



National
College of
Ireland

National College of Ireland

BSc(Hons) in Computing

Software Development

2020/2021

Damien Fleminks

X16102118

X16102118@student.ncirl.ie



Technical Report
Mid-Term

Contents

Executive Summary	2
1.0 Introduction.....	2
1.1. Business Challenge	2
1.2. Background.....	3
1.3. Competitor Analysis	3
1.4. Research Process.....	5
1.5. Technology	6
2.0 System.....	7
2.1. Requirements	7
2.1.1. Use Case Diagram.....	7
2.1.2. Functional Requirements	8
2.1.2.1. Requirement 1: Create Avatar	8
2.1.2.2. Requirement 2: Interact with Chatbot NPC	8
2.1.2.3. Requirement 3: Walk around the world.....	9
2.1.2.4. Requirement 4: Text Chat with other players	9
2.1.2.5. Requirement 5: Audio Chat with other players	10
2.1.2.6. Requirement 6: Play a mini game with another player.....	10
2.1.2.7. Requirement 7: Send notification Request to another player	11
2.1.2.8. Requirement 8: Create a private room with players	11
2.1.3. Requirement 9: screen sharing in private room	12
2.1.3.1. Requirement 10: Configure a Game	12
2.1.3.2. Requirement 11: Assign players to a Game.....	13
2.2. Design & Architecture	13
2.3. Implementation	14
2.4. Graphical User Interface (GUI)	19
2.5. Testing.....	22
2.6. Evaluation	23
3.0 Conclusions.....	26
4.0 References.....	27
5.0 Appendices.....	27
5.1. Reflective Journal & Supervisor Interactions	29
5.2. Project Plan	33
5.3. Project proposal	34

Executive Summary

The solution is called OfficePlay and offers a virtual office experience for businesses who have employees working from home. Employees can meet up in a virtual world and walk around, talk, collaborate, have private meetings, play games with each other and interact with chat bots for support.

We started with a business challenge of connecting employees who are working from home because of the lockdown. We conducted brainstorm sessions and a competitor analysis to find ideas and use cases. We then presented this idea to real potential end customers and conducted user research to define the most valuable requirements for our solution. The requirements are defined in detail in chapter 2.1 and the research process and documents can be found in chapter 1 and appendix a.

The solution is built upon the Salesforce platform, and this required to overcome interesting technical challenges for which we designed design patterns and algorithms. In chapter 2.2 and 2.3 a detailed review of those challenges and the overall architecture and implementation is discussed. In chapter 2.4 a current draft of the user interface is presented to give a visual understanding of the different use cases. In the last two chapters you can find the references and appendices.

1.0 Introduction

1.1. Business Challenge

As someone who is personally affected by the lockdown measurements and been working from home for over 7 months now, I can easily put myself in the shoes of the end-user. Even though working remotely has its benefits, we are missing real human contact and informal discussions throughout the day with colleagues during a coffee break.

The idea started with the concept of a virtual world much like a video game to address challenges with virtual collaboration and meetings. The ongoing digital transformation of businesses to remote working and current challenges in virtual collaboration is the underlining business challenge we want to solve.

1.2. Background

Since the coronavirus outbreak over half the global workforce have been working from home and virtual meetings have become the standard. Recent research conducted by [Future Forum research](#) on the future of remote working concluded that most employees have not been less productive and in some cases increased productivity. Over 80% of surveyed knowledge workers do not want to go back to the old way of working and businesses are transforming to embrace virtual collaboration. Remote working will become the new normal.

Since this opens up new opportunities to innovate in virtual collaboration, we researched new ideas by brainstorming and conducting interviews with remote workers and concluded that we will aim to create a virtual office experience much like a video game in which colleagues can casually meet and collaborate in ways that is currently not easy to do.

1.3. Competitor Analysis

To find more inspiration and research if other solutions are not providing the same solution, we analysed multiple apps offering similar features. During our competitor research we are happy to not find any existing solution to our idea of a virtual game world to collaborate with colleagues which is also build on the Salesforce platform. We did however come up with some new features to add to our app. We will give a few examples of other solutions and analyse how they are different.

Visual Office (Slack App)



[Visual Office for slack](#) is a plugin for the Slack that provides a visualisation of office rooms such as a coffee room or work area that are connected to slack channels, with the app you can see which people are in which rooms (slack channel).

The idea is very similar to what we aim to achieve, however you are not able to move a character inside the room or interact in any way within the world. It only provides a picture of a room showing number of people inside.

Zoom



[Zoom](#) definitely took off as one of the most popular Virtual collaboration tools in the last year during the lockdowns as it provides a reliable and intuitive user interface for cost free video conferencing.

Zoom does tackle a similar business challenge as ours as it provides a way to collaborate virtually, however Zoom only offers video conferencing and you would need to setup a meeting and joining it would set an expectation to be fully engaged, you cannot just randomly walk into someone and have a small chat or walk around and talk with different people.

Roomkey



[Roomkey](#) comes very close to our own solution as it provides a virtual world where you can create an avatar and host virtual meetings and events. They are mostly focussed on panel discussions, conferences, presentations and brainstorm sessions.

After testing the application, we were disappointed to discover that you are not able to freely walk around the room, you are basically stuck on the char. For having purposeful presentations and barnstorm sessions it still allows a great and unique experience different than traditional video conferencing tool.

Wonder



[Wonder](#) provides a great solution unique from other virtual collaboration tools and an important use case we will also include in our app. They provide video conferencing with multiple isolated areas, you can create a circle in which people can drag their avatar in and you would only share your video with people inside the circle.

1.4. Research Process

As part of the ideation process I conducted a brainstorm session using a mind mapping framework (*Appendix A*) to explore creative ideas. Starting with the business challenge to solve the need for human connection using a digital platform, an idea was formed to develop a virtual world where you can create an avatar with which you can walk around and interact with other people. In this virtual world we could implement use cases such as business conferences, meeting with colleagues, have informal discussions throughout the day. Since I work as a consultant in Salesforce, which is currently leading initiatives to transform the digital collaboration in businesses with products like Slack and an integrated CRM platform, I wanted to leverage this platform for the solution.

Requirements gathering user research

Starting with a clear business challenge and high-level use cases it was important to discover clear requirements that would solve real problems. We conducted a user research to really understand the needs and motivations of our potential end users. The results of this user research provided us with the feedback to define and prioritise the exact requirements we needed to implement.

For the research 4 participants had been selected with diverse backgrounds. Two of the participants were working as Salesforce consultants, one is a Manager of Success Managers and the fourth a User Experience Designer. All participants have had to work from home because of the lockdowns and therefore their feedback would be very valuable as they are a perfect example of the ideal end-user.

For all the participants an open interview was conducted, giving them an overview of the business challenge and use case. Questions were also asked about their experience working from home and challenges they faced. Qualitative feedback was gathered which we leveraged to define the requirements.

After the interviews it was concluded that we would provide a virtual office experience which current solutions don't provide. What most participants really missed for example was how easy it was in an office to randomly meet team members and have an unexpected informal chat in the canteen while getting some coffee. Current virtual tools such as Google Hangouts for video chats are limited and you would usually not setup an informal meeting with someone you do not know well. Participants felt more disconnected from their team.

1.5. Technology

The whole solution is developed and integrated on the Salesforce Platform. Salesforce is a solution provider which is leading digital transformation in businesses and offers a range of cloud software which unites the whole marketing, sales, commerce, service, and IT with customer 360 through one integrated CRM platform.

The salesforce platform allows developers to create business applications which integrates with the entire suite of connected apps and offers an app store called [AppExchange](#) where you can distribute your business apps, much like the apple app store but for enterprise apps.

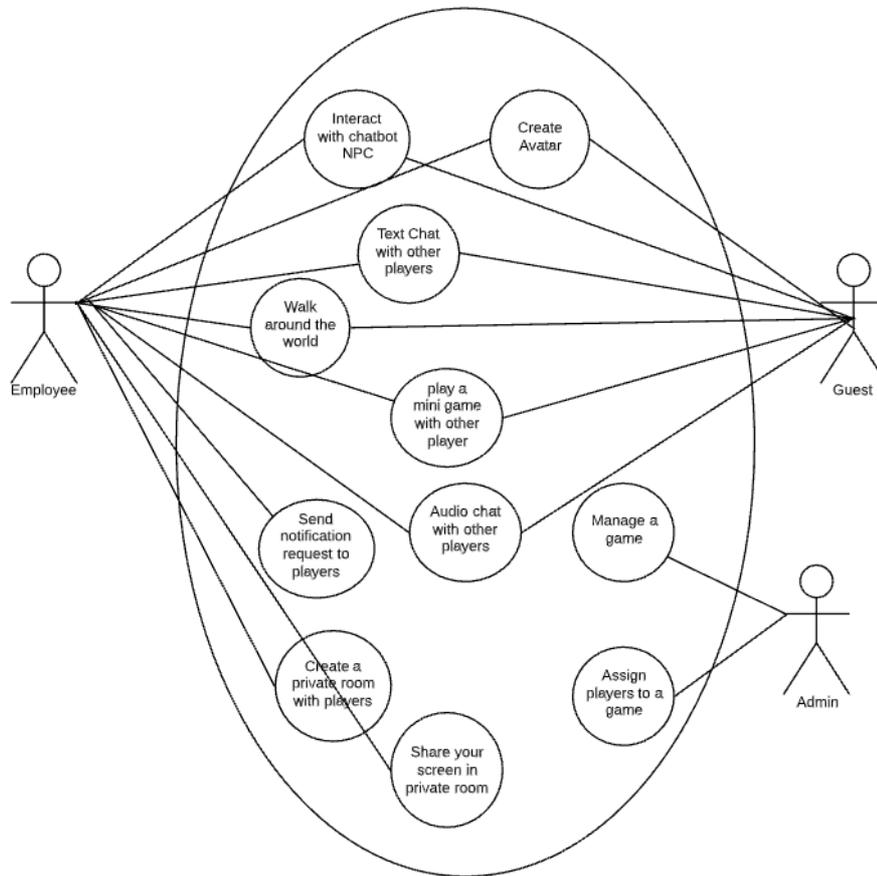
The force.com platform offers a range of programming frameworks to develop a complete solution such as JavaScript and web components for the front-end , for the back-end we can use Apex, which is a language very similar to Java. For some interface elements we will use the lightning web component (LWC) framework in Salesforce such as the initial login and setup screens. The actual game itself will be developed with the WebGL API using Phaser JS in TypeScript.

