Documentation

Github: https://github.com/Whiskey-Sour/Telerik-Apocalypse.git.

Team: Pavel Pavlov, Denitsa Dimitrova, Shammy Bg, Reni Getskovska, Stanislav Bozhikov, Pavel Dimitrov.

Not Participating: Ivan Trendafilov

Game Overview:

The resulting game is done in the traditional platformer style. The game includes movement and shooting mechanics. The obstacles present in the game are: high platforms, patrolling enemies, deadly ground, and a tracking turret. The aim of the game is to collect all instances of JS tokens. Player ammo is limited and can't be replenished.

Game Story:

The trainers at Telerik Academy thought it would be fun to have some robots running around the building. Unfortunately, things got out of hand and now the only one left is a young student called John. In order to stop the robots he must learn how to code. Help him collect all JavaScript logos!

Game Controls:

- Left/Right arrow keys- movement
- Up arrow key- jumping
- Spacebar-firing

General Code Structure:

Four Modules, external libraries, unit-tests, style sheet, and 2 html files for running both the game and the unit tests.

Modules:

main-module: is a module built out from the other three, has a method that initiates the game, and carries access to the other three modules.

game-module: has a method that builds the game, and handles all necessary operations for its running.

The module consist of 5 major elements. First initializing of Phaser Game, and creating of the canvas. The game then calls a preload() which loads in all the assets that will be used in the game.

After assets are loaded, the game calls create function, which generates objects that are initially present in the game world. Objects are organized in groups when possible, which act as both a parent class and a container for all instances of this class. The result is that every new object added to the group receives common features to all members. The project has a custom build object that takes a string and create a group with physical body. Create function calls : createWorld(), createAllGroups(), createPlatforms(), createPlayer(), drawHearts(), drawAmmo(), createBots(), createSpikes(), createBonusTokens(), createTurret().

Next update method is called. All dynamic elements are processed in the method. This method receives the user input. All the method calls itself and acts as the main game loop. Method called each game loop iteration: createController(), playerUpdate(), botsUpdate(), cameraUpdate(), playerCollision(), bulletsUpdate(), turretUpdate(), indicatorUpdate(player, oldLives, gameGroupWithPhysics, oldAmmo), endGameCheck().

After each loop of the update method is complete, a render method follows. The game doesn't modify anything at the time of rendering.

menu-module: has a method that builds the menu, and handles all necessary operations for its running.

It contains three main functions: addDiv(), addButtons() and addEvents(). They all use DOM manipulation and jQuery to build the menu. Here, the user is given the option to either play the game or take a look at the 'about' section. When any of the buttons ('Play' or 'About') is clicked, DOM is modified.

about-module: has a method that builds the about page and handles all necessary operations for its running.

It has two main components: a story section and a controls section. The latter uses snap svg library to show what controls are used to play the game and to create a simple animation. The story section describes the story of the game. It also contains a 'back button' – when clicked the body's inner HTML is cleared and the main menu is built again.

External Libraries:

- Phaser: used in the game-module. Game Engine as a canvas framework, used to build the game.
- jQuery(2.14 mini): used in the menu-module. Employed to create the menu.
- snap svg: used in the about-module. Employed to create the about page and its animations.

- Index.html :

- Runs game, and loads the needed scripts.

unit-test.html

- Runs unit tests and calls the needed scripts

style.css

- Usage in menu:
  - Buttons and background.
- Usage in about
  - Style the text and add back button.