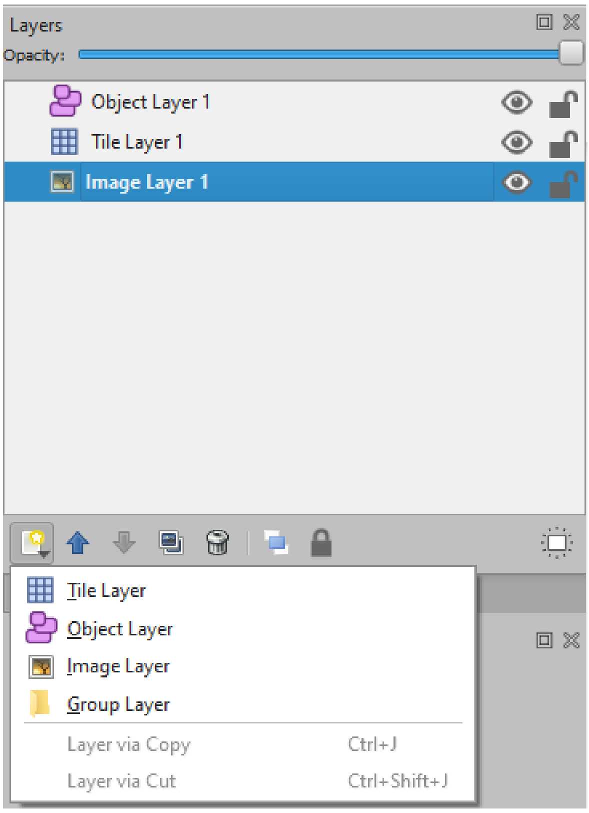
How to create your own level

1) Install Tiled Map Editor from here:
   https://thorbjorn.itch.io/tiled
2) Create a new map with settings
   a. Orientation: Orthogonal
   b. Tile layer format: CSV
   c. Tile render order: Right Down
   d. Map size: fixed, with Width and Height as you wish, but preferably at least 23 on Width and 12 on Height to fill the whole screen
   e. Tile size: recommended 64 x 64 px, but you can choose other size if you have appropriate sized tileset for terrain
3) Every map has to have those 3 layers: Image layer, Tile layer and Object layer, so add them



4) In Image layer, choose an image that will serve as a background. The image has to have at least 768px in height (to fill the whole screen) and will be repeated horizontally.
5) In Tile layer, you will be placing terrain blocks in the grid. Create a new Tileset (based on a single image or collection of images) or import existing by choosing Map -> Add external Tileset and choose a file with .tsx extension. You can import a game's default tileset by choosing a file DefaultTileset.tsx in /src/main/resources/Objects. All tiles have to have a custom property "kind" with one of values: {"solid", "water", "air"}. DefaultTileset has those properties already set.
6) In Object layer, you will be placing all the other objects. Create a new Tileset (based on a collection of images of objects) or import existing Tileset (default Tileset for objects is DeaultObjects.tsx in src/main/resources/Objects). Objects can have their own skin, and there can be multiple objects

with the same properties, but different skin. Every object has mandatory property “class” with one of the values: {“Hero”, “SimpleEnemy”, “Turret”, “Star”}.

a. Hero – they can have a property “type” with one of values {“fire”, “ice”}. In this case, they have properties “movementSpeed”, “jumpSpeed”, “swimmingSpeed”, “lives” and “HUDImageSource” as optional (will be filled with some default values), otherwise (if type is not specified or has other value) they have “HUDImageSource” mandatory.
    i. movementSpeed – info about speed of the hero (default 280 for “fire”, 250 for “ice type”, 300 for other), must be a number
    ii. jumpSpeed – info about jump height of the hero (default 410 for “fire”, 250 for “ice type”, 350 for other), must be a number
    iii. swimmingSpeed – info about swimming speed, (default 0 for “fire”, 120 for “ice type”, 0 for other) put 0 if the hero should drown in water, must be a number
    iv. lives – how much lives does the hero have at the start of the level (default 3 for “fire”, 4 for “ice type”, 3 for other), must be an integer
    v. HUDImageSource – path to the file (with the base repository /src/main/resources/) with the image of the hero that will be displayed on the top of the screen (default “Objects/HeroFireSmall.png” for “fire”, “Objects/HeroIceSmall.png” for “ice”). Specifying the path of image that doesn’t exist in base game might result in a need of the recompilation of the game.

b. SimpleEnemy – they have only one mandatory property “routeLength” which means, how much distance (in pixels) to the right should the enemy go before going back to the original position (and repeat the route)

c. Turret – they have a mandatory property “type” with one of the values {“fire”, “ice”, “combined”}, a mandatory property “shootingAngle” meaning the angle (in degrees) at which the turret should shoot, with 0 meaning to shoot down and 90 meaning to shoot right. Must be a number. Also, there are optional properties “shootingInterval”, “shootingSpeed” and “pathToBulletImage”.
    i. shootingInterval – info, how often (in seconds) the turret shoots, default to 1.5, must be a number.
    ii. shootingSpeed – info, how fast is a projectile shot by a turret, default to 400, must be a number
    iii. pathToBulletImage - path to the file (with the base repository /src/main/resources/) with the image of the projectile shot by the turret.(default “Objects/BulletFire.png” for “fire”, “Objects/BulletIce.png” for “ice”, “Objects/BulletCombined.png” for “combined”). Specifying the path of image that doesn’t exist in base game might result in a need of the recompilation of the game.

d. Star – they have an optional property “value” which says how many score points will the player get for claiming this star. Default to 1, must be an integer. Take note that the level is completed when the score is equal or higher to the sum of scores of all start in the level (which means that if you want to make a level, where the player doesn’t have to get all stars, make some star/stars with negative value)

7) Save the file and load the level from the game and enjoy your custom level! In case of troubles, contact me via mail DWaloszek@seznam.cz