GitHub: <https://github.com/flemx/virtual-office>

2.0 System

2.1. Requirements

2.1.1. Use Case Diagram



2.1.2. Functional Requirements

2.1.2.1. Requirement 1: Create Avatar

Description

The player will have his own virtual avatar to walk around with and identify himself with. During the start of a new game the player can choose an avatar and name.

Activation

Once the player starts a new game he will be given the option to customise his character

Main Flow

1. The player starts a new game
2. The player can choose between multiple characters to choose as avatar
3. The player enters a name for the player
4. The player submits his choice
5. The system validates the username choice
6. The system saves the player details to the database
7. The player enters the world map

Exception Flow

1. Starts from Main flow 5
2. The systems detect a duplicate or invalid username
3. An error message is returned to the user interface indicating the reason why the username is invalid
4. Player can retry with a new username
5. Once the username is valid, continue in main flow 6

Termination

The player details have been successfully saved to the database

2.1.2.2. Requirement 2: Interact with Chatbot NPC

Description

The player can click on an NPC and start a chat conversation in the chat window. The chatbot NPC is connected to the Salesforce Chatbot API.

Activation

Once the player clicks on an NPC.

Main Flow

1. The player clicks on an NPC
2. The Chat window pops open with a separate tab
3. The NPC bot connects to the Chatbot API
4. The player can interact with the pre-configured chat transcript of the chatbot

Termination

1. The player closed the chat window tab

2.1.2.3. Requirement 3: Walk around the world

Description

The player can freely walk around the office world map by clicking on a tile. The character will automatically walk to the tile using the shortest path to where the player clicked.

Activation

Once the player clicks on a tile on the map that is a valid tile.

Main Flow

- 1 The player clicks on a valid tile on the map
- 2 All the valid tiles are loaded in the Pathfinding algorithm as a graph
- 3 The Pathfinding algorithm calculates the shortest path using the Dijkstra algorithm
- 4 The shorted path is returned and the character walks automatically to the destination

2.1.2.4. Requirement 4: Text Chat with other players

Description

A chat box will be accessible to the player in which they can talk to everyone in the same room and open a private message to a single player.

Activation

Once the player clicks on the chat box

Main Flow

1. The player clicks on the chat box and it pops open
2. In the chat box the player can type a message and submit
3. The message is being pushed by the system to all online players using the streaming API and displayed in the chat box

Alternate Flow

1. The player clicks on another player and presses the button that says private chat
2. A tab opens up in the chat box
3. Start from the main flow

2.1.2.5. Requirement 5: Audio Chat with other players

Description

Players can open an audio channel when they are nearby each other. A circle will be visible once 2 or more players come close indicating that you can hear each other talk inside the circle.

Activation

After 2 or more players come close to each other

Main Flow

1. Once 2+ players come close a circle shows and the microphone and speaker activates
2. Through a third-party API the audio stream will be provided
3. The players can speak through audio

Termination

After the player leaves the circle, the audio will stop and they players are not able to hear each other

2.1.2.6. Requirement 6: Play a mini game with another player

Description

Just like you can often play table football or tennis in a real office, there will be a mini game players can play with each other.

Activation

The player clicks on another player and presses the button that says play mini game

Main Flow

1. The player clicks the mini game button
2. A window pop ups and both players can start the game
3. The game will be of short duration and stops if one of the players have won the game

Termination

One of the players won the game or presses the cancel button of the mini game window.

2.1.2.7. Requirement 7: Send notification Request to another player

Description

From the UI window of all players that are assigned to the game a notification can be send with a message so the other player will get a push notification on their device when they are online.

Activation

The player clicks on another player and presses the button that says send notification

Main Flow

1. The player presses the button that says send notification
2. A window pop ups and the player can write a short message and press submit
3. The system will forward the message as push notification to the other player who will receive this message on their device when they are online

2.1.2.8. Requirement 8: Create a private room with players

Description

The player can host a private room and invite other players to collaborate and communicate in private.

Activation

The player presses the button in the menu that says create private room

Main Flow

1. The player presses the button to create a private room
2. A room is created and saved to the database
3. In the user interface under private rooms, the player can add other players
4. The player adds x players to the room
5. The players that have been added are receiving a message request to join the private room and the room also appears in their private rooms list
6. The player can click on the private room and press go to room
7. The player is teleported to the new room

Termination

The player presses the options button in the private rooms list and chooses to delete the room

2.1.3. Requirement 9: screen sharing in private room

Description

Inside a private room, a virtual monitor will be present that can be used to share content like a PowerPoint slide or a video from a URL, there are also solutions available online to share your screen through a UR.

Activation

The player clicks on the virtual monitor inside the private room and enters a URL to display

Main Flow

1. The player enters a URL after pressing on the menu item of the virtual monitor
2. The website of the given URL will be displayed using an iframe
3. Other players in the room can also see the content of the URL inside the virtual screen

Termination

The player presses the options button in the virtual screen and deletes the URL

2.1.3.1. Requirement 10: Configure a Game

Description

From the admin interface, the administrator can create a new game and select a map to use for the office, there are several options such as maximum number of players or allowing guest users in the game.

Activation

The admin goes to the admin interface and click the add new game button

Main Flow

1. Add new game button is pressed
2. A pop-up form will display to enter details of the game
 - a. The name
 - b. Office Map
 - c. Number of players
 - d. Activate status
 - e. Guest users allowed
3. Once submitted the game is created and visible in the list of available games
4. Same options can be edited

Termination

The player opens the edit screen of the game and presses delete

2.1.3.2. Requirement 11: Assign players to a Game

Description

From the admin interface, the administrator can add or remove player to a configured game, the players will then be able to access the game.

Activation

The admin goes to the admin interface and clicks add new player from a game configuration

Main Flow

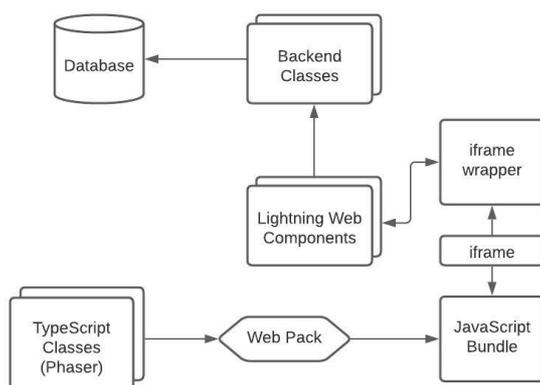
1. Add new player button is pressed from game configuration
2. A pop-up form will display with a field to enter a name
3. Admin enters a player name and the form searches for matching players
4. The admin selects a player and press add
5. The player is added to the game
6. The player will be able to access the game

Alternate Flow

1. The admin presses the option button on an existing player in the game configuration screen
2. The admin presses remove from the option menu
3. The player is removed from the game configuration
4. The player is not able to access the game anymore

2.2. Design & Architecture

High level Architecture illustration



The whole stack is deployed on the Salesforce platform. Within the Salesforce framework we can build front-end solutions with JavaScript and communicate with the back end using Apex programming language which is very similar to Java and from Apex we can interact with the SQL database.

The solution is built upon multiple layers, The first JavaScript layer contains most of the game logic and is responsible for calculating decisions such as pathfinding, animations, rendering game objects and camera movement. This layer is build using Phaser JS framework, which is a light JavaScript framework to provide basic building blocks for game development. This layer is furthermore developed using TypeScript, which is a programming language that compiles to JavaScript and adds typing and more Object-Oriented programming features to JavaScript.

