

Created for the AUEB Course: Multimedia Technology, Xylomenos George
(<https://eclass.aueb.gr/courses/INF179/>)

Topic Selection: C. Game using p5.js

Title: REDET-521

Short description: The project is about a platform game with a user-controlled character with screen scrolling as the character moves. A survival game where the player moves across the unknowns of the planet REDET-521 to gather resources (mostly gears) to repair his rocket in order to return home. Furthermore he/she will have to stay alive experiencing the hardships of living in a planet with sparse oxygen. Its theme is inspired by the movie “The Martian” and by the game “Don’t Starve Together”.

Installation instructions:

The game program is written in HTML, CSS and p5.js (a library of Javascript). All you need to do to run it is install VS code, install p5.vscode extension of the VS code program, then open the folder “game” (which contains the HTML, CSS, JS files and the assets used in the game) with VS code (instructions also available at: <https://p5js.org/tutorials/setting-up-your-environment/> at the sub-category “Using VS Code”). Keep in mind that the folder “game” should be placed inside the folder that VS Code uses for its projects. For example, if you have already have a project running in VS code, e.g. at folderA/folderB/folderForVSCodeProjects/ProjectAButterfly, you will have to include the “game” folder at the folderA/folderB/folderForVSCodeProjects/ path.

After that, open the startScreen.html at the editor and choose the bottom right option “Go Live”. If the option does not appear, you may have to add the extension “Live Server” at VS Code or something similar. The start screen will guide you through the game.

List of external packages:

P5.js libraries were used for the animation and the sound effect of the game. The file “assets” contains images and sounds used for the game. The back, front, left, right, launching folders contain images (hand-made sprites) that aid to the animation of the astronaut walking and the rocket ship launching. The inconsolata.otf was used for the texts appearing in the game as normal text options do not work when you use the parameter WEBGL with the (central) createCanvas function of p5.js. WEBGL was used to access and utilize the camera object that “follows” the movement of the astronaut in game.

User instructions:

The only controls needed for the game are the W, A, S, D buttons for the movement towards up, left, right and down of the astronaut as well as the mouse-wheel for zooming in/out. The basic game over scenarios occur when the oxygen meter reaches 0 (as indicted at the top right of the screen) or one of the opponents (head) depicted as Giant Worms passes over the player depicting that it “eats” him/her. The winning condition is reached when the player finds, repairs and launches his spaceship. For more tips/info check the “Cheat sheet” section below. There should be no browser (or other) restrictions, the game was tested using the VS code and the Opera browser.

Cheat sheet (skip this section if you first want to try your luck at the game):

- The main problem for the astronaut is running out of oxygen so the main focus at the start is staying close to the base (to replenish oxygen when needed) and gathering grey boxes (containing gears) around it so that you can build the oxygen backpack (by clicking on the bottom left corner after you have gathered 10 gears or more). The oxygen backpack increases the oxygen capacity by 70 so after that you are ready to explore more.

- There is no map provided at the game so when you get out of base to explore you should remember the way back.

- Around the base, just outside your vision field when at base, there is a worm circling it so beware when leaving and then returning to the base not to get in its way.

- The rocket ship is located at the down right corner of the map (after some footprints found on the ground) and you can replenish oxygen there too (**near** the rocket ship). To repair it you will have to have its battery with you (located on the left side of the map, starting from the base, beware, its surrounded by two worms cutting circles around it) and 30-40 gears (the exact number is randomly generated every time).

- Gears can be found around the base (1-4 gears per box), at a gear pile at the down left corner of the map (20 gears there) and there are also grey boxes appearing at the upper right corner of the map (6, 7, 8 or 15 gears per box).

- After these, you can go to the rocket ship and it will be repaired (there are different phases of the rocket ship and reactions of the astronaut depending on if you have the battery or the gears with you when close to the rocket ship). By pressing X you ignite the rocket ship, by keeping the L pressed you prepare it for launching and after some seconds win the game.

