# Project Test Plan

|  |  |  |
| --- | --- | --- |
| Feature | Description | Complete Date |
| Problem Statement | This document provides a brief description of our project, including the game rules. | 3/15/2013 |
| Game State | Users can decide the time limit for the game or they can decide to turn the timer off altogether. This also includes objects such as the timer and score, as well as logic for starting a new game. | 3/22/2013 |
| Pokemon Evolutions | When the user makes various plays, the pokemon may evolve into game pieces with special effects. | 3/22/2013 |
| Pokemon Grid Check Board Algorithm - Column or Row of 3 | This is the most basic implementation of the updating the game board, accounting for rows and columns of only three tokens of the same type in a row. It does not actually update the board, but simply marks rows and columns of three as null to be updated later. | 3/22/2013 |
| Pokemon Tokens | These are the different game pieces for the game. | 3/22/2013 |
| Basic Game GUI | This is the basic layout for the game GUI, without interactive functionality. Make sure text is loaded from the correct file to account for locale. | 4/12/2013 |
| Refactor Check Board Algorithm |  | 4/12/2013 |
| Update Board Algorithm | Given a current board state, this updates the board, causing the tokens to fall to fill empty spots and generating new tokens when necessary. | 4/12/2013 |
| GUI Interaction | The user will be able to interact with the GUI. | 4/19/2013 |
| Pokemon Grid Check Board Algorithm - Column or Row of 5 | This checks the board for rows or columns of 5, and marks nulls accordingly. This also adds a ditto to the board. | 4/19/2013 |
| Pokemon Grid Check Board Algorithm - Column or Row of 6 | This checks the board for rows or columns of 6, and marks nulls accordingly. This also adds a 2nd evolution pokemon token to the board. | 4/19/2013 |
| Pokemon Grid Check Board Algorithm- Column or Row of 4 | This checks the board for rows or columns of 4, and marks nulls accordingly. This also adds a 1st evolution pokemon token to the board. | 4/19/2013 |
| Pokemon Grid Check Board Algorithm - 1st Evolution Swap | This checks the board to see if a 1st evolution pokemon token is swapped, and marks nulls accordingly. | 4/26/2013 |
| Pokemon Grid Check Board Algorithm - 2nd Evolution Swap | This checks the board to see if a 2nd evolution pokemon token is swapped, and marks nulls accordingly. | 4/26/2013 |
| Pokemon Grid Check Board Algorithm - Ditto Swap | This checks the board to see if a ditto was swapped, and marks nulls accordingly. | 4/26/2013 |

# Coding Standards

We will be using [C# coding standards](ClientTeam-CodingStandards.pdf) from Interactive Intelligence.

# Code Coverage

We plan to use DotCover.

# Architecture

The [domain class diagram](../PokemonBejeweled/PokemonBejeweled/DomainClassDiagram.cd) is already present on our git repo, and will stay current as we update our project.

# User Scenario

Milestone 3

We plan on completing a basic GUI, implementing the algorithm for updating the board, and refactoring the check board algorithm - column or Row of 3. The user will be able to start a game and see the first board generated. They will also be able to start a new game and change the settings.

We will test the algorithm for updating the board when there are rows or columns of three pokemon missing, following the same scenarios used for testing the check board algorithm. At this time, we will not be attempting to test the GUI due to the complications that that introduces.

Milestone 4

We plan on completing the connection to the GUI, creating more test cases for the game board, and finishing the game board algorithm. The user should be able to choose two of the game pieces, choose how long they want to play and quit the game. We will also fix the look of the game board so that the game pieces have the appropriate pictures.