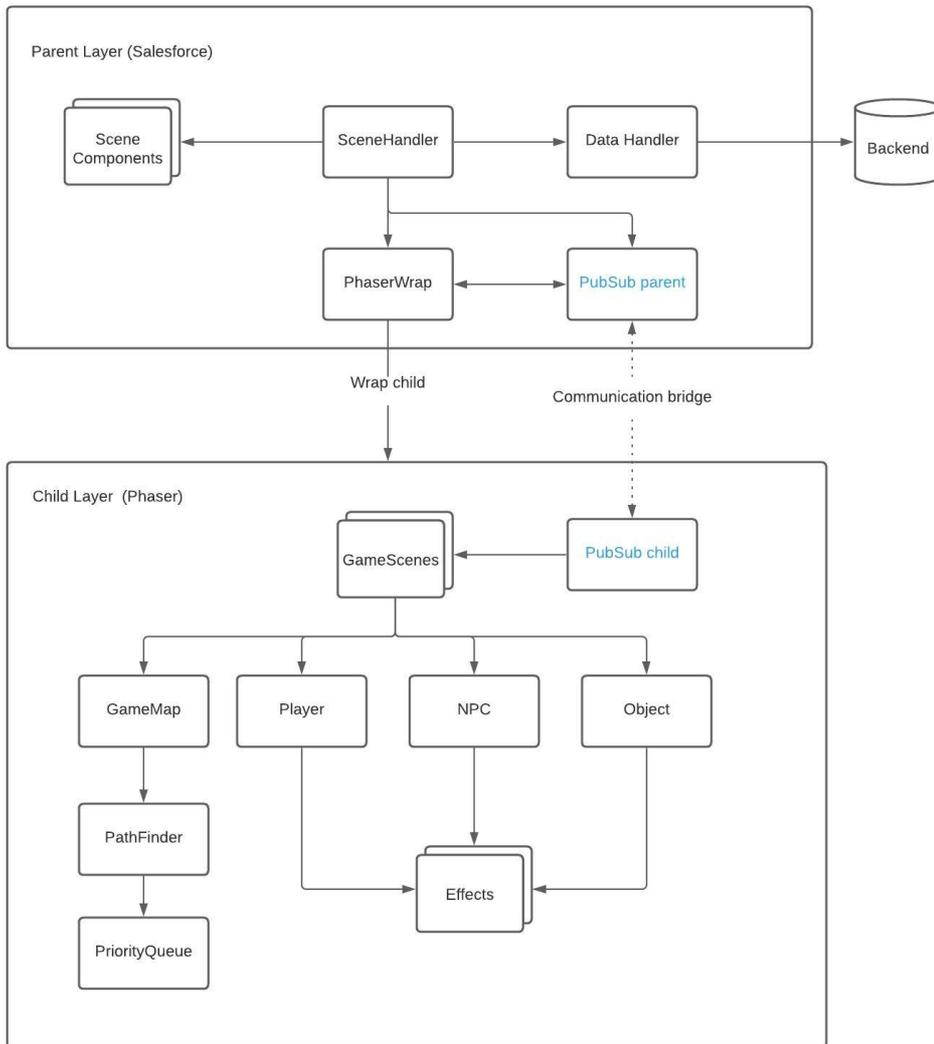
The Salesforce platform also has its own JavaScript framework called Lightning Web Components. The Lightning Web Components framework leverages the web standardized Web Components which allows encapsulation and modularization of the components. The Lightning Web Component framework can be compared with JavaScript frameworks like React and Angular JS. It furthermore offers smooth integration functionality to the Salesforce API's. For that reason, we decided to leverage this framework to build some of the UI elements for the game such as a Chat box. These UI components are then placed over the Phaser WebGL canvas.

There are some challenges to overcome with this design choice. The first problem is that the Phaser JS library could not be natively imported into the Lightning Web Component framework due to the Salesforce Lightning Locker service. The Salesforce platform has built in security on the JavaScript layer and certain JavaScript API functions are blocked in the Lightning Web Components framework. To overcome this limitation, we can load the JavaScript of the compiled TypeScript and Phaser JS code from an iframe and render it inside the Lightning Web Component. To efficiently load all the TypeScript classes into the web component we leveraged a package called Webpack which converts all relevant TypeScript classes and imported modules into a single JavaScript file so we can easily load this file and provide configurations to make the development easier.

The next challenge to overcome in the architecture was the communication between the parent Web Component and the JavaScript contained inside the child iframe. The game logic had to send signals for example to load a chat window or the parent needed to be able to send data retrieved from the Salesforce API down to the game. A custom-made design pattern has been designed to overcome this challenge, which we will explain in more detail in the next chapter.

2.3. Implementation

High level UML illustration



In the UML illustration above, a high-level overview can be observed to show the relationship between the most important Classes. The lower child layer classes are developed in TypeScript and contains most of the game logic. The Game logic leverages the Phaser framework that provides functionality to support the development of game mechanics. The GameScene class in the top of this layer maintains the state of the game and imports all the different classes to render the objects and logic of the game. There are some classes that contain more complex algorithms and interesting design patterns which I will highlight in more detail.

PathFinder Class Algorithm

```

3
4  /**
5   * Pathfinder Class
6   * This class can generate a graph from the tile map with all possible paths
7   * by looping through all the map tile layers
8   * and calculates the shortest path using Dijkstra algorithm and a binary heap priority queue
9   * @ Damien Fleminks
10  */
11  You, 11 hours ago | 1 author (You)
12  export default class Pathfinder {
13
14    private prioQueue: PriorityQueue;
15    private adjacencyList: Node;
16    private distances: Record<string, number>;
17    private previous: Record<string, string | undefined>;
18
19    constructor(coordinates: MapCoordinates){
20      this.adjacencyList = this.generateGraph(coordinates);
21      // Keep track of the distances from starting node to goal node
22      this.distances = {};
23      // // Keep track of the paths
24      this.previous = {};
25      // Make a priority queue of the distances
26      this.prioQueue = new PriorityQueue();
27    }

```

The Pathfinder Class contains the most complicated algorithms developed so far, above is a snippet of the code, the full classes can be found in the GitHub repository. This algorithm is used in the game when you want to move the character on the map from location A to B. The game map contains some locations which contains collisions so the character cannot walk there, this algorithm can find the shortest path possible by manoeuvring between the collisions. The algorithms start by taking in a 2D Array of the map with Boolean values for the tiles containing collisions.

```

/**
 * Return a graph with all possible paths,
 * all coordinates to easily look them up
 * and a list of the illegal collision coordinates
 * @param (*) coordinates
 */
private generateGraph(coordinates: MapCoordinates): Node {
  const graph: Node = {};
  // Generate vertexes from map dimensions
  for (let y = 0; y < coordinates.length; y += 1) {
    for (let x = 0; x < coordinates[0].length; x += 1) {
      const edges = [];
      // Create all possible neighbours with associated cost
      const neighbours: Array<NodeNaighbour>= [
        { [`${x - 1},${y}`]: { coor: { x: x - 1, y }, cost: 2 } }, // walk left
        { [`${x},${y + 1}`]: { coor: { x, y: y + 1 }, cost: 2 } }, // walk down
        { [`${x + 1},${y}`]: { coor: { x: x + 1, y }, cost: 2 } }, // walk right
        { [`${x},${y - 1}`]: { coor: { x, y: y - 1 }, cost: 2 } }, // walk up
        { [`${x - 1},${y + 1}`]: { coor: { x: x - 1, y: y + 1 }, cost: 2.5 } }, // walk diagonally left-down
        { [`${x + 1},${y + 1}`]: { coor: { x: x + 1, y: y + 1 }, cost: 2.5 } }, // walk diagonally right-down
        { [`${x - 1},${y - 1}`]: { coor: { x: x - 1, y: y - 1 }, cost: 2.5 } }, // walk diagonally left-up
        { [`${x + 1},${y - 1}`]: { coor: { x: x + 1, y: y - 1 }, cost: 2.5 } }, // walk diagonally right-up
      ];
      // If tile coordinate is a valid move, add a node with edges to valid neighbouring adjacencyList
      if(!coordinates[y][x].illegal){
        graph[`${x},${y}`] = neighbours.filter((naighbour: NodeNaighbour)=> {
          let neighbourY = neighbour[Object.keys(naighbour)[0]].coor.y;
          let neighbourX = neighbour[Object.keys(naighbour)[0]].coor.x;
          try {
            return !coordinates[neighbourY][neighbourX].illegal;
          }
          catch(e) {
            // catch inaccessible undefined exception for non-existing neighbours
          }
        });
      }
    }
  }
  return graph;
}

```

From the 2D array with coordinates, a weighted graph data structure is constructed which will generate an adjacency list object containing all vertexes with possible steps from every valid tile the character can walk. Using a Dijkstra algorithm in combination with the PriorityQueue class, which is designed as a binary heap priority queue, the optimal path is calculated and returned. This is a very important algorithm to make all the characters move on the map and the most challenging to design

Dijkstra Algorithm function logic

```
93  /**
94   * Dijkstra algorithm that iterates through a graph
95   * and find the shortest path for a given start and destination vertex
96   * @param {*} start
97   * @param {*} goal
98   * @param {*} graph
99   */
100 public findPath(start: Coordinate, goal: Coordinate): Coordinate[] {
101   // Initialise new data structures
102   this.initDijkstra(`x${start.x}y${start.y}` as const);
103   let smallest: NodeKey | undefined;
104   const path: Coordinate[] = [];
105
106   while (this.prioQueue.getValueNum()) {
107     smallest = this.prioQueue.dequeue().value as NodeKey;
108     // Stop algorithm when it has found the shortest path to the goal
109     if (smallest as string === `x${goal.x}y${goal.y}` as string) {
110       while (this.previous[smallest]) {
111         path.push(this.formatCoord(smallest));
112         if (this.previous[smallest]) {
113           smallest = this.previous[smallest] as NodeKey;
114         }
115       }
116       break;
117     }
118     if (smallest || this.distances[smallest] !== Infinity) {
119       for (const neighbour of this.adjacencyList[smallest]) {
120         // Find neighboring node
121         const nextNode = this.adjacencyList[smallest].filter(
122           (v) => Object.keys(v)[0] === Object.keys(neighbour)[0],
123         )[0];
124         // Calculate new distance to neighboring node
125         const candidate = this.distances[smallest] + nextNode[Object.keys(nextNode)[0]].cost;
126         const nextNeighbor = Object.keys(nextNode)[0];
127
128         if (candidate < this.distances[Object.keys(nextNode)[0]]) {
129           // update new smallest distance to neighbor
130           this.distances[nextNeighbor] = candidate;
131           // Updating previous - how we got to neighbor
132           this.previous[nextNeighbor] = smallest;
133           // Enqueue in prio queue
134           this.prioQueue.enqueue({value: nextNeighbor, priority: candidate});
135         }
136       }
137     }
138   }
139   return path.concat(this.formatCoord(smallest as NodeKey)).reverse();
140 }
```