High-level documentation of the program:

The game starts with the startScreen.html leading you to the guideScreen.html. Both are being styled with the screensDesign.css. The guideScreen.html then leads you to the REDET521_30FPS.html that runs the p5.js libraries at its head tag, has a footer and an aside tag that display the oxygen backpack on the bottom left and the oxygen and gears on the upper right of the screen. The rest of the screen is painted by the main tag in which the main p5.js script (alpha1at30FPS.js) runs.

The alpha1at30FPS.js has many global variables that are accessed by different functions throughout the game. The main functions used are the preload (preloading the assets needed for the game before running it), the setup that defines the canvas dimensions as these of the window used at the time and the draw function which runs after them and continuously drawing the frames seen.

The draw function calls the function drawTerrain (which draws the map terrain of the game), updates the oxygen and gears situation (linking the JS objects with the corresponding DOM HTML elements), the drawBase defining and designing the starting circle where the player can replenish oxygen, the handleEvents that monitors the game over conditions, when a cargo spawns (and delaying the appearance of the next if the current one is picked by the player), the drawWorm that designs the worm around the base and the two worms circling around the battery of the rocket ship (also checks if the worms got the astronaut thus affecting a game over condition), the drawPileOfGears which designs the pile of 20 gears (also checking if the player picked them up so that it stops being drawn), the drawRocketShip that draws the rocket ship in its different phases and displays instructions to the player accordingly and finally checks if a gameOver condition has been reached.

Additionally, the draw function draws one screen for the game while playing and a different one -final screen- (similar background but not moving) for when the player has won. There is lazy drawing (not drawing) being used in parts of the map that the player can not yet see. There is also a smoothnessFactor used when defining how often (per how many frames) an event occurs so that you may turn the game at 30 or 60 FPS (smoothnessFactor=1 or 2) so that the changing of the FPS does not affect the frequency of events. The game runs at 30 fps as is so that is is lighter without sacrificing a lot in visual performance.

Plenty of comments can also be found at the source code, guiding you through the whole running of the game.

Software resources used:

- p5.js
- VS code
- Opera browser
- painting (all of the visual assets are custom-made)
- painting 3D (all of the visual assets are custom-made)
- LibreOffice Calc (for the calculation of frames for every event to happen)

Other resources used:

- <https://www.youtube.com/watch?v=FXqEyKD5Ub4&t=0s> (Masahiro Sakurai Risk and Reward [Game Essence] – YouTube), about designing a basic risk and reward system for the game (getting as many boxes with gears you can but without straying too far away from the basement),
- <https://editor.p5js.org/aaronsherwood/sketches/H7D2yV3he> , visualization of how I would like the astronaut to move, to make it look realistic as he was walking,
- <https://www.drawing123.com/astronaut/#5> , inspiration for the designing of the astronaut,
- https://www.freepik.com/free-vector/hand-drawn-mars-landscape-background_3253093.htm#fromView=keyword&page=1&position=1&uuid=cddd7ece-8a46-4814-9b83-07840f40940f&new_detail=true , inspiration for the designing of the REDET-521 terrain,
- <https://p5js.org/reference/> , for all the useful stuff that was available and used for this project,
- <https://p5js.org/tutorials/repeating-with-loops/> , for the designing of the movement of the worms,
- <https://eclass.aueb.gr/courses/INF384/> “Web Development” course by Vasileios Zafeiris, for the needed HTML, CSS, JS to make the above work together.

Problems encountered and solutions:

- as a multi-media project, the main challenge was getting accustomed to different aspects of media and the time needed to do this. Had to find what to design as a game, which would be the main theme, goal and its mechanics, how to design its visual aspects, how to bind them together and progressively add stuff without breaking the previous ones made. The basic problems were making the astronaut move smoothly, how to make HTML and JS elements interact with each other while being shown at the same screen, how to avoid the trembling of shapes being drawn with p5 while the camera followed the movement of the astronaut. They were resolved more or less with patience and google search.

All in all, this project was a really exciting one getting you through various aspects of designing a game !