**Objective:**

Project Jabberwocky is an educational integrated development environment (IDE) that evolves as a user programs in the environment. Jabberwocky starts out as a simple editor with only a few basic commands but as a user programs in the environment advanced features are unlocked. By slowly introducing new features, new programmers are not overwhelmed and they cannot use more advanced features as a crutch during learning. By the time the user has unlocked all the features they are left with a full-featured IDE.

**End Users:**

A Programmer: This is the main user behind the application. They are the ones who are learning to program and will be traversing the levels of the IDE.

**Use Cases:**

User writes a program.

Actors: The Programmer

Precondition: A source code editor is presented to the user.

Trigger action:

The user creates a new source file.

User advances a level.

Actors: The Program

Precondition:

A user has a profile storing their current status.

A user has just compiled their program.

Trigger action:

The user has successfully compiled enough programs to demonstrate competence.

User compiles/runs a program.

Actors: The Programmer

Precondition:

A user has a complete program with no syntax errors.

Trigger action:

The user hits the run button.

**Workflow:**

User writes a program

User types in text.

Editor records text.

Editor checks syntax.

User advances a level

Editor has program compiled.

Editor checks program.

Editor checks users profile for level status, determines if they can move on.

User compiles a program.

User hits build button.

Editor uses syntax checker.

Editor runs the program.

**Features:**

User writes a program

Editor must work as a normal editor for users.

Advanced features should be unavailable until they get unlocked.

User advances a level

A select set of features should be unlocked that are appropriate to their skill level.

Functional Components:

Algorithm to determine speed of leveling.

Modules that are composed of features to be unlocked.

Non Functional Components:

User must be able to edit programs from scratch without being obstructed.

Levels must be unlocked at a point in time when users are ready to move on.

Advanced functionality must not be present for new users and must unlock later on.