Publish Subscribe Design pattern implementation

```
you, 2 weeks ago | author (you)
export default class PubSubChild {

  private callbacks: Record<string, Set<Function>>;

  constructor() {
    //this.lexOrigin = new URLSearchParams(window.location.search).get('origin');
    this.callbacks = {};
    // Set event listener for received messages
    this.callbackHandler();
  }

  /**
   * Subscribe for message events and execute callback method
   * @param callback
   * @param eventName
   */
  public subscribe(callback: Function, nameEvent: EventName): void {
    const eventName: string = EventName[nameEvent];
    if (!this.callbacks[eventName]) {
      this.callbacks[eventName] = new Set();
    }
    this.callbacks[eventName].add(callback);
  }

  /**
   * Send messages to the parent Lightning Web Component
   */
  public publish(message: EventMessage): void {
    window.parent.postMessage( message, '*' );
  }

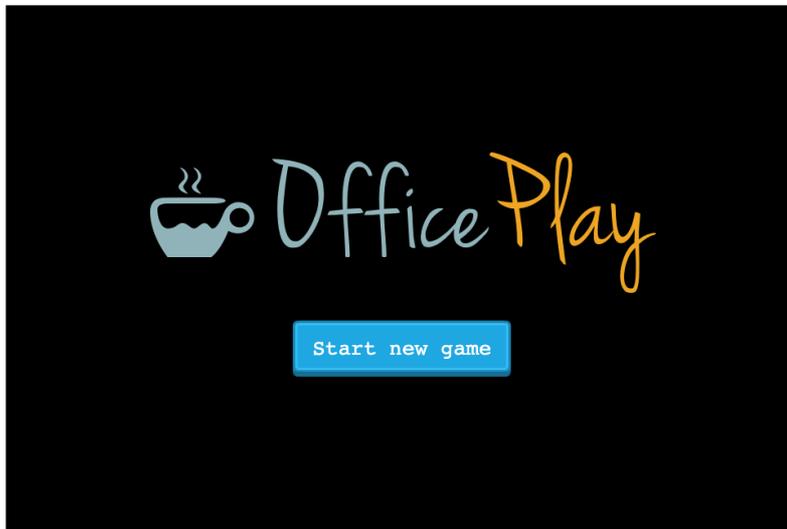
  /**
   * Handle message events
   */
  private callbackHandler(): void {
    window.addEventListener("message", (e) => {
      if (this.callbacks[e.data.eventName]) {
        this.callbacks[e.data.eventName].forEach(callback => {
          try {
            callback(e.data.data);
          } catch (error) {
            console.error('Error in callback function', error);
          }
        });
      }
    });
  }
}
```

In the previous chapter we explained one of the challenges being that we have these two layers which are decoupled from each other through an iframe and this makes it difficult for the functions to communicate between each other. In JavaScript you can broadcast data through the DOM using Events, which can be picked up by event listeners that can conditionally execute logic on certain events. To broadcast events to an iframe or a parent, you can send a MessageEvent using the Window.postMessage() function, which is the only method allowing cross-origin communication.

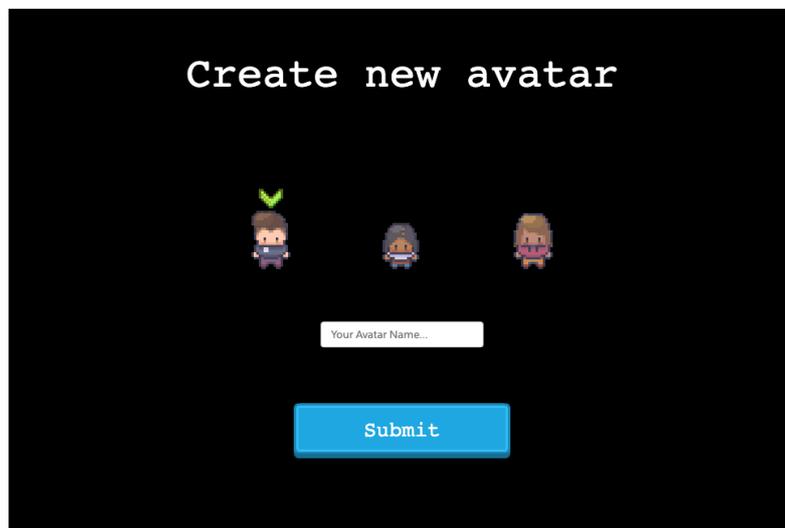
Writing many of these methods throughout the code would make it difficult to keep track of all the events since we would need many of these for different type of events. To design an elegant solution, we implemented a custom variation of the Publish Subscribe (PubSub) design pattern. The PubSub pattern provides a framework to allow scaling and loose coupling between the receivers and senders of data. In our example we have combined this pattern with the event listeners and postMessage functions in JavaScript so we can subscribe to an event type and save a callback function from the Class that calls the subscribe function. When we publish the event using postMessage on the other layer the event is picked up and the callback function is run. In the top illustration of this chapter ‘High level UML illustration’ you can see how the PuSub classes connect the layers with a communication bridge.

2.4. Graphical User Interface (GUI)

While we are still developing some of the screens, we have drafted and developed some components which would give a good understanding of our use cases. We will highlight the most important wireframes below.



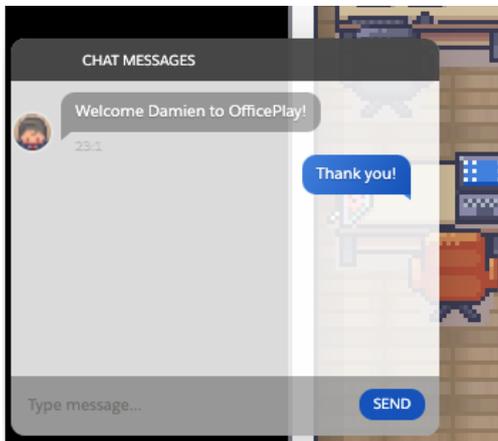
The start screen of the game a new player is presented with.



The character creation screen for new players.



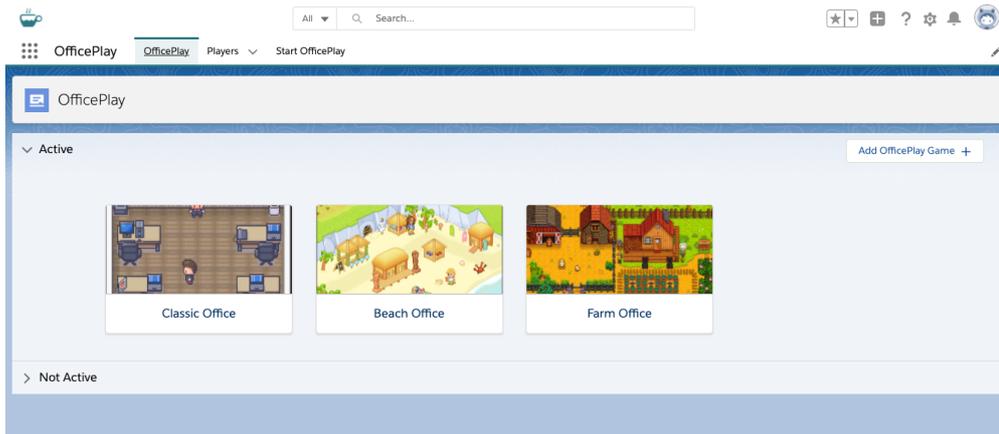
An overview of how the office map would look like.



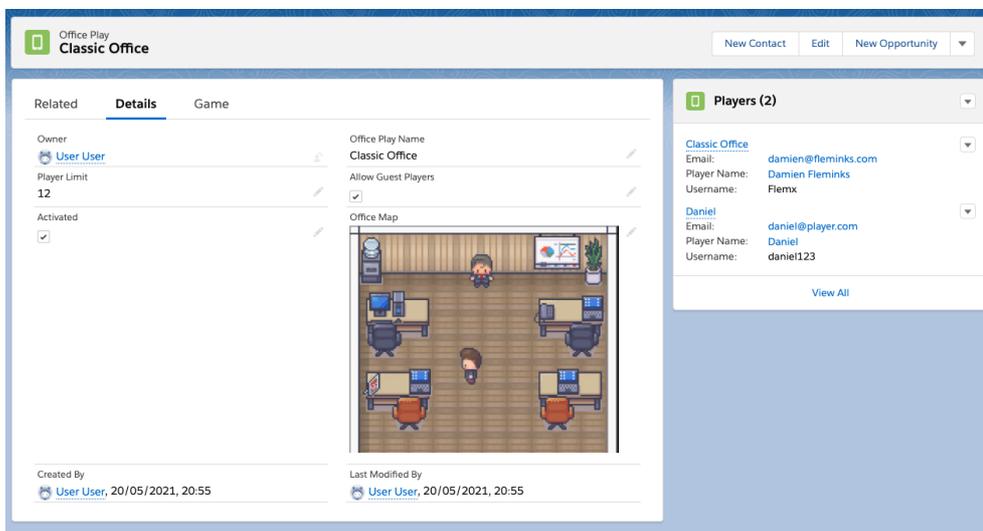
The chat box interface, which will be leveraged for multiplayer interaction and talking to the NPCs.



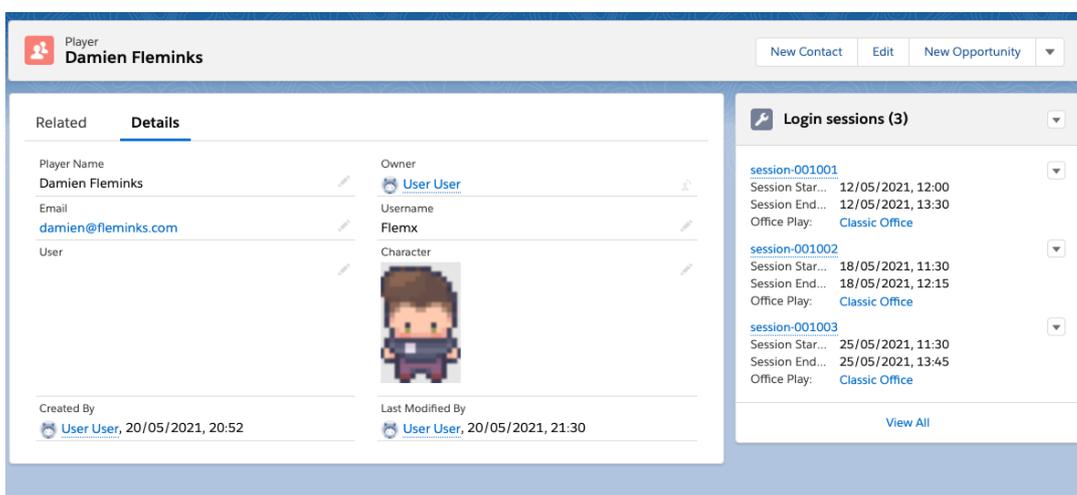
Lounge areas for colleagues to hang out and play games.



