Celtra challenge - HTML5 game

This game is being developed to enter Celtra's HTML5 game challenge.

URL to repository: https://github.com/markogresak/celtra-game

Game and it's features

The game is being played on infinite, randomly generated 2D platform of blocks.

When user opens the game, they are presented with Username input field, which is later used as their display name in game. The gameplay is quite simple, one can move left or right on platform and attack other players. There was also idea of leveling up, earning gold and using shop to buy better weapon or armor, but unfortunately, 2 months was not enough time to implement all of these ideas.

Controls:

- move left: `A` or ← or touch left half of screen
- move right: `D` or → or touch right half of screen
- attack: `Space` or touch attack button (mobile only)

About the project

The game is being developed using dart programming language.

I'm using dart because it has promising features to avoid issues that occur when working with JavaScript and other transpilers.

It has *optional* typing system, offers true object oriented design and great on-run error reporting, to name a few.

Dart SDK offers a great amout of libraries, most important for this project is `dart:html`, which supports DOM, canvas and even WebSockets. On server, `dart:io`, among other features, offers WebSocket interactions. This means the client side doesn't need any additional libraries, it's all running inside Dart VM (if using Dartium) or inside `main.dart.js`, which is generated when building project, either with `pub build` or `pub serve`.

Also, Dart dev team claims dart runs faster or just as fast as corresponding JavaScript.

How to run project

1. Install dart

Easiest way is to download and install dart SDK + editor from dartlang.org.

Other options:

OS X using Homebrew:

```
$ brew tap dart-lang/dart
$ brew install dart dartium
```

Or *read more about installing on different configurations*, i.e. Linux operating systems, including servers.

2. Clone repository

```
git clone git@github.com:markogresak/celtra-game.git
```

2.1. Update `SERVER_ADDRESS`

Update `SERVER_ADDRESS` value inside `web/dart/serverConnect.dart` to match address of WebSocket server (the address of machine used at 3.1.).

3. Running servers

In order for game to run, WebSocket and hosting server have to be running. Reason for using 2 servers is so the two can run on separate machines with very little change to code.

3.1. Run WebSocket server

1. Change directory to `server/`, from project root, run `cd server`.
2. Run server: `dart server.dart`.
3. *Note: make sure port 9999 is accessible.*

3.2. Run hosting server

1. Change directory to project root.
2. Run: `pub serve`
3. Optionally add `--hostname` to pub serve, so server can be accessed from other devices, e.g.: `pub serve --hostname 192.168.1.X`, where *X* is local ip address.

License

MIT. More info in LICENSE file.