Michał Szczygieł, G8927
Aleksander Śmierciak, G8925

Interactive character sheet for Dungeons & Dragons Next edition

HTML5 Web application programming

Introduction

Dungeons & Dragons is a fantasy role-playing game dating as far into the past as 1974, when its first edition was published. The back then innovation of assigning each player not a military division, as opposed to how miniature wargames work, but a single character who interacts with the surrounding world with his or her decisions and mediation of the so-called Game Master made the game immediately successful and even now still considered best-selling and best-known in the RPG genre.

Each player except for the Game Master holds his or hers character sheet, in which many game mechanics-related attributes, skills and feats of character are noted, so that the randomness of in-game fate is restricted by one's predisposition. Finding equipment, gaining significant experience or prowess in either battlefield or craftsmanship, learning new spells, suffering from sickness or damaging blows - all of these and much more is to be scribed in the character sheet.

39 years later we are presented with Wizards of the Coast's (current Dungeons & Dragons publisher) imminent release of the next edition, dubbed Dungeons & Dragons Next. Bent on improving the pace, intensity and performance as well as many caveats that the previous, 4.0, edition has brought, WotC opened a public beta-like system, in which those who register are able to download playtest packages and familiarize themselves with changed rules as well as bring WotC much needed feedback.

Attached to the playtest package is a well-known to an RPG gamer character sheet, which we as a team of two students intend on bringing to HTML5, CSS3 and JavaScript technologies, to provide any player with the possibility of actively and dynamically editing its content as well as saving his or hers work for later use.

Reasons for such project topic

As weird sounding and irrelevant for a non-player this may sound, character sheets are the most frequent changed pieces of paper for RPG gamers in the course of a game. One can easily imagine a piece of paper repeatedly treated with pencil and eraser (or worse, with a pen, scratching the old values) wearing off really fast.

An interactive, responsive and effort-minimizing character sheet is what is rarely found in the Internet. Most solutions to the character sheet problem revolve around spreadsheet files with formulas for calculations or editable PDFs, both which bring about compatibility and portability problems and fall prey to the software lock-in issue. In the age of rich Web applications, in the age of HTML5 standard nearing its complete implementation and supplemented by both CSS3 and JS support in modern Web browsers, a fully Web-based solution has to exist.

Unfortunately, this is not the case. Web-based approaches generally are incomplete, insufficient, require some kind of fee (which prevents us from testing them completely) and/or incorporate Flash, which itself falls from Web browser vendors' grace currently.

Thus, an idea of writing a character sheet that'd solve the problems of frequent modifying and also retain high portability and usability thanks to being Web-based was born.

Layout

The character sheet is basically a form with many text, numeric and alphanumeric inputs, most of which have an impact on others.

Examples:

1. Character's strength ability impacts his or hers carrying weight limit, as well as push and pull limit for particularly heavy objects; it impacts strength modifier as well.
2. Character's constitution ability impacts his or hers hit points total - numeric representation of how healthy and vital a character is currently.
3. Character's dexterity ability as well as character's armor mastery skills and the armor itself impact character's armor class, which as a final modifier is used when a damage is applied to said character.
4. Unfortunate character's sicknesses and curses may lower (either temporarily or permanently) character's abilities, effectively also lowering any dependent attributes and modifiers.

These features make layout relatively hard and challenging, since everything needs to be in perfect order for a player to be usable and efficient.

The initial layout will mimic playtest package's character sheet and might be further developed to provide a better gaming experience.

The materials we rely on and the character sheet itself are available from the Wizards of the Coast™ webpage for Dungeons and Dragons Next edition.

Architecture

In order to create this project, we decided on incorporating advanced HTML5 form elements and markup tags, as much CSS3 as possible in order to minimize JavaScript usage for visual and JavaScript as a calculator and information database for the form fields.