Admin overview to manage all games that are active in the company.



Configuration interface of a single game.



The player details and login sessions.

2.5. Testing

In order to deploy production code to the Salesforce platform, working unit tests must be included in the application back-end code with a minimum code coverage of 70%. The reason for these strict requirements is to protect the resources so other customers will not be affected because multiple customers share the same servers.

Within the Salesforce platform there is a user interface in which you can select your test classes and it will run all your unit tests and show the total code coverage, you can run this manually to evaluate your testing code before deploying your app. When you deploy your application code, all these tests will run automatically and it will block your code from deploying if the testing code contains errors or doesn't meet the code coverage requirements, below screenshot illustrates the testing user interface taken from our application. We achieved a 93% code coverage of all the Apex application code and all the test classes have passed.

Testing GUI

The screenshot shows the Salesforce testing interface. At the top, there's a table with columns: Class, Method, Duration, Result, Errors. Below this, there's a 'Class Code Coverage' table with columns: Class, Percent, Lines. The bottom part of the screenshot shows a 'Tests' tab with a table of test runs, including columns for Status, Test Run, Enqueued Time, Duration, Failures, and Total. To the right of this is an 'Overall Code Coverage' table with columns: Class, Percent, Lines.

Class	Method	Duration	Result	Errors
OfficePlayUtilityTest	getGamesTest	0:00	Pass	
PlayerEventTest	publishChatTest	0:00	Pass	
PlayerEventTest	publishPlayerTest	0:00	Pass	
PlayerUtilityTest	createPlayerTest	0:00	Pass	
PlayerUtilityTest	getPlayerTest	0:00	Pass	

Class	Percent	Lines
OfficePlayUtility	100%	2/2

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	7071X00001mpx54	Mon Aug 02 2021 11:47:07 GM...		0	2
✓	OfficePlayUtilityTest			0	1
✓	getGamesTest		0:00		
✓	PlayerEventTest			0	2
✓	publishChatTest		0:00		
✓	publishPlayerTest		0:00		
✓	PlayerUtilityTest			0	2
✓	createPlayerTest		0:00		
✓	getPlayerTest		0:00		

Class	Percent	Lines
Overall	93%	
OfficePlayUtility	100%	2/2
OfficePlayUtilityTest	0%	0/1
PlayerEvent	100%	6/6
PlayerEventTest	0%	0/1
PlayerUtility	100%	23/23

PlayerUtilityTest Test Class

```
/**
 * A testSetup method to setup test data that will be reused by multiple test methods
 * We will test one Office_Play_Config__c record with and one without a player
 */
@testSetup
    You, a day ago • Implemented unit tests for PlayerUtility
static void setup() {
    String userId = UserInfo.getUserId();
    Office_Play_Config__c office1 = new Office_Play_Config__c(
        Name = 'TestOffice1'
    );
    Office_Play_Config__c office2 = new Office_Play_Config__c(
        Name = 'TestOffice2'
    );
    insert office1;
    insert office2;
    Player__c player = new Player__c(
        Name = 'TestPlayer',
        User__c = userId,
        Office_Play_Config__c = office1.Id
    );
    insert player;
}

/**
 * This method tests the different outcomes for the getPlayer() method
 */
@isTest
    Run Test | Debug Test
static void getPlayerTest() {
    String userId = UserInfo.getUserId();
    Office_Play_Config__c office = [
        SELECT Id
        FROM Office_Play_Config__c
        WHERE Name = 'TestOffice1'
        LIMIT 1
    ];
    Player__c player = PlayerUtility.getPlayer(office.Id);
    System.assertEquals(userId, player.User__c);
    Player__c player2 = PlayerUtility.getPlayer('dont exists');
    System.assertEquals(null, player2);
}
```

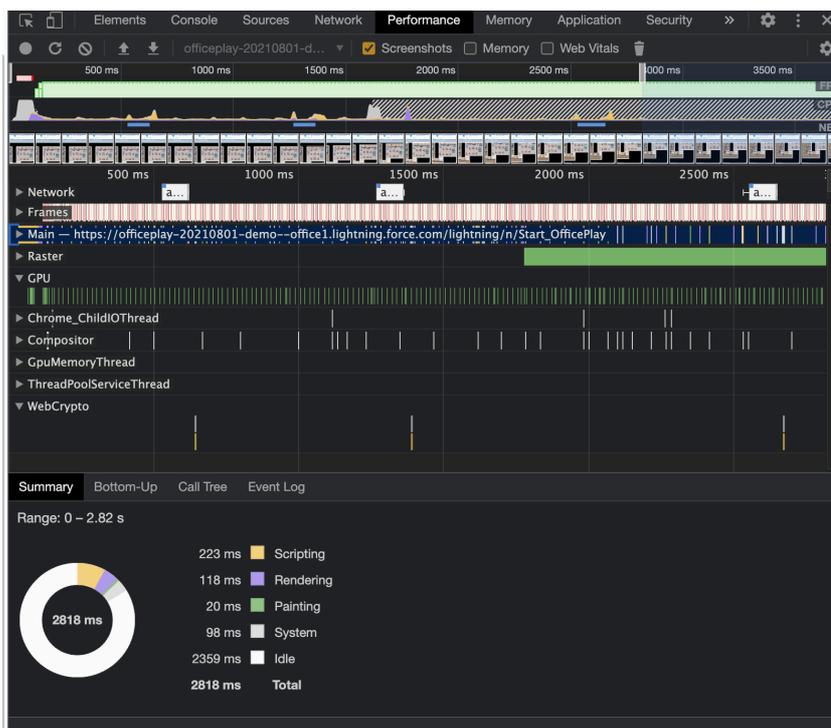
In the above illustration ‘*PlayerUtilityTest Test Class*’ an example is shown of a test class. In this class we first call a `@testSetup` method which generates a set of data to be used in multiple methods, so that we don’t need to create the same data multiple times. When you insert data into the database inside a test class, the data will be deleted after the tests have completed and you won’t be able to access or adjust actual production data unless you specify in the class to do so.

2.6. Evaluation

An important factor to evaluate and test is the performance of our application. The application will run on the Salesforce environment of customers and if the application performs badly, it can affect the other applications due to Salesforce limits being enforced. Salesforce enforces certain limits of custom deployed applications such as execution time of the Apex backend code and database time to run SOQL queries or execute DML statements. We run system tests to ensure these limits are not hit and evaluated the runtime of the processes.

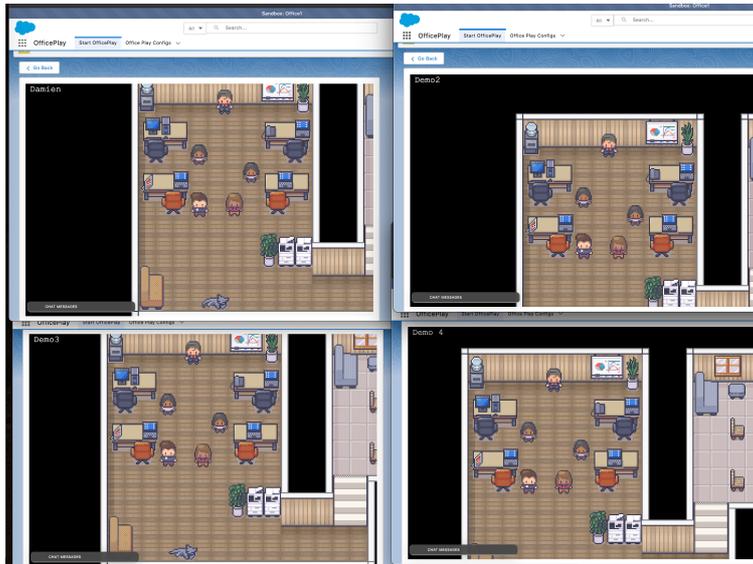
Since most of the applications logic is handled inside the browser, we have also executed performance tests inside the browser. This would not impact the Salesforce platform itself and therefore there are no limits to protect the application but it can affect the user experience and even causes the browser to crash due to CPU overload.

Browser performance test 1



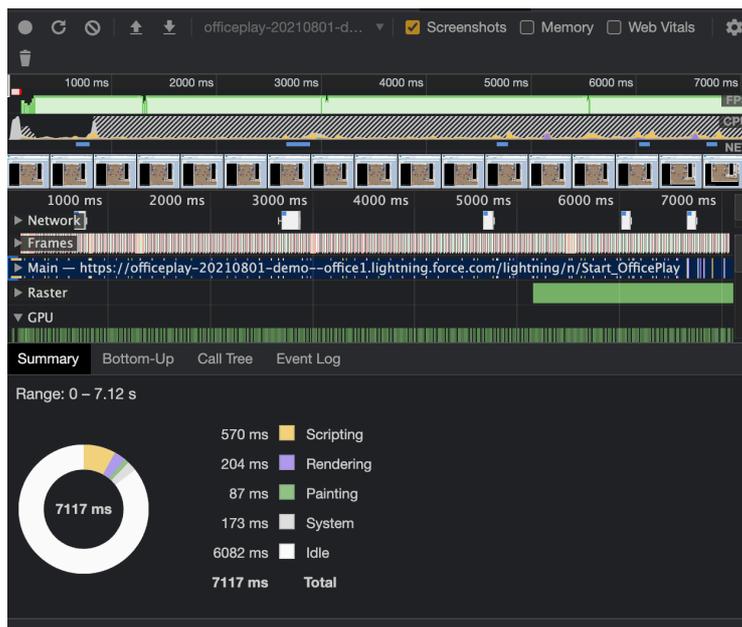
In image ‘*Browser performance test 1*’ above, you can observe the output of a browser performance test we have performed with a single user. We captured logs for 3 seconds total and measured the CPU time. We observed that most of the CPU time is spend on scripting which would include running resource intensive algorithms such as the Dijkstra algorithm to calculate all possible paths. Overall, most of the CPU time was idle so there was not a large impact on the browser performance.

