

Example database structure

table users

- id (int) – a primary key for the table
- email (string) – email address for the user
- password (string) – some kind of hash derived from the user password
- progress (int) – the maximum level id that the player has finished

table levels

- id (int) – a primary key for the table
- name (string) – name of the level
- thumbnail (string) – URL for thumbnail for the level
- waves (int) – number of waves of monster before the level is finished
- start_gold (int) – amount of gold that the player has in the beginning of the level

table level_paths

- id (int) – a primary key for the table
- level_id (int) – foreign key from the levels table
- x (int) – the X coordinate of a point in a path
- y (int) – the Y coordinate of a point in a path

table level_waves

- id (int) – a primary key for the table
- level_id (int) – foreign key from the levels table
- wave_id (int) – id of the wave
- monster_id (int) – foreign key from the monsters table

table monsters

- id (int) – a primary key for the table
- name (string) – name of the monster
- other monster properties can be defined here or in the classes in the corresponding classes on the client side

table towers

- id (int) – a primary key for the table
- name (string) – name of the tower
- description (string) – short description of the tower
- other tower properties can be defined here or in the classes in the corresponding classes on the client side

Example application structure

controller SignIn – responsible for users login/register

- action index – displays login, registration and forgotten password forms to user. The later two must be initially hidden until the user asks for them.
- action login – receives email and password from the login form and verifies them against the users table in the database. On success the user is considered authenticated and redirected to the Game Lobby page.
- action register – receives email and password from the registration form, validates them against the application rules for email and password and creates new record in the users table. The user is then considered authenticated and redirected to the Game Lobby page.
- action forgottenPassword – receives an email address from the forgotten password form. Searches the `users` table for such email and if found generates a new password and sends it to the address.

controller GameLobby

- action index – displays the list of the available levels taken from the levels table. Some of the levels that are not unlocked can be disabled or marked as not yet available.

controller Level

- action index – receives levelId and loads the full information for a selected level taken from the `levels` and `level_paths` tables. Some of this information is delivered in format acceptable for the client side application (for example JSON or XML). The client side application then uses this information to render the level on the screen and start the game.
- action finishLevel – Called when a user successfully kills all the monsters in a level. Receives levelId and marks the user progress in the `users` table. For the purposes of this application no further validation is required.