Matthew Fong

IGM 330 - Game Documentation

For my canvas game project, my initial game idea was to make a single player running simulator game similar to *Run 2*. However, after some more thinking and pre-planning, I changed my "game plan" and decided to make a two player "King of the Hill" type game similar to *Bara Bari Ball* from Playstation's *Sportsfriends.*

**Media Requirements:**

**Sound:**  In terms of sounds, I included a looping background song from *Space Jam's* theme, "Are You Ready For This", which I thought was very fitting and appropriate for the game. I also included sound effects for when the player "punches" and a popping sound from the ball’s effect when a player is carrying it. All sound files are in mp3 format.

**Images:** In terms of images, all images used (background, controls, and player sprite sheets) are all optimized bitmap graphics, saved as PNGs. I feel they are consistent in theme and style with the simple pixel theme and consistent color contrast (red vs blue).

**Text/Fonts:** In terms of text/fonts used, I feel using a pixely font is theme appropriate with the pixel character sprites and enhances the simplicity of the game. The size of the text is also sized appropriately for the players to read and is easy on the eyes. The font used is from fontsquirrel.com, which is also referenced below in the documentation.

**Canvas Drawing/Animation:** Most of the game's assets are drawn into an HTML5 <canvas> tag utilizing the Canvas 2D drawing API. The only game assets that are not drawn into the <canvas> tag are the start button, control button and play again button, which are displayed through <img> tags. Images, such as background, controls, and player sprite sheets, are drawn into the <canvas> tag with ctx.drawImage(). The platforms and ball are drawn using canvas' standard shape drawing methods. An image animation is implemented through the character sprite sheets, in their idle and running animation states. Player animations are also smoothed with dt, "delta time". Animations are canceled with window.cancelAnimationFrame(), especially when the game is paused or the game browser loses focus. When a player is carrying the ball, the ball becomes active, which will draw a simple particle system emitting from the ball. As mentioned before, the sprites, color palette, and visuals of the game really brings together that simplistic and pixely feel of the game.

**Interaction Requirements:**

**Controls:** Both players can control their respective character sprites on the screen using keyboard controls. Controls are easy to use, in that basic movement is mapped to either WASD keys or arrow keys. Player ability to interact is mapped to a single key, G or numpad\_0. A "key daemon" is used to capture multiple key inputs simultaneously so both players have intuitive and responsive key inputs.

**Usability Requirements:**

The game will pause/unpause with window.onblur and window.onfocus. In terms of teaching, the user controls are intuitive; game controls are easily accessible and displayed to players through the controls UI. The game states are also very intuitive as well; start menu, pause menu, in-game state, and game over screen are clearly shown. While in-game, both players’ scores are also displayed in their respective sides of the screen. There is also an in-game timer that is constantly displayed so players know how long the game lasts until a winner is declared. In terms of difficulty, the game is fairly easy. I feel the controls are easy and intuitive and the gameplay is repetitive. Therefore, once the players grasp the gameplay, they can easily play again. Mouse and onscreen UI buttons are used rather than interface keyboard shortcuts. When the user hovers over a UI button, it is visually clear that that button is in focus because the button will be underlined. In terms of screens, there is a title/introduction screen, as well as controls screen. There is also a main in-game screen and a game over/play again screen. Finally, the developer's name is also displayed in the title/introduction screen.

**Experience/Game Design Requirements:**

Although my initial game design/plan did change over the course of production, I feel once I figured out what my final game decision was, I was able to stick to that game plan and implement a fairly playable and approachable game. The game requires two players as it is a local multiplayer game. The players' objective is to hold onto the ball as long as possible and stay within their designated regions. The player with the most points by the end of the time limit is the winner. Players have the ability to jump and fall onto various platforms as they attempt to avoid from or chase the other player. Players may also "punch" and steal the ball away from the other player. The character sprites are realistic in the game world as they demonstrate very simple running and idle character animations.