Multiple users test



Since the application will be used by multiple users simultaneously, we have performed a stress test with multiple users at the same time. In the application logic the scripting and algorithms such as pathfinding are executed on the client side so those calculations are multiplied by the number of users concurrently online.

Browser performance test 2



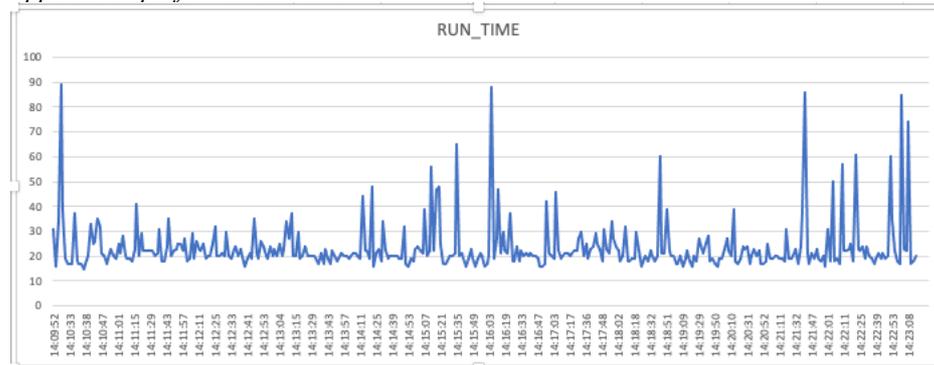
In image ‘*Browser performance test 2*’ above you can observe the output of the stress test we performed with 4 concurrent users. We ran the test slightly longer than the previous test, 7 seconds instead of 3 seconds. In the results we can observe that more time is consumed by the scripting, which explains the fact that multiple concurrent algorithms are executed to perform calculations on all the players. The overall performance impact is still very low on the browser as most of the CPU time is still idle so we are happy with the result.

Application performance logs

TIMESTAMP_DERIVED	RUN_TIME	CPU_TIME	EXEC_TIME	DB_TOTAL_TIME	ENTRY_POINT
14:09:52	31	11	1	17	OfficePlayUtility.getGames
14:10:14	16	8	1	7	OfficePlayUtility.getGames
14:10:19	34	14	2	18	PlayerUtility.getPlayer
14:10:27	89	32	12	70	apex-//PlayerUtility/ACTION\$createPlayer
14:10:28	39	20	7	21	PlayerEvent.publishPlayer
14:10:29	19	10	7	8	PlayerEvent.publishPlayer
14:10:31	17	8	6	7	PlayerEvent.publishPlayer
14:10:33	17	9	5	7	PlayerEvent.publishPlayer
14:10:33	17	9	6	7	PlayerEvent.publishPlayer
14:10:34	37	10	26	7	PlayerEvent.publishPlayer
14:10:35	18	9	7	7	PlayerEvent.publishPlayer
14:10:35	17	9	5	7	PlayerEvent.publishPlayer
14:10:36	17	8	6	7	PlayerEvent.publishPlayer
14:10:37	15	9	6	6	PlayerEvent.publishPlayer
14:10:38	17	9	6	7	PlayerEvent.publishPlayer
14:10:39	20	12	6	9	PlayerEvent.publishPlayer
14:10:39	33	12	16	12	PlayerEvent.publishPlayer
14:10:40	25	13	7	14	PlayerEvent.publishPlayer
14:10:41	26	11	14	8	PlayerEvent.publishPlayer
14:10:43	35	10	18	9	PlayerEvent.publishPlayer

To evaluate the performance impact on the salesforce environment, we monitored the application performance logs while the application was running by multiple users to ensure no limitations were hit. An important limit we need to avoid is the Apex execution runtime of 5 seconds. Once the application code executes longer than 5 seconds, the Salesforce platform can block any other applications from running by the same customer environment. In above image ‘*Application performance logs*’ we exported the CSV output of all the Apex methods that have been executed and we monitored the RUN_TIME, the values you see above are in milliseconds. The ENTRY_POINT column defines the class and method that was executed, we can see the PlayerEvent.publishPlayer methods has been executed the most. This method is called to publish streaming events to ping the player status to other players. Looking at the overall RUN_TIME of the application code in the chart below ‘*Application performance chart*’ we can see the runtime has not exceeded 100 milliseconds which is well below the hard limit of 5 seconds. We have concluded that the application performance is excellent and very low risk.

Application performance chart



3.0 Conclusions

This is a very innovative solution since we are the first to offer this on the Salesforce platform and especially during a time where most of the office workers are working from home, so this is a much-needed solution to help colleagues connect again.

The application architecture was a highly experimental design since such an application with multiplayer gaming experience features has never been done before on the Salesforce platform and I had concerns that towards the end of the development of the multiplayer features we would run into limitations and complications, but we came up with creative design approaches to overcome them.

We were able to implement the core features such as an integrated gaming experience within the Salesforce application and database platform, and the most important features have been completed such as the multiplayer and chat capabilities. This project has so much more potential, and we developed the core framework to build upon more features. Some features I would have liked to implement if we had a project team or more time would be real audio & video conversations and integrations to other collocation tools such as Slack and Microsoft Teams. I am very happy with the final solution and learned that there are not many limitations to the imagination.

“The only way to discover the limits of the possible is to go beyond them into the impossible.”
–Arthur C. Clarke

4.0 References

D'Antoni, A.V.; Zipp, G.P. (2006). "Applications of the Mind Map Learning Technique in Chiropractic Education: A Pilot Study and Literature".

Kawabata, R., Kasah, K. (2007). "Systems Analysis for Collaborative System by Use Case Diagram", *Journal of Integrated Design & Process Science*

Salesforce EMEA, 2020. *Why Salesforce Embraced Remote Working and Why You Should Too*. [Online]. Available at: <https://www.salesforce.com/eu/blog/2020/06/Ohana-at-home-why-salesforce-embraced-remote-working-and-why-you-should-too.html>

C. Gash, Slack team, 2020. *Moving beyond remote: Workplace transformation in the wake of Covid-19*. [Online]. Available at: <https://slack.com/intl/en-nl/blog/collaboration/workplace-transformation-in-the-wake-of-covid-19>

J. Hirschfeld, P. Fogart, S. Frantz, 2020. *Coronavirus: How the world of work may change forever*. [Online]. Available at: <https://www.bbc.com/worklife/article/20201023-coronavirus-how-will-the-pandemic-change-the-way-we-work>

M. Wall, The Irish Times, 2021. *New laws will give employees in the State the option to permanently work from home*. [Online]. Available at: <https://www-irishtimes.com.cdn.ampproject.org/c/s/www.irishtimes.com/news/ireland/irish-news/new-laws-will-give-employees-in-the-state-the-option-to-permanently-work-from-home-1.4458770?mode=amp>

P. Eugster, , 2003. ResearchGate. "The Many Faces of Publish/Subscribe"

5.0 Appendices

5.1. Reflective Journal & Supervisor Interactions

5.2. Project Plan

5.3. Project proposal

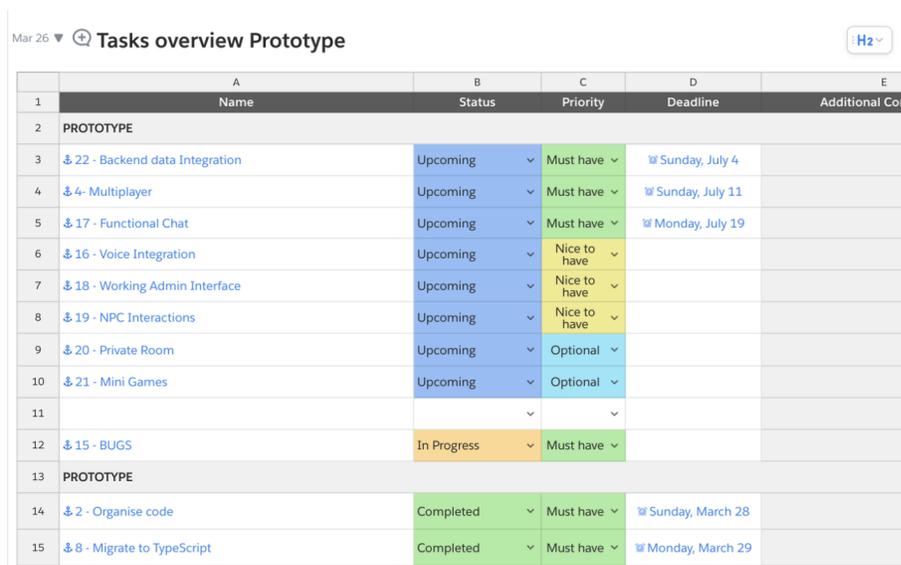
Reflective Journal

Damien Fleminks
Oct '20 - Aug '21

June & July

On the 2nd of July I had the last meeting with my supervisor Adrian Chis. She provided me with feedback on the mid-term submission. I received a grade of 77.5% which I am very happy with. She provided feedback on my submission and gave advice for the final submission.