Functional requirements

1. Game-related interaction
    a. user should be able to fill in the character sheet and reflect his or hers character with the application fully, including:
        i. description (physique and looks), such as race, height, weight, hair and eyes color;
        ii. experience, such as current experience points, points to next level and the experience level itself;
        iii. abilities, such as strength, dexterity, constitution, intelligence, wisdom and charisma;
        iv. attributes, such as speed, hit dice, initiative, vision, size, all numerical and/or text according to Dungeons & Dragons ruleset;
        v. languages known and wisdom possessed;
        vi. attack, combat prowess and maneuvers;
        vii. class features, racial traits, skills and feats;
        viii. spell notes, prepared spells and spellbook;
        ix. additional character notes;
        x. equipment & treasure.
    b. user can be additionally provided with playtest notes field for describing his beta experience with Dungeons & Dragons Next edition.
    c. user should be able to perform changes at any time, reflecting his or hers character state change,
    d. user should be able to reflect his or hers character level up in the character sheet.
2. Form-related interaction
    a. user effort in creating a character sheet of his or hers own should be minimal.
3. File operations
    a. user should be provided with a lean and fast interface, not unlike the original character sheet,
    b. user should be able to save his or hers work,
    c. user should be able to retrieve once saved work.

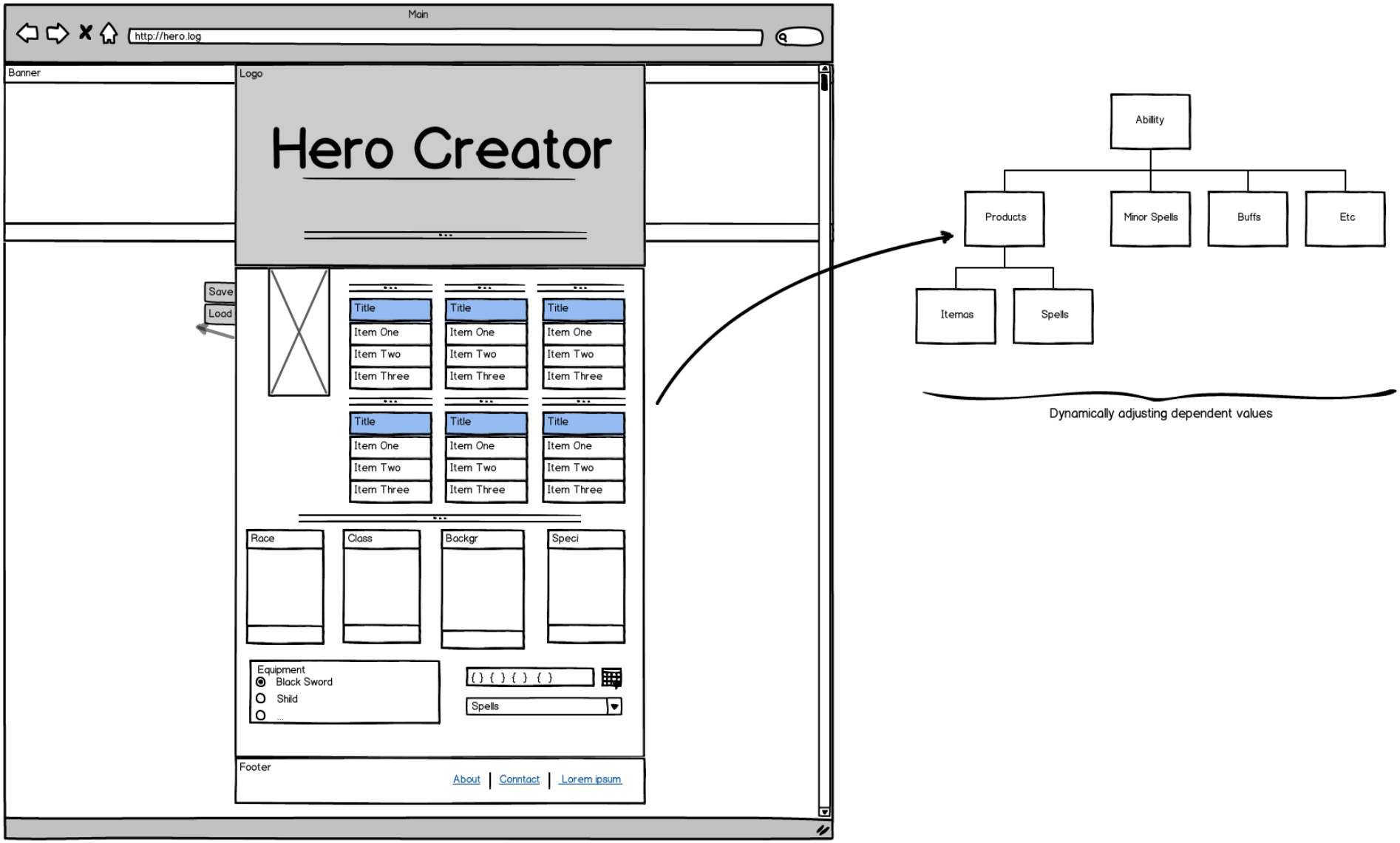
Non-functional requirements

1. Game-related interaction
    a. all character properties should be subject to Dungeons & Dragons mechanism ruleset, described in handbooks (referencing F1. a)),
    b. playtest notes field can allow the user to submit his data to the creators of Dungeons & Dragons Next edition (referencing F1. b)),
    c. changes made to one's character's properties should be subject to validating before accepting new value and updating other, dependent values (referencing F1. c)),
    d. changes made to one's character should be visible immediately after validating, not requiring any page refresh or additional accept button actions from user (referencing F1. c)),
    e. current experience level and experience points should be adjustable by the player by adding values he or she received or by typing the new value (referencing F1. d)),
    f. changing experience level should grant the player's character development points, as stated in rulesets and charts (referencing F1. d)).
2. Form-related interaction
    d. user should be provided with autocompletion and dropdown menus with items as often as possible to minimize effort in creating a character sheet (referencing F2. a)),
