Prodigy (working title)

**Prodigy is an online tutorial system designed to motivate young students of second languages. It is a powerful tool for teachers to present plain class content in a fun and creative way.**

**Intro**

Say you are a student of psychology (in the real world) and, while it’s nothing to be ashamed of, your English is, less than perfect. So, like the brave do, you decide to enroll to some public English school. In that school, you learn. Or at least you’re supposed to. How do you learn? Depends on you. How are you taught or guided? That depends on the methodology the school follows, but no less important are the people around you – friends and teachers.

You’ve been there for quite some time and made little progress (which is perfectly fine). One day you come across a fellow student. You remember to things about her: she used to be much heavier than she is now, and her English was extremely poor. You chat for a while, and then you notice she’s not to bad (as far as her English goes). You immediately stop her mid speech, and ask:

“What’s your secret?”

“I just practice” she replies.

Now it hits you (you’re a fast learner…), practice is key. You have hardly done any sort of practical exercises. They are an absolute pain the butt (you say to yourself). Well good news, you’re not alone. And don’t beat yourself up about it; most of them are not very fun. And yes, practice absolutely should be.

So why do most practical exercises share this boring nature? One answer – unimaginative content creators. The first common trade great activities share is that they are centered on one main subject learned. Students need to know what it is they are practicing – ambiguity is horrible. The second is presentation. Both are at full control by the teacher. While the subject might be dangerously boring (grammar rules…), it certainly doesn’t have to be presented as such.

**Vision**

An online environment where students practice and learn, engulfed by a story of a young Prodigy on his quest to conquer a field of interest. This is what prodigy is all about. Making study and practice fun.

//TODO: complete vision artifact

Glossary

Question – a multiple choice problem.

Section – a subject-matter of practice.

Level – a group of questions. Has a name, a description and belongs to one subject (zone).

Choice Tokens – one or more unique textual content from which answers are built.

Answer – a sequence of choice tokens made from those available to a question.

Solution – the set of all correct answers to a question.

Question Feedback – a textual feedback that is assigned to an answer.

**Question Design**

A question holds what we refer to as a headline. The headline is a sentence that creates context to the instructions of the exercise. The exercise’s instructions provide a lead-in sentence, which requires you to create an answer, based on the context of each question – that is the headline. For example:

* Instructions: *Put in to the plural*
* A question:
  + Headline: **A boy has a ball**
  + Answer: *Boys have balls*

The choices to a question are gathered from the separate tokens of an answer, as specified by the creator.

Dummy choices are introduced in one of two ways:

1. By assigning them to a choice that exists in an answer.
2. By explicitly assigning them to appear at the index of the answer, once the user is located there.

Building on our previous example, assume the answer is to be constructed in three steps, i.e.

1. Boys
2. Have
3. Balls

Using option 1:

* Boys -> Boy // Boy is a dummy and will appear whenever Boys appear

Using option 2:

* Boy -> 0 // Boy is a dummy and will appear at index zero of the answer

In this example, both options produce the same result. The user will have to choose between boys and boy when starting to build the answer. In the first option however, if the token “boys” were to appear again later in the answer, “boy” will again appear as well.

Every question may have more than one alternate (equally correct) answer.

**Game Goals**

The goal at each level is to complete a number of questions without losing all hit points (without “dying”).

There are multiple levels of goals in each Level. These are determined by the amount of questions that need to be answered at a level. The first goal level is the minimal threshold required to complete a level. The third is the most difficult to reach, and the second goal is meant to invoke player motivation to reach the third goal level.

The

We denote X, Y, and Z are the percentage of questions that need to be completed at the first, second and third goal levels, and N as the total number of questions in a level. We include the reward received at each level as crucial to player progression in the game, denoted as Reward point. Thus the following may be stated for formally:

1. Complete X/N questions – Receive 1 Reward point.
2. Complete Y/N questions – receive 2 Reward points.
3. Complete Z/N questions – receive 3 Reward points.

These simple goals may be augmented with additional conditions regarding other game elements, such as user health, power and time.

**Multiplayer Gameplay (superset of single player mode)**

A game instance (round) happens in a single level. At least one player is required to start a game.