Now it was time to focus on the implementation of the final Proof of Concept. I started with a new project plan to really focus on the most important core features of the application. The next steps would be to integrate the front-end with the Salesforce database and lay the foundation for multiplayer features.



Mar 26 Tasks overview Prototype H2

	A	B	C	D	E
1	Name	Status	Priority	Deadline	Additional Com
2	PROTOTYPE				
3	& 22 - Backend data Integration	Upcoming	Must have	Sunday, July 4	
4	& 4- Multiplayer	Upcoming	Must have	Sunday, July 11	
5	& 17 - Functional Chat	Upcoming	Must have	Monday, July 19	
6	& 16 - Voice Integration	Upcoming	Nice to have		
7	& 18 - Working Admin Interface	Upcoming	Nice to have		
8	& 19 - NPC Interactions	Upcoming	Nice to have		
9	& 20 - Private Room	Upcoming	Optional		
10	& 21 - Mini Games	Upcoming	Optional		
11					
12	& 15 - BUGS	In Progress	Must have		
13	PROTOTYPE				
14	& 2 - Organise code	Completed	Must have	Sunday, March 28	
15	& 8 - Migrate to TypeScript	Completed	Must have	Monday, March 29	

The multiplayer architecture was very challenging and it had never been done before on the Salesforce platform so I could not find any documentation and I had concerns that towards the end of the development of the multiplayer features we would run into limitations and complications, but I came up with creative design approaches to overcome them

April & May

Completely restructured all the JavaScript code into TypeScript, it took a lot more time than anticipated due to the complexity of the Pathfinding algorithms. Especially some complex algorithms such as the Pathfinder were difficult to refactor. I realized how bad you can write the logic with plain JavaScript and therefore it was more difficult to trace back the logic. Broke it down in more modular functions and added custom types to better understand the logic. This was a very important step to build a strong foundation.

At the end of May I submitted my mid-term report and prototype. The prototype visually represented the vision and features that would be included, but it was not yet fully integrated into the backend application code.



February & March

The new submission for the mid term project was in May and so I started to make new preparations to review my requirements and prototype. I had 2 meetings with Adriana, my supervisor. She reviewed my report draft and prototype progression, the feedback I received was to include more features in the prototype so the audience could really understand what the solution is about, she also recommended to create better architecture diagrams and recommended to include use case diagrams in the report.

In March I made a lot of research to rethink the architecture and solution design of the prototype, I learned how to leverage WebPack to bundle all the modules into a single js file for better performance and easier development. The solution was written in plain javascript so far and considered to refactor the whole code in TypeScript, which is an object oriented static typed language for JS. The benefits for restructuring the code would be:

- Leverage all advantages of OOP
- Much better structure in code and readability
- Enforces you to write better code
- Better to debug
- Easier to refactor code

I followed a Udemy course to learn the new TypeScript language, <https://www.udemy.com/course/understanding-typescript/>

December & January

In the beginning of December on the 2nd and 10th of December I had 2 meetings with my supervisor Adriana Chis. We discussed in detail the user research I performed for the requirement elicitation and the feedback for the ethics commission feedback. I had recorded the participants during the user interviews and it was important to have a strategy to protect their privacy. We also discussed the structure for the mid point report and expectations for the prototype.

After I had done my research and defined the requirements for my project, I started to work on my prototype to lay the foundation for the implementation. Towards the end of December my newborn son unfortunately got very ill and I had to postpone the project and a deferral was granted.

November

The feedback from my supervisor about the project pitch was that I still had to provide clearer features and address the innovative aspects. Since the initial idea was still quite abstract and just a concept I planned to organise some user interviews with the goal to find the best use case for the virtual platform idea.

4 people volunteered to do an interview. I asked about their current challenges with remote working and what use cases they would love to see in a virtual world within their daily work. After the interviews it was clear I was going ahead with a virtual office, one where you could actually walk around in with a character and meet with your colleagues all around the world.

I finalised the features and completed the project proposal which I submitted on the 13th of November.

October

In October we were tasked to come up with a good idea and submit the project Pitch. I have spent a lot of time in September and October trying to come up with a good idea. The current Covid-19 pandemic opened up a lot of changes and therefore also new opportunities. Over half of the global workforce have been working from home since then and virtual group meetings have become the standard. I decided to focus my business challenge therefore around virtual collaboration.

After some brainstorm sessions I came up with the idea of creating a virtual world to host collaborative sessions for businesses with a focus on interactive gaming experiences integrated on the Salesforce platform. I used mind mapping exercises to explore use cases.



On the 13th of October my son was born (first child), and therefore was not completely focussed on the project for the coming weeks but I had enough ideas for my project Pitch and submitted that 5 days later.

Project Planning

Tasks overview

	A	B	C	D	E
1	Name	Status	Priority	Deadline	Additional Comments
2	🔗 22 - Backend data Integration	Completed	Must have	📅 Sunday, July 4	
3	🔗 4 - Multiplayer	Completed	Must have	📅 Wednesday, July 21	
4	🔗 17 - Functional Chat	Completed	Must have	📅 Thursday, July 29	
5	🔗 16 - Voice Integration	Upcoming	Nice to have		
6	🔗 18 - Working Admin Interface	In Progress	Nice to have	📅 Thursday, July 29	
7	🔗 19 - NPC Interactions	Completed	Nice to have	📅 Saturday, July 17	
8	🔗 20 - Private Room	Upcoming	Optional		
9	🔗 21 - Mini Games	Upcoming	Optional		
10	🔗 15 - BUGS	In Progress	Must have		
11	🔗 2 - Organise code	Completed	Must have	📅 Sunday, March 28	
12	🔗 8 - Migrate to TypeScript	Completed	Must have	📅 Monday, March 29	
13	🔗 3 - Salesforce Integration	Completed	Must have	📅 Sunday, April 11	
14	🔗 6 - Start screen & User interface	Completed	Must have	📅 Sunday, May 9	
15	🔗 1 - Choose Avatar & Name	Completed	Must have	📅 Thursday, May 13	
16	🔗 5 - Simple Chat	Completed	Must have	📅 Monday, May 17	
17	🔗 14 - Develop Basic Admin interface	Completed	Must have	📅 Sunday, May 16	
18	🔗 11 - Expand the map	Completed	Nice to have	📅 Sunday, May 9	
19	🔗 12 - Screen Resize	Completed	Must have	📅 Sunday, April 25	
20	🔗 7 - Miro Integration	On Hold	Optional		
21	🔗 10 - Slack Integration	On Hold	Optional		
22	🔗 13 - Convert Phaser natively into LWC	Upcoming	Nice to have		

National College of Ireland

Project Proposal Virtual Office Experience

BSc(Hons) in Computing
Software Development

2020/2021

Damien Fleminks

X16102118

X16102118@student.ncirl.ie

Contents

1.0	Project Introduction	2
2.0	Research & Features	2
	Business challenge.....	2
	User research	3
	Features	3
3.0	Technical Details.....	5
	Implementation platforms.....	5
	Development frameworks	5
4.0	Project Plan	6
5.0	Evaluation.....	6
6.0	References.....	7
7.0	Appendixes.....	7
	Appendix A - Interview questions.....	8
	Appendix B - Interview Project brief.....	9
	Appendix C - Consent Form.....	10

1.0 Project Introduction

Since the coronavirus outbreak over half the global workforce have been working from home and virtual meetings have become the standard. Recent research conducted by [Future Forum research](#) on the future concluded that most employees have not been less productive and in some cases increased productivity. Over 80% of surveyed knowledge workers do not want to go back to the old way of working and businesses are transforming to embrace virtual collaboration. Remote working will become the new normal.

Since this opens up new opportunities to innovate in virtual collaboration, we researched new ideas by brainstorming and conducting interviews with remote workers and concluded that we will aim to create a virtual office experience much like a video game in which colleagues can casually meet and collaborate in ways that is currently not easy to do.

2.0 Research & Features

Business challenge

As someone who is personally affected by the lockdown measurements and been working from home for over 7 months now, I can easily put myself in the shoes of the end-user. Even though working remotely has its benefits, I have been missing many aspects of meeting people face to face for business and getting tired of power point presentations and video calls, and I know I am not alone.

The idea started with the concept of a virtual world much like a video game to address challenges with virtual collaboration and meetings. The ongoing digital transformation of businesses to remote working and current challenges in virtual collaboration is the underlining business challenge we want to solve.

During my competitor analysis I could not find any other such solutions and especially not on the Salesforce App Store on which I plan to develop the solution on. The closest solution is from Teooh which developed a virtual meeting tool in which you can create an avatar and sit on a table together to talk and share a screen but this is limited in that it is only made to host presentations. Other solutions are mainly providing video calling features.

User research

With a clear business challenge and rough idea for an innovative use case, user interviews had been conducted with the goal to validate, gather feedback and explore new use cases when you could create something in a virtual environment for business users. Also get different perspectives of what is important for a virtual experience to be engaging and valuable.

The interview questions were open questions to gather qualitative feedback and participants have been selected with a variation of different roles and responsibilities, such as a team manager or software developer. All participants are also working with the Salesforce platform and have been working remotely for some time now.

Many participants agreed that they have been missing more informal catch ups with colleagues **such as tapping someone on the shoulder to grab a quick coffee.** Current virtual tools used such as Google Hangouts for video chats are limited and you would not easily setup an informal meeting with someone you do not know well. Participants felt more disconnected from their team.

Few use cases have been brought up, such as hosting a large virtual conference or creating an interactive user journey for solution sales. One idea was brought up multiple times and relates to the business challenge, the idea to have a virtual office in which you can walk around with your colleagues and therefore making it more easily to grab someone's attention and have an informal **conversation. This is the use case we will further build out and perform more research to define all the requirements.**

In appendix A you can find the interview questions taken during the research.

