
CELTRA'S PROGRAMMING CHALLENGE 2017

HTML5, JS GAME

Ivan Čaleta-Car

ABOUT THE PROJECT

The task was to create a 2D game playable in various web browsers of mobile devices using only Javascript without any external libraries (with HTML5 and CSS).

BRIEF DOCUMENTATION AND IMPLEMENTATION

The game is named **PIXEL SURVIVOR**, as the player is made up of rectangles. All the graphics for the game were created using Adobe Illustrator and Photoshop, and the sounds were found on the Internet and are open source. The two web pages are fully responsive, and the content adjusts according to screen size.

The game content scales up to a point. This was solved as follows: character size, collectables and obstacles have default sizes and if the screen width is smaller than 800 px, the player size will be reduced by 30%, the size of collectables by 20% and obstacles will be reduced by 35%. This avoids complex mathematical operations and provides a solution that doesn't affect user experience.

The main character is controlled with keyboard controls(PC) or with onscreen buttons. The keyboard listeners are always added to the site, as a user can connect a keyboard to his device and continue to play. If the user is playing on a mobile phone, he first needs to choose type of controls: left handed or right handed and can switch this option during the game. If he is using a large touchscreen device (without a keyboard), he can choose to display the additional

controls from the panel on the left. This panel also displays a mute button. The mute button is displayed in the top right corner of the screen on a mobile device.

For these types of games (endless and similar) the user will likely give up playing very quickly so my goal was to create the simplest controls possible so that they don't affect his engagement. Gyro controls are not implemented because I think they are not necessary for this type of game and controlling the player would be too hard to control and they would certainly demotivate the user to keep playing. Swipe was also not implemented because this would be also hard for the user to play as the character can move in any direction and it would be imprecise. My opinion is that buttons are the optimal solution.

The score is measured in seconds survived and highscore, displayed on the top of the screen, is saved in a cookie.

The player can collect three types of powerups:

- *Jump powerup* which enables multiple jumps and lasts for 4 seconds
- *Score powerup* which adds 5 seconds to the score
- *God mode* which gives the player the power of invincibility for 12 seconds

The jump powerup and god mode also change the player appearance. After some period, the canvas background changes color and reverts later, as well as the switching of spawn location of obstacles which adds to the dynamic of the game.

It is important to note that when the game is restarted, only the canvas is reset without refreshing the site, which means only one page load is needed, and after all resources have been cached, to play as many times as the user wants.

The playing instructions can be found on the home page ([index.html](#)).

The game should work in any browser which supports HTML5. It has been tested in Chrome, Mozilla Firefox, Microsoft Edge on PC (Windows) and on several Android devices using Chrome.

CLASS LIST

Below a class list and hierarchy is shown, along with short explanations.

- RESOURCE - *contains paths to game and site resources*
- UTILITY - *contains various static functions for various tasks*
- KEYS – *contains keyboard keys definitions*
- PLAYER – *player class*
 - PLAYER CONTROLLER – *contains functions for player movement*
- CANVAS – *class for generating the canvas depending on screen size*
 - GAME CONTROLLER – *main class for controlling the game; contains instances of other controller classes*
- EXTRA CONTROLS HANDLER – *contains functions for onscreen controls*
- OBJECT GENERATION – *superclass for obstacle and powerup classes*
 - OBSTACLE – *superclass for obstacles*
 - OBSTACLE TYPE ZERO
 - OBSTACLE TYPE ONE
 - POWERUP – *superclass for powerups*
 - JUMP POWERUP
 - SCORE POWERUP
 - GOD MODE
- OBJECT GEN CONTROLLER – *class for timing the object generation*
- CANVAS WRITER – *contains functions for showing text on canvas; size is 16px and is universal for all devices*
 - SCORE CONTROLLER – *updates score and shows it on canvas*
 - INFO TEXT – *contains functions for showing information about powerups on canvas*
- COOKIE – *contains static functions for setting and getting highscore value in a cookie*
- GAME OVER
- SPLIT SCREEN - *used for splitting the canvas into intervals to avoid redundant collision checking*
 - COLLISION HANDLER – *used for checking for collisions between player and obstacles/powerups*
- GARBAGE COLLECTOR – *deletes obstacles that go off screen and inactive and uncollected powerups*
- SOUND CONTROLLER – *used to control game sounds, which are loaded once per website load*
- GAME – *contains functions for game setup and start*

The project has room for improvement regarding minor bug fixes and code optimisation, powerups and obstacles, new levels etc. which can easily be implemented by expanding the right classes or creating new ones.