Programming for Realtime – Technical Document  
  
Made by Nikolay Ivanov- CMV1G – 376469

**Game concept description**

The game concept is about controlling the character Mario to reach the end of the level and rescue Princess Peach. For the purpose of the prototype, the game is completed at the point where Mario reaches the end of the first level. The level Mario traverses includes various monsters and obstacles to avoid (goombas, pitfalls). He must use his abilities and collect mushrooms to increase his chances of success and reach the end of the level.

## **List of game elements**

* Tile map
  + Floor
  + Bricks
  + Pipes
  + Flag pole
  + Castle
* Sprite sheets
  + Small Mario
  + Big Mario
  + Coins
  + Goombas (hostile mob)
  + Mushroom tiles
* Sprites
  + Mushrooms
  + Flag
* Sound effects
  + Coin pick-up
  + Power-up appearing
  + Power-up picked up
  + Stomp (hitting a hostile mob)
  + Small Mario jump
  + Big Mario jump
  + Power-down (hit by a hostile mob)
  + Flag pick-up
  + Brick smash
* Background Music
* HUD
  + Score
  + Timer
* Scenes
  + Game loop
  + Win
  + Lose (hit by goomba when you are small Mario)
  + Lose by Timer going zero

Every game asset which does not belong to me was linked with respect to their creators on the web-page of the game and they were free to the public.

## **Implementing the game elements**

The tile map was prepared before-hand using Tiled, a software for creating tile maps and exporting them to CSV and JSON formats. I used the JSON format of the tile map to create the different visual layers of the game such as the background (clouds, castle, grass). I also used different layers for the collisions with the player such as the brick tiles, floor and stairs. I then targeted the layers I wanted by using their JSON array indexes and enabled the collision, so that the player’s physics body can interact with them.

The sprite sheets I used for animating the character, hostile mobs, coins, mushroom tiles and bricks being smashed had predefined width and height for each frame that was displayed at a time and for elements that were on the background such as goombas, coins and mushroom tiles, I had the property to allow looping the frames. I also enabled the sprite sheets to have a physics body, so that they would interact with the game’s tile map and each other by using the Phaser Arcade physics.

The sprites are the same as the sprite sheets but have no animations, so I only included the sprite image path and width and height properties. I mainly used that for the mushrooms because they were static and did not require animations.

The sound effects were as simple as preloading the sound and using its key and “play” method on specific times such as collecting coins or smashing bricks.

The background music always runs in the background and is initialized the same way as a sound effect but is looped and runs the player presses the “space” button.

Text elements were added to the game that were used to display how much score the player collected in the game (interacting with other physics bodies). I also added a timer that goes down every second until it reaches zero, where the game ends and shows the timer passed end screen.

Finally, the game contains three scenes, the main one being the game itself and the other two start when Mario gets hit by a goomba when he is small or falling off the world and the other end-game screen is the timer hitting zero, which is simply a game scene being started at different moments in the game.