Features

With the virtual office in mind there are some features that will most certainly be included, other features will need further investigation to validate and new features might be added. To validate the final features a prototype or sketch will be made and presented to the participants in the research group to gather further feedback.

Create your own avatar

The user will have his own virtual avatar to walk around with and identify himself with. It will be possible to customise this avatar and add a photo of their own.

Multiple rooms

The idea is that there is a main room in which you would be sitting behind a desk while working, you can walk around there and see your colleagues behind their 'virtual' desk. There would be multiple rooms such as a private meeting room, kitchen area or playroom. You could easily see where everyone is from a map and travel to the different rooms.

Chat and Audio

A chat feature will be included to chat with the whole group or in private groups. It will also be possible to use the microphone to talk, an idea is to have an open mic which is only heard when in close proximity of each other.

Presentations

There would be meeting rooms in which you could privately gather to discuss or share your screen to present, the idea is that you are able to enter a URL so you can share your browser tool of preference to present a slide deck or share your screen for example.

Notifications & requests

You would be able to send a request to someone for example to take a quick break and go for a coffee in the virtual lounge room or speak privately in a meeting room. A push notification will need to be triggered through the browser as the person might not be looking at the virtual office window.

Profile & status

Every user would have a profile with their photo and personal information and can set their status if they for example do not want to be disturbed or are away from their keyboard.

AI Characters

Computer simulated characters could be walking around the office that you can interact with. There could for example be a wellness room with a meditation expert that can give you tips for working from home and do guided meditations or a chef in the kitchen for healthy recipes.

Mini Games

Many offices have games in the lounge room like table football or ping pong and we will explore similar options within the game.

3.0 Technical Details

Implementation platforms

From a technology and innovation perspective we are exploring the possibilities to build this on top of the salesforce platform. Salesforce is a solution provider which is leading digital transformation in businesses and offers a range of cloud software which unites the whole marketing, sales, commerce, service and IT with customer 360 through one integrated CRM platform.

The salesforce platform allows developers to create business applications which integrates with the entire suite of connected apps. The game can therefore interact with all the business apps and API's, you could for example create an NPC in the game which is connected to a chatbot or integrate a virtual shop with e-commerce and track all the data interactions.

Since I work with Salesforce in my career this would also be useful to build my Salesforce experience and portfolio and a very interesting way to showcase how you can push the Salesforce platform capabilities.

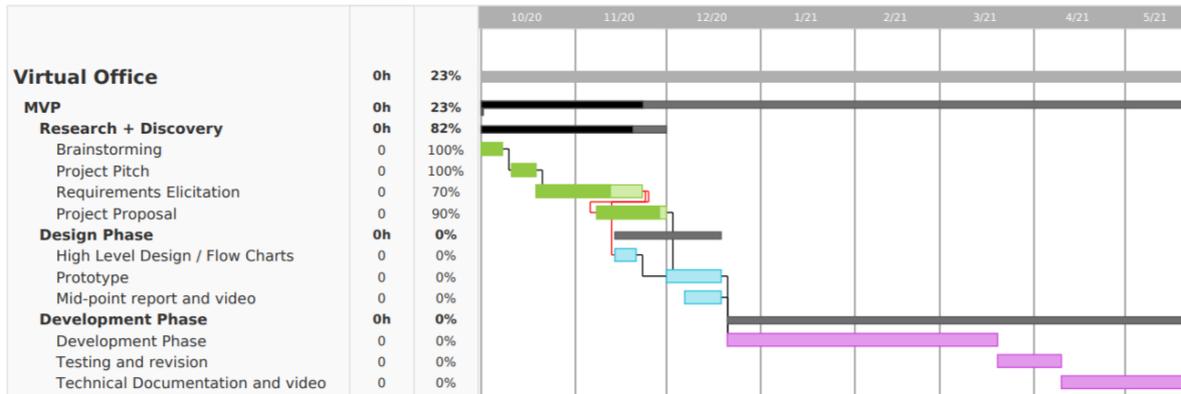
For the voice chat capabilities, a streaming provider will be selected that can provide an extensive API to build a customised solution on. Currently I am looking into Agora.io as a streaming platform as they offer a complete API.

Development frameworks

The force.com platform offers a range of programming frameworks to develop a complete solution such as JavaScript and web components for the front-end and for the back-end we can use Apex, which is a language very similar to Java.

For some of the interface we will use the lightning web component (LWC) framework in Salesforce such as the initial login and setup screens. The actual game itself will mostly be developed with the WebGL API using Phaser JS, which is a JavaScript framework to help with the game design and directly integrates with the WebGL API, no game editors will be used and we will develop both the front-end as the back-end with an object oriented design approach. Since we are building a multiplayer experience, we will also integrate this with the Apex back-end framework and the Salesforce Database.

4.0 Project Plan



5.0 Evaluation

A group of volunteers with different career backgrounds will be selected who are currently working remotely due to the new measurements and have experience will virtual collaboration tools. During the requirements elicitation interviews will be conducted to gather qualitative feedback to define the use cases and features.

Once a prototype is created a user testing session will be performed in which the users will test and give feedback on some core features before designing the final solution.

During the development process unit tests will be implemented in the back-end programming stack and regularly executed to ensure decent code execution, plus a minimum of 70% code coverage will be ensured from the unit tests.

Upon the end of the development phase, a series of integration tests will be performed to ensure the integrity of the end-to-end process and this would also involve user participation. Based on the feedback and results a final revision will be performed to flash out significant problems and fix bugs.

6.0 References

Salesforce EMEA, 2020. *Why Salesforce Embraced Remote Working and Why You Should Too*. [Online]. Available at: <https://www.salesforce.com/eu/blog/2020/06/Ohana-at-home-why-salesforce-embraced-remote-working-and-why-you-should-too.html>

C. Gash, Slack team, 2020. *Moving beyond remote: Workplace transformation in the wake of Covid-19*. [Online]. Available at: <https://slack.com/intl/en-nl/blog/collaboration/workplace-transformation-in-the-wake-of-covid-19>

J. Hirschfeld, P. Fogart, S. Frantz, 2020. *Coronavirus: How the world of work may change forever*. [Online]. Available at: <https://www.bbc.com/worklife/article/20201023-coronavirus-how-will-the-pandemic-change-the-way-we-work>

7.0 Appendixes

On the next pages the documents used during the user research are included such as the interview questions, documents shared with participants and the consent form.

Project Name : Interactive Salesforce game experience

Research Team : Damien Fleminks

Participant name:

Participant role:

- **Introduction about project**
- **Since the lockdown, how have you been doing virtual meetings & events and what are you missing from the onsite events?**
- **If you would have access to a virtual world you could use with colleagues or customers, what kind of use case do you believe would be most fun and valuable?**
 - Give some examples (Dreamforce, conference, Sale meeting)
 - Ask follow up questions, why?
 - If conference tool, what features are needed?
- **Do you think it can also be a single player experience or should it be with multiple users?**
- **Currently I am looking to develop the world in a 2D tops down pixel design, but also open to explore 3D or virtual Reality, what do you think would be best ?**
- **Since we are looking into a gaming experience, can you think of some nice gamification elements that would make the experience more fun?**
- **Since the experience will be built on top of Salesforce, we have access to all Salesforce features such as Chatbot for NPC's, trigger Flows or use leads/cases. What Salesforce features do you think would be good to leverage?**
- **.Do you think it is useful to track user data such as chat conversations or interactions in the game and what kind of data would you monitor?**
- **If you would use this platform to setup an experience, what customisation options would be important, such as creating your own room / world or dialogues with characters ?**
- **Can you think of anything else you would like to add?**

Appendix B - Interview Project brief

It has been over 8 months ago since the WHO declared a worldwide coronavirus pandemic which we are all familiar with. Over half of the global workforce have been working from home since then and virtual group meetings have become the standard.

Recent research on the future of remote working and employee surveys found that the majority of employees have not been less productive and in some cases even increased productivity. Over 80% of surveyed knowledge workers don't want to go back to the old way of working ([Future Forum research](#)) and businesses are transforming to embrace virtual collaboration.

Since this opens up new opportunities to innovate in virtual collaboration, we are researching new interactive ways to host virtual events and meetings through an online gaming platform.

A virtual world environment opens up a lot of possibilities such as organising a network event where all attendees could walk around and network with other people more intimately, which for example traditional video conferencing tools won't allow. Meanwhile there could be different kinds of speakers inside this virtual world who can share their screen and present to an audience.

It could also be leveraged for a Solution Sales meeting, for example if you are selling a vision to a groceries store chain on how a store of the future would look like, you could create a virtual store in the game in which the customer can experience this for real.

From a technology and innovation perspective we are exploring the possibilities to build this on top of the Salesforce platform. The game can therefore interact with all the business apps and API's, you could for example create an NPC in the game which is connected to a chatbot or a virtual shop integrated to the commerce solution so you can track all the data and showcase the possibilities of the salesforce platform in an interactive way to customers. It is also easy to install this solution from the Salesforce App Store and share with customers.

The goal of the interviews are to validate, gather feedback and explore new use cases when you could create something in a virtual environment for business users. Also get different perspectives of what is important for a virtual experience to be engaging and valuable.

Project Name : Interactive Salesforce game experience

Research Team : Damien Fleminks

I agree to participate in the requirements elicitation study conducted by Damien Fleminks.

Participation involves being interviewed and completing a series of tasks pertaining to the web interface by a researcher from the National College of Ireland. I allow the researcher to take written notes during sessions and allow recording (by audio/video).

I understand that the information shared and recorded is for research purposes only and that my name and image will not be used for any other purpose. I relinquish any rights to the recording and understand the recording may be copied and used by the researcher Damien Fleminks without further permission.

I understand that participation in this requirements elicitation study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

Date:

Please print your name:

Please sign : _____

Thank you!

We appreciate your participation.