The players are divided into two teams, each with a maximum limit of players. The purpose of the teams is to gain control over all questions in a level. This is a team effort, which also incorporates some vs. elements.

The game ends when all questions have been completed, or when a team loses all its combined hit points.

The game starts with the teams at different challenges, and the players scattered randomly across different questions within the starting challenge.

The choices available to a question are hidden until the question is engaged. This forces the player to first focus on the question at hand, and not the available choices. It also requires a certain strategy of question engagement, meaning that players must first engage the questions they are most confident about.

Once a player engages a question, the available choices are discovered. The question is now occupied by the engaging player, meaning that no other player may submit answers to that question. Players are however able to see the answers that are submitted by others.

Hit points are lost if a player submits an incorrect answer.

Every challenge in a quest (question) is supplemented with a set of possible answers. This is the main interaction. A player clicks on an answer and receives feedback. Sound shall be used to enhance the experience, indicating a correct answer or a mistake.

Engage Question Use-case

User is at an incomplete question

User clicks the engage button.

The event details are propagated to the server. The Question is added to the client and locked from other clients.

The choices and solutions are returned.

The user is at index=0 of the answer. He clicks a choice. The following happens:

1. At client side, all the answers that contain a matching choice at current answer index are retrieved.
2. If no answers match
   1. Show choice in red.
   2. Update screen with new game state.
   3. Send event details to server.
   4. Update screen with game state from server (happens when game states differ).
3. Else one or more answers are found
   1. Show selected choice in green and add to the formed choice sequence.
   2. From the matching answers, if one is found such that: answer index == answer.size-1
      1. Complete answer submitted.
      2. Repeat 2.b – 2.d.
4. Repeat all until all answers are discovered.

Features

Glossary

MAX\_CHOICES = #maximum allowed unique choices in a question = 8

MIN\_CHOICES = #minimum allowed unique choices in a question = 1

MAX\_ANSWER\_SIZE = #maximum allowed sequence of choices that makes an answer.

MAX\_DUMMIES = #maximum allowed dummy choices in a question = MAX\_CHOICES – 1

MIN\_DUMMIES = #minimum allowed unique dummies in a question = 0

DUMMIESq = #dummy choices in question q

REAL\_CHOICESi = #correct choices that can be submitted at location i of an answer

CHOICE\_SCENARIO = an empty sequence of x choices, together with the available choices.

DUMMY\_HIT\_PROBABILITYi = #dummies / #reals

HIT\_POINTS\_QUESTION = 100

SOLUTION\_SIZE = #answers left in question (>0)

COMBO = #consecutively submitted correct choices in a question

HIT\_POINTS\_MULTIPLIER = COMBO \* some normalizing factor

HIT\_POINTS\_CORRECT\_ANSWER = HIT\_POINTS \_QUESTION / #answers in question

HIT\_POINTS\_ANSWER =

QUESTION\_STREAK = #question eliminated in a row without dying.

HIT\_POINTS\_ASSIST =

The purpose and goal of each level is to accumulate as much intelligence point as possible. This is done through completing questions. Every level carries a number that acts as a hit point multiplier. This is to create an illusion of higher difficulty

Intelligence Power

This is the game score.

Every answer in a question is assigned an amount of hit points. The number of hit points granted for submitting a correct answer is calculated as a fraction of the total number of answers in a question.

The players start with an intelligence level at zero.

When an answer is correctly submitted, the player‘s intelligence level increments by the answers hit point amount, and the question loses the same amount.

When an incorrect answer is submitted, the player loses the same amount of hit points from his health.

Combo is an intelligence multiplier that takes affect when a correct answer is submitted. It starts counting when two consecutive choices have been correctly submitted, and increments with each correctly selected choice.

It rewards players for submitting a long answer; those that are made of a number of consecutive choices.

XP (Intelligence Level)

Player is granted points after every move which results in a positive outcome. For submitting a correct answer, player is granted an amount of points equal (or multiplied by some constant factor) to the weight of the missing piece. For the correct use of any item, the player is rewarded a constant amount of bonus points. Optionally, each item may have its own XP scheme, where the points relevant to the used item are increased, thus allowing upgrades.

