

Game example : Fun with most simple ALM steps for dummies and kids

OK, doing full ALM with Polarion is really fun - if you've ever experienced what application development means if you haven't it.

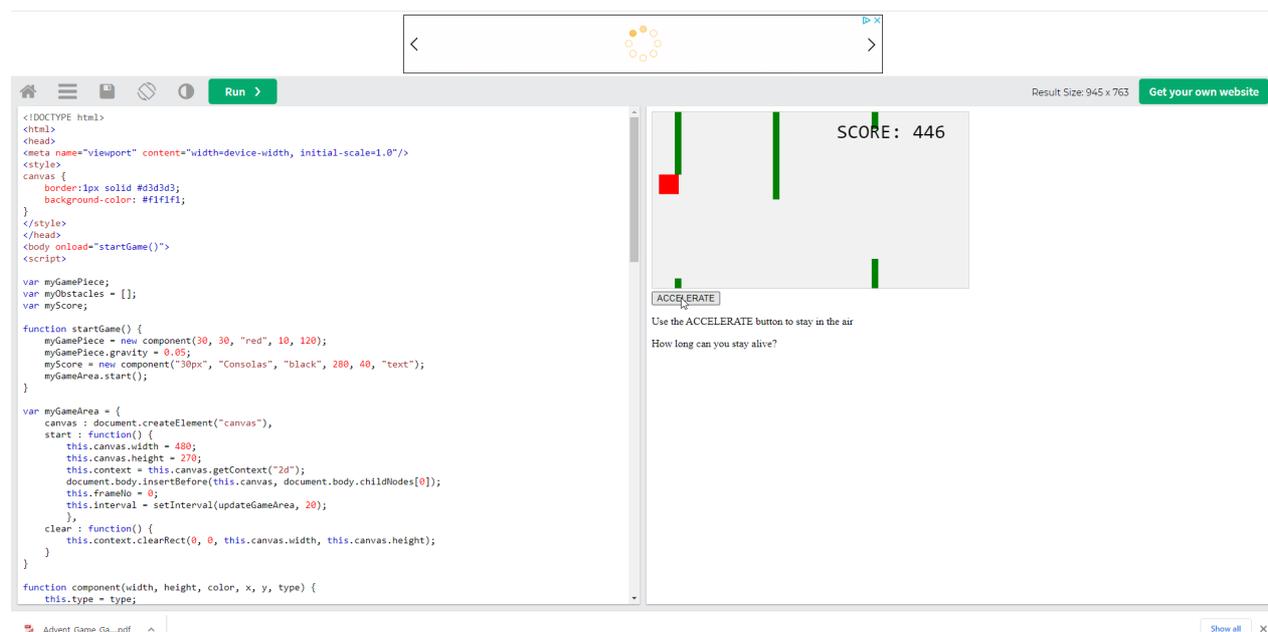
But how does it feel for somebody who never developed an application ? Newbies, dummies, kids ? Wouldn't it be great if one could experience the benefits of requirements, tasks, linked revisions etc with an easy example application which...

.. can be run and tested on any computer (needing only a web browser)

- consists only of one file
- can be easily adapted and developed further without much knowledge, maybe even by kids
- is even fun to run : a game !!!

Santa found it for you : it's an easy browser game at [W3Schools.com](https://www.w3schools.com) (where you can also gather a lot of valuable web programming knowledge, by easy step-by-step examples. Our Santas learned a lot there...)

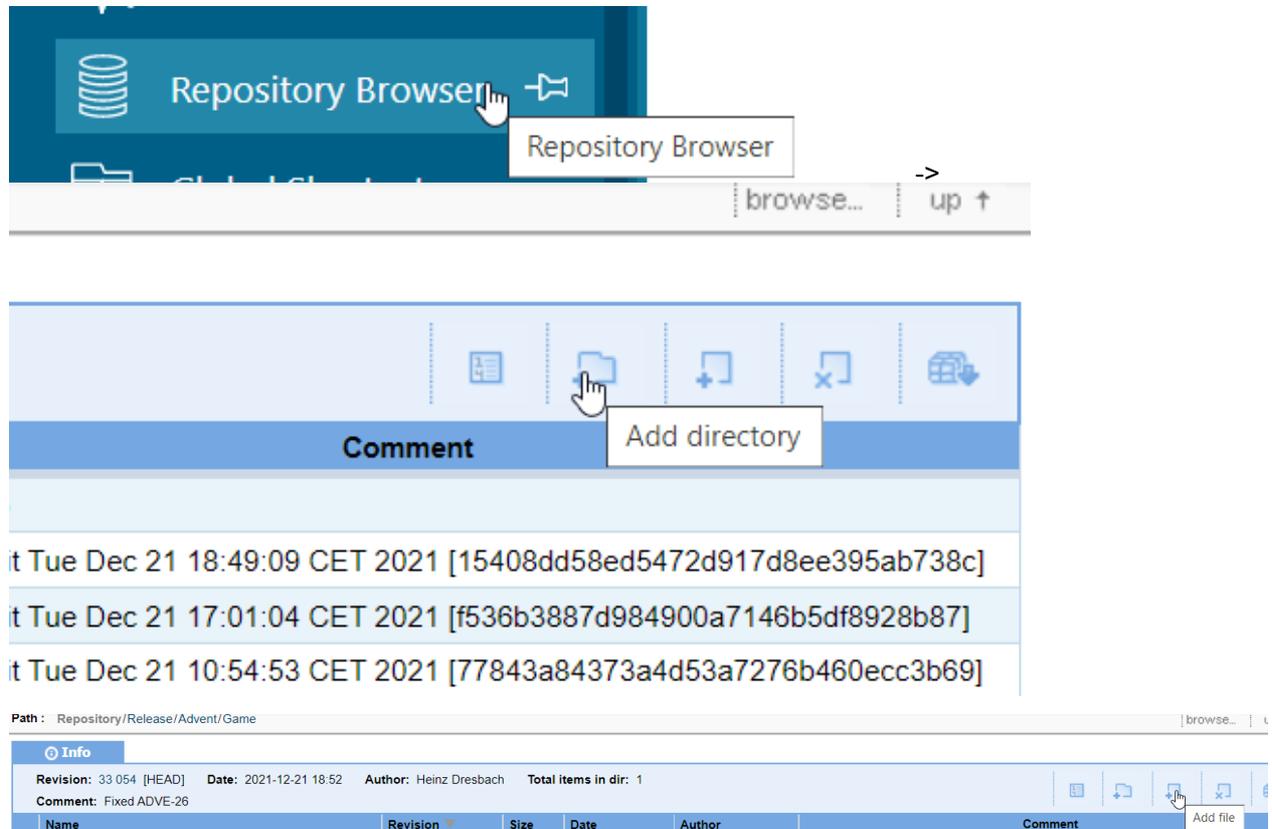
Try the game here : https://www.w3schools.com/graphics/tryit.asp?filename=trygame_default_gravity