**Coding Requirements:**

I have not used any JS library without pre-approval. In terms of JS code, my game preloads images. An object literal is used through the ball object. A module pattern is used as functions and variables are all encapsulated inside a game app module. At least two function constructors are used, specifically in creating the players and platforms. In terms of code conventions and practices, my code is logically separated into appropriate JS files so that it is not all encapsulated into one single file. I have also factored out nearly identical code to avoid violating D.R.Y (Don't Repeat Yourself). "use strict" lines are also placed at the top of every JS file. Function constructors are called with the "new" keyword and are capitalized, whereas all other variables and function names begin with a lowercase letter. Declaring variables uses the "var" keyword as well. Every JS statement ends with a semi-colon and all functions are commented with descriptions of what they do. Console.log()s are also factored out.

**Above and Beyond:**

For above and beyond features, I implemented a level/map selection screen. Players can choose from 3 different levels/maps to play in. I tried to vary the layout of the maps so that player movement does not feel repetitive. As mentioned before, although the game is very simplistic and graphically, there might not be a lot going on, I feel the theme, style, look and feel of the game is really consistent and comes together really well.

**What went Right:**

In terms of what went right, I feel I was able to make a very simplistic and easy-to-play game. The look and feel was exactly what I was striving for. I also feel since the game is visually very simple and I mostly focused on the physics of the game, I feel the functionality came out really well, despite having limited time due to my complete turnaround in game ideas during the production process.

**What went Wrong:**

In terms of what went wrong, I feel I spent quite a while trying to get my character sprites to display properly frame by frame. At times, it appears the player disappears and I think this has to do with trying to draw different sprite images depending on the player’s state machine. If more time permitted, I would have liked to add more features and physics into the game. Since I drew most of my inspiration from Playstation's *Bara Bari Ball,* I would have liked to add ball physics. For example, if the player who has the ball is punched by the other player, but the other player has not yet picked up the ball, the ball might have a gravitational force pulling it downwards until it lands on a platform or falls off the map, which will respawn the ball. Another functionality I would have liked to add is, maybe the player who has the ball can voluntarily drop the ball allowing them to punch the other player and stagger them for a short duration of time, or maybe, the ball carrier has a cooldown ability to “push” the other player away for a short duration of time; essentially giving the ball carrier the ability to further interact with the other player instead of just running away. Lastly, I would also like to incorporate the “S” and “Down Arror” keys to allow the players to drop down from the platform they are currently standing on.

**Self grade:** I would grade myself an overall grade of 94% out of 100%.

**Resources:**

* punch sound: https://www.youtube.com/watch?v=z2M9y9RfeZ8&t=22s
* bubble sound: https://www.youtube.com/watch?v=FCDeiBFju6M
* background song: https://www.youtube.com/watch?v=DJ6CcEOmlYU
* character sprite image: https://opengameart.org/content/classic-hero
* font: http://www.underware.nl/fonts/unibody/download/
* arrow keys: http://www.101computing.net/wp/wp-content/uploads/arrowKeys-300x205.png
* WASD keys: https://img.clipartfest.com/e850c93e46c5904f1cd2141a5ed85071\_or-arrow-keys-rotate-canvas-wasd-key-clipart\_288-187.png
* numpad\_0 key: https://wpclipart.com/computer/keyboard\_keys/number\_pad/computer\_key\_0\_Ins.png
* g key: https://img.clipartfest.com/d5ed800e7ae1107913793c7d2cae88a4\_clipart-computer-keyboard-keyboard-key-clipart\_531-511.png
* p key: https://www.techonthenet.com/clipart/keyboard/images/letter\_p.png
* sprite animation tutorial: http://www.williammalone.com/articles/create-html5-canvas-javascript-sprite-animation/

**Inspirations:**

* **initial idea:** https://www.youtube.com/watch?v=D6asQbA4cP8
* **final idea:** https://www.youtube.com/watch?v=7l\_l\_THQDwE