Elimination

The ability to eliminate pieces that are irrelevant to the current challenge. This is done by first toggling the eliminate button on (or two finger touch if touch is implemented), and then clicking on an answer to eliminate. Once an answer is eliminated it does not appear as an available answer in the challenge any more - the shield stripe is shrunk. No feedback is received when eliminating an answer but additional points are granted at challenge end for correct use of ability.

Elimination is required to start a combo (?)

Submit

Submitting an answer is done by clicking on a shield stripe. When a correct answer is submitted, the UI indicates it to the user – the shield stripe is shrunk, combo increases if possible, points are rewarded and a written feedback is shown ,such as “good” or “great”.

Combo (Correct Answer Streak)

Combo is an intelligence multiplier that takes affect when a correct answer is submitted. It starts counting when two consecutive choices have been correctly submitted, and increments with each correctly selected choice.

It rewards players for submitting a long answer; those that are made of a number of consecutive choices.

Answer Streak

Shield Power

At the end of each challenge, the shield is charged based on the amount of correctly used stripes. The shield remains charge throughout all challenges in a level. It enables the use of Special Items.

The charge is based on the amount of expanded stripes at the beginning of each challenge. Every shrunk stripe (resulting from correct use only, either by submission or elimination) charges the shield with some amount. Combo should affect the shield charge.

Magic Items (Special Abilities)

The player can use special items (e.g. health or help) where each use of an item results in either a positive or negative outcome. An item must first be charged before it can be used (this can be achieved by either the combo mechanism, or using some sort of “fuel” to charge items up before or during gameplay.).

Magic Items are charged using the shield power.

// BEGIN IRRELEVANT

Once an item is selected and charged, the outcome depends on the result of the next answer submission. If the submitted answer was correct, the positive outcome of the item is in effect. If the answer was incorrect, the negative outcome is in effect.

// END IRRELEVANT

Some of the items are: (perhaps divide to different characters)

1. Heal: Increases player health by some amount. Very hard to charge.
2. Time Extension: Stops clock count for X seconds, where X is determined as a constant fraction of the initial clock time. Relevant only for timed quests.
3. Undo: provides undo operations of the last submitted piece. Cannot be used after last piece is submitted.
4. Armor – protects player from health decrease if wrong answer is submitted.
5. Wild Card – a “Joker” answer. Always correct, but the real answer is not revealed.
6. Automatic Elimination: automatically eliminates one (or possibly more) piece that is irrelevant to the current missing piece.
7. Reveal – Marks all correct choices of a solution in sequence. Starts with the first choice, if the player clicks on it, the next correct choice is marked. Currently the most powerful item.

Achievement Levels (Intelligence)

An achievement level is a condition that is evaluated during a quest, which concerns either the percentage of challenges to conquer, or that of pieces to be submitted correctly. There are three (number should not be hardcoded) achievement levels in which a character may complete a puzzle. A quest master must determine values for all three conditions when creating a quest.

**A quest is completed if during play, at least one of the achievement levels is reached.**

For example:

1. 60% = Easy (
2. 80% = Normal
3. 100%+ = Hard

Additional constraints may be imposed in case of timed quests.

Solution Granting

Once a quest has been completed, the character is granted the correct answers (solution) which were submitted by him. Once a solution is in possession, it is available for review purposes at the library (a place where characters may meet, and review passed quests and their solutions).

World Progression

The world is an ordered list of nodes. Characters progress from one node to the next after accomplishing all assignments at a node. A node contains one or more quests, and assignments impose conditions on how to accomplish quests.

Solution Ranking Algorithm

The algorithm acknowledges the order of answers in submitted solutions, calculates the amount of discrepancies between

Let be the current challenge, and be the set of all existing solutions in the challenge.

We define as a vector of strings constructed from the available answers in .

Let be a predefined solution to , where is an integer in the range , and as the solution submitted by the user , where .

We refer to an item in a solution vector as , where is an integer in the range .

Finally, we define to be an item with an empty string.

The goal of the algorithm is to remove items from so that eventually it shall contain a single solution closest to .

The algorithm returns an integer vector ,where , and which holds the following flags:

1. : indicating a correct answer at index j.
2. : indicating an incorrect answer at index j.

We define the following functions:

1. : retains all solutions that contain the item .
2. : retains all solutions that equal the length of the second argument.
3. :

Guy Manzurola