< Source : [w3schools.com](https://www.w3schools.com) >

Copy the sourcecode from the left window into a textfile "game.htm".

In Polarion's Repository Browser, add a directory "Game", and add the file to it.



Now, you have a perfect base for playing ALM with Polarion : create a Requirement like "The figure must be blue".

In the game.htm, change the following line as indicated :

```
function startGame() {  
    myGamePiece = new component(30, 30, "blue", 10, 120);
```

Doubleclick the file to start the game , and you will see that the game figure now is blue.

Upload the changed file to the repository again, and in the comment type "Changed figure color to blue as requested in <Workitem ID of the requirement> ". Maybe set the requirement to a state like "Done" ,

Path : Repository/Release/Advent/Game

Add file

File path:
 game.htm

Comment:
Changed color to blue as requested in ADVE-27

In the requirement's item view, everyone now can see the linked revision together with your comment given.

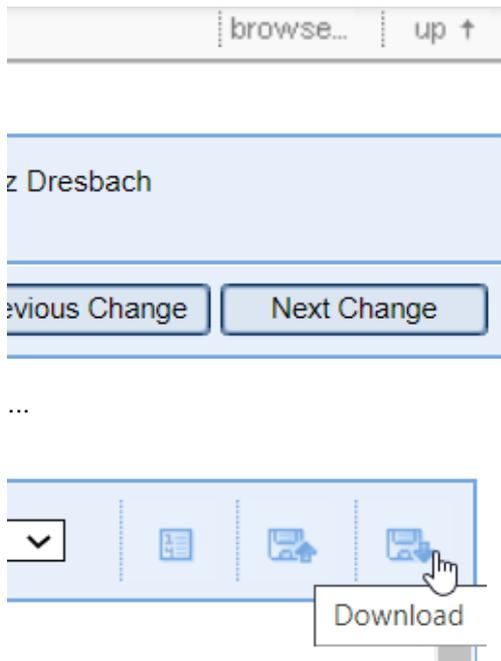
↩

Linked Revisions	
Revision	Message
 33057	Changed figure color to blue as requested in ADVE-27

With a click on it, everyone now can navigate from there to the repository browser to see the changes that have been done to fulfill the requirement

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
5 <style>
6 canvas {
7 border:1px solid #d3d3d3;
8 background-color: #f1f1f1;
9 }
10 </style>
11 </head>
12 <body onload="startGame()" >
13 <script>
14
15 var myGamePiece;
16 var myObstacles = [];
17 var myScore;
18
19 function startGame() {
20 myGamePiece = new component(20, 20, "red", 10, 120);
21 myGamePiece.gravity = 0.05;
22 myScore = new component("#333", "#ffffff", "51px", 200, 40, "font-size: 20px; font-weight: bold; text-align: center; color: #333; background-color: #fff; padding: 5px; border: 1px solid #ccc; width: fit-content; margin: 0 auto;");
23 }
24 </script>
25 </body>
26 </html>
```

And with hitting "browse" in the upper right corner, to download the new version of the games.



Next in the story could be a requirement "The figure must be so small, that I can reach 10000 points easily". Maybe also define a testcase for it.

You can imagine what to do then: download the game from the repository again, reduce the size of the figure (in the same codeline as above) from 30,30 to 20,20. Upload it again, referring to the requirements ID in the comment, just like before.

Then maybe execute the testcase. If you don't reach 10000 points, have it a defect ticket created, link that to the game revision etc. As you see, with such easy code example, you can easily explain actually any ALM method and situation to anybody. And you can improve your web programming skills with it.

Or you just sit back and enjoy a nice play...

Have fun !