

## GTA I Project : Minion Edition Report

Farida Sarhank

Yasmeen Mohamed

Menna Emad

### Wednesday Nov 9

We sat and discussed the theme of our project. Due to the fact the project emphasized creativity we thought that creating a game with the same functionality but with a completely different theme and elements would set us apart completely as well as give us more to work with in terms of map design. So, after expressing to each other that we thought a candy map would be cute , the minion seemed to be the best character to fit in that vision.

We then decided to look up and save the pictures of the minion, and obstacles and obtained the street and borders from the asset link that was provided in the assignment. After that , we decided it was best to set the view and scene which showed us our then empty map. Everything seemed to be going well and we were excited for what was yet to come

### Thursday Nov 10

We decided it was now time to start placing items on the map. In order to do that we attempted a txt file to create our game board. We took inspiration from the tweety game example that was provided and created a map with borders and obstacles that were strewn across the map. We then also decided that our play should move on -1s , the house should be in the dead center and the obstacles would be placed in values of 70 with the minion not being able to stand on anything greater than 0.

To place the items, we used the else if conditions with the numbers we preferred and were ready to see our map. We unfortunately got a null pix map that was a black screen similar to the one we had seen yesterday.

After researching, we realized that the game could not read any of the images and thus produced no output on the scene. This led to us implementing a resource file which finally produced output.

### **Saturday Nov 12**

After all the obstacles were set and the board finally appeared we focused on the minion , power pellet , and bullet.

We used the move function provided in the mock game as to make the minion fully functional and positioned him at 7 7 as that was the middle of the board. We then implement the colliding item functions on the pellet and bullet and set their positions.

Following that, we thought we had to overlay pictures as the street was set to occupy all -1 spaces but we didn't get how to get other elements on top of the street and thus thought we had to overlay picture.

Through rigorous attempts at thinking our way through this we figured if we set the background as the street, we chose it would alleviate our problem and that worked expertly and made our game look cleaner.

We also had an issue that our board was too big and thus required us to scroll the actual scene to see the full scene which always glitched the minion.

### **Sunday Nov 13**

We realized that the only way to fix the minion glitch and odd need to scroll in the game was by decreasing the size of the array and our map.

This was devastating as felt like we were undoing all the effort we had done as this decrease in size yielded multiple errors and ruined every position we had tirelessly worked to perfect. However, we eventually managed to get a size that worked for us through trial and error of the array size and playing with the length and width of the widget.

### **Monday Nov 14**

We worked on changing the image of the minion when he touched the gun and the teddy bear. This was tough as we didn't have the slightest clue on how to tackle such a task.

We all brainstormed and asked ourselves whether there was a timer function in QT. After finding it through documentation and finding the QTimer function of connect we were underway to achieving the desired outcome. The only issue was that time connect needed us to create a function that returned the minion but the function was fairly easy to implement. The function's only issue was that it wouldn't actually be read by the compiler and thus the picture never actually returned. We realized the issue may be that the function (minion return) needed to be placed in public SLOT. This did in fact work and we did it with every function to ensure it was used.

### **Wednesday Nov 16**

After reassessing the project requirements, we had noticed that our map was in no way a maze due to the way our obstacles were strewn and the blatant lack of paths. This meant we had to reconstruct our txt file and make it into an interconnected map which was harder than anticipated and needed immense trial and error but we eventually got there. We also decided to add our enemies and their random movements. We were unsure of how to do this at first but after playing around with the player's movement code we eventually figured it had to do with the RAND() function and a modulus of 4 due to the four directions and enemy could take. The only issue we now faced was that our kill enemy function was not working although the map seemed accurate but the timer connect never allowed us to enter the function.

### **Thursday Nov 17**

The kill enemy function was soon forgotten and we decided to work on implementing the player's lives. Through watching videos on youtube, it was specified that the way to do so was to create a global variable but the only way to do that was to add all our code in main into a cpp called game. This obliterated our code and nothing worked anymore. It was our biggest failure at this point and we were taken aback to say the least. However, we learned a valuable lesson to not implement code we did not understand. We thankfully had a backup file.

### **Nov 18th Friday**

Seeing as we had a backup file, we undid everything we did the day prior, and decided to retrieve our efforts on the kill enemy function which did not end up working as something was wrong with the timer. stop function. This led to kill enemy becoming an infinite loop and only stop after both enemies died from one banana. We tried playing around with it in every way possible but nothing had

worked and thus we all went home more demotivated than before with nothing but a broke function.

### **Saturday Nov 19th**

Although we exhausted all our efforts in kill enemy, we had one more idea which was to put the timer in the function. This finally worked and we felt as though we were regaining some much-needed footing back.

With our confidence back, we attempted the reset function which was a success from the start with a few bugs here and there.

### **Sunday Nov 20th**

We felt close to the finish and thus started making the exit gate. It unfortunately had an error where it appeared after only killing the first enemy. We also fixed a flaw in our newly created reset function where the minion upon resetting after dying once would spawn in the middle then appear in the last place it died after the key event. It was fixed by adding a row=7 and a column=7 instead of set position in our reset function which we randomly figured out after trial. Moreover, we also created a reset all function that did everything the reset function did except it also added the power pellet. The issue is that it would cause the entire code to crash upon being used. After immense debugging we noticed that the issue was with the fact we forgot to add a this->t=t for the power pellet. It also taught us that the most miniscule of mistakes reverberated through the code immensely.

### **Monday Nov 21**

We fixed the door by adding a counter for enemies and made it functional by removing the player upon it's collision. We also displayed the lives on the screen and made it decrease with every death. As well as, cleaned up the code and added comments. We also did the win and lose output which we struggled in as adding Qimage to our scene caused the code to crash so we figured classes was the way to go, and it finally worked. Finally, to end a good day's work we created the functionality of invincible and normal which worked great using set plain text.

### **Tuesday Nov 22**

Finally, with the project day amongst us we did the last two things we had been avoiding for the entire course of our project : the functionality of shooting the closest enemy to the player and the timer of the power pellet. After hours of trail

and error and raiding stack overflow, the answer arose in the QTimer documentation with the function of remainingtime and the creation of the function upatecountdown which stack overflow aided us with. As for the distance, it had to do with simple maths and the school taught Manhattan equation of distance. After both worked we started altering our fonts and font colors and checking if we wanted to change anything.





















