

HTML5 arcade game web development project

This project is set up in a way that there are 4 scenes: The menu that loads in upon opening it, the game itself, a pause menu and a game over menu. First, I will go over the game features, then discuss the other scenes and finally give justification to what points I believe I should be awarded for what feature.

Description of the game and gameplay

The main goal of the game is to get as high of a score as you can. Players collect stars which increase their scores and cause various items to spawn when semi-random scores are achieved. These items are boots that increase the player's movement speed, money that is worth more points than stars, a stopwatch that allows players to temporarily freeze the movement of enemies, and a bomb that can destroy a limited number of walls. There are two types of enemies, ghosts and a zombie. Ghosts move randomly around the map and can go through walls, while the zombie always follows the player but cannot pass through walls. At the beginning of the game a given number of walls are generated and as the game progresses, more and more walls are added at a regular interval. The player can navigate the map using the arrow keys or wasd keys for movement, either shift for sprinting, mouse and space for item interactions. All objects in the game have custom skins and animations. The player animations are more complicated as they are tied to specific player movement. All enemies have attack animations when they kill the player. All sprites are from an open source and are given credit to their creators in comments.

Main menu

First scene that loads upon starting the program. It has a basic description of controls. To start the game the player needs to choose a difficulty. This difficulty level affects a few key points of the game. The higher the difficulty, the more ghosts spawn, the lower the player movespeed, and the more points are needed to get items.

Pause

The game can be paused by pressing escape. This causes the game to freeze and for a menu to pop up where the player can resume or go back to the main menu.

Game over

If the player collides with an enemy, they die. When this happens, the game scene slowly fades into the background and the game over scene appears. This has the score the player managed to reach and the high score that was reached in the past. This high score is saved to browser memory. There are two buttons, one to restart the game and one to go back to the main menu.

Points justification

I hope it's not a problem that the overall points add up to more than 40, I just got a little carried away while making the game and I would not like to underplay the number of features added just to fit into the 40 max point range.

Feature	Max points
Predifined	
Well written PDF report	3
Application is responsive and can be used on desktop environment	2
Application works on Firefox, Safari, Edge and Chrome	3
The application has clear directory structure and everything is organized well	2
Custom	
There are clearly separated menus and paused states from gameplay	3
There are 3 different difficulties which all modify parts of the game	2
There are unique animations to all collectables, enemies and the player	3
+ Player animation is complex (running, walking, different directions)	1
+ On player death, only the enemy that killed plays killing animation	1
There are multiple unique items that all have different effects	2
+ All walls are interactive objects because of bomb functionality	1
+ Bomb uses pointer tracking to destroy walls	2
+ All enemy movements are done in a way where it can be paused	1
Player movement is complex (sprint, modifiers to it, choice wasd)	2
Fine tuned hitboxes for enemy, wall and player interactions	3
Walls are automatically generated without issues	1
+ Initial wall generation based on screensize	1
Game screensize is adjusted to window size and gridsize	3
There are popup messages that let the player know about key moments	2
+ Multiple popups are allowed at the same time	1
Ghost movement done by utilizing physics groups	2
Zombie movement dynamically follows player	2
All game modifiers can be easily modified due to good practices	3
Game can be paused and resumed without change to gamestate	1
On game over, the final snapshot fades to be the background	1
Score is collected and shown, highscore is shown	1
+ Highscore is stored in browser memory	1
Buttons are implemented in menus	1
Player diagonal movement is normalized	1
Application has a (mostly) well documented git history	1
Total number of points	53