    e. some fields should be filled automatically depending on values of other fields to minimize user effort in creating a character sheet (referencing F2. a)).
3. File operations
    a. the user interface should mimic the original character sheet; at the very least, a printable original layout should be provided in addition to the default one (referencing F3. a)),
    b. user's work should be saved as a JSON file (referencing F3. b) and F3. c)),
    c. a validation of JSON content should take place when loading previously saved work, including checking for both JSON file structure validity and relevancy of the content itself.

Michał Szczygieł, G8927
Aleksander Śmierciak, G8925

Template and Layout

Michał Szczygieł, G8927
Aleksander Śmierciak, G8925



Compatibility of achieved solution

Our work resulted in creation of a modern Web browser-oriented webpage with minimalistic CSS - only working in a few different browsers, most notably Mozilla Firefox 20.0 (alpha at the time of writing), Opera 12.12 and Google Chrome 24.0.

To provide more browsers compatibility, certain fallback CSS3 features would need to be introduced. For example, what was used heavily for the layout: flex box design and multi column design is not yet implemented completely in the modern browsers. The solution would be to detect such situation and resort to floats and/or pixel positioning.

Timetable

	Michał Szczygieł	Aleksander Śmierciak
Week 45	Initial concept, planning	Initial concept, planning, writing specification
Week 46	Mockups, ideas for new features, vector graphics, raster graphics	D&D mechanics, providing the basis for the work
...	Holidays	Holidays
Week 1	JavaScript save/load functionality, JavaScript dynamic loading for the fields	HTML5 / CSS3 layout
Week 2	Movie trailer for the project, project presentation	HTML5 / CSS3 layout some more, finishing touches, project documentation

What was done

Everything from the ground to a working and styled character sheet system, in which values and user inputs are validated both on leaving the input field and on the JavaScript load, though only in terms of input length and character sets used.

What hasn’t been done

More sophisticated validation, level up system, weapon system, spell system, classes, items and etc. showdown...
The project will most certainly be continued as it brought a lot of educational (sometimes in a painful way) and entertainment value.

Hour load

The hour load was heavy for this project, as nearly as much as 80 hours per person were spent, not even beginning to tackle the advanced ambitious functionality written in the specification.
The slowest development noted was CSS layout due to its quirks and odds and JavaScript because of its lack of compiler/debugger-like support (we’re programmers at heart :-)).