light">Game</a>
```

```
<!-- Ensure payment system is properly linked to main website -->

<a href="payment-system.html" class="list-group-item list-group-item-action
bg-light">Payment System</a>
```

```
// test browser sync functionality

describe('browserSync()', function() {

...

});
```

```
// test the loading of dependency modules

describe('modules()', function() {

...

});
```

Game integration tests:

```
extends "res://addons/gut/test.gd"
class TestPlayerClass:
	extends "res://addons/gut/test.gd"
	func before_all():
		gut.p("----Player Integration Test----")
	.
	.
	.

	func after_all():
		gut.p("--------------------------------")

extends "res://addons/gut/test.gd"
class TestCardClass:
	extends "res://addons/gut/test.gd"
	func before_all():
		gut.p("----Card Integration Test----")
	.
	.
	.

	func after_all():
		gut.p("-----------------------------")

extends "res://addons/gut/test.gd"
class CardGridClass:
	extends "res://addons/gut/test.gd"
	func before_all():
		gut.p("----Card Integration Test----")
	.
	.
	.

	func after_all():
		gut.p("-----------------------------")
```

```
# Big bang approach: load multiple files and make sure they all pass
extends SceneTree
func _init():
    load('res://scripts/unittest.gd').run([
        'res://scripts/player_unit_tests.gd',
        'res://scripts/card_unit_tests.gd',
        'res://scripts/game_unit_tests.gd'
    ])
    quit()
```

Payment system integration test:

```
// Run all components in the payment system with big bang approach
describe('Payment System Tests', function() {
  var task = {
    name: 'integration test'
      ...
  };
```

Web interface integration test:

```
// Run all components in the web interface with big bang approach
describe('Web Interface Tests', function() {
  var task = {
    name: 'integration test'
      ...
  };
```

Integration test (for all):
Open the ‘index.html’ file for the project once each subsystem has had their integration testing. This should string together every component of the project. We can go through each page to ensure they are working properly.

Regression test procedure:
Highest priority at the top, lowest priorities at the bottom.