****

**Fire Will Kill Us All**

Work Division Report

Created by:

Atanas Marchev

Filippo Nardocci

Nikolay Ganev

Yosif Kiradzhiev

Stef van den Tempel

**Table of Contents**

**Week 1 3**

Kick off

**Week 2 3**

Project plan

**Week 3 3**

Project plan and URS

**Iteration 1**

**Week 4 3**

Base Grid System

**Week 5 4**

Fill Algorithm

Fire Spawning

**Week 6 5**

Fire Spreading

**Week 7 5**

People Spawning

People Pathfinding

Unity POC

**Iteration 2**

**Week 1 7**

Airflow

**Week 2 8**

Dijkstra and line of sight

Fire Spread logics

**Week 3** 10

Unit tests

**Iteration 3**

**Week 1** 11

(Unity) Character movement

Start Stop Buttons

**Week 2** 12

Composite Colors

**Week 3** 12

Bugs

Unity Modeling and design

**Week 1**

**Assignee:**

**Whole team**

**Task description**

Had the kickoff and started planning the base concept for the app

**Time elapsed:**

**~ 3 hours (per member)**

**Week 2**

**Assignee:**

**Whole team**

**Task description**

Worked on the project plan and we were still discussing what exactly the app should be and how to make that happen

**Time elapsed:**

**~5 hours (per member)**

**Week 3**

**Assignee:**

**Whole team**

**Task description**

Finished up the project plan, we also did the URS in the form of git issues, and we were polishing up our ideas, going into the details of the app

**Time elapsed:**

**~6 hours (per member)**

**Iteration 1**

**Week 4**

**Drawing and basic functionality of the Form, and Base Grid System**

**Assignee:**

Yosif Kiradzhiev

**Task description**

Yosif: I started researching how to draw on forms (a bit of previous experience) and gradually building the application, and fitting it to my liking. After I had the drawing kind of figured out, I started to construct the 2d array that was going to be the base of the application (also had a bit of previous experience with 2d arrays). I went through a few different prototypes until I created the one that was mostly to my liking. Created methods to traverse through the array, search, change values in it and overall be able to use it in a meaningful way. Also created the level creation functionality.

**Time elapsed:**

**2 days(~16-18 hours)**

**Week 5**

**Fill Algorithm**

**Assignee:**

Yosif Kiradzhiev

**Task description**

Yosif: I had to create an algorithm that can be used to distinguish the outside of the floor (the tiles that were not going to be used in the current simulation), to distinguish rooms and for the fire algorithm, so I did that. It’s a pretty simple solution. You pass a Tile to the recursive method and it gives you a list with all the Tiles connected to it that are of the same color. I created a version of the method both for the 2d array and the bitmaps that we use to save and load the maps from, so it works with the created maps as well.

**Time elapsed:**

**~6 hours**

**Fire Spawning**

**Assignee:**

Atanas Marchev

**Task description**

I started by drawing out a fire sprite and adding it to the project properties. After that I used some of Yosif’s code for drawing – I had to change it because it was deleting the tile it was drawing on and then drawing on it again: I needed to keep the tile below and just draw on top. Added a button to start fires at a mouse click on the grid. After the drawing was working, I added a fire class that consists of some parameters and method signatures for the fire spreading.

**Time elapsed: 3 hours**

**Week 6**

**Fire Spreading**

**Assignee:**

Atanas Marchev, Filippo Nardocii

**Task description**

Atanas:

I began the fire spreading by making it work without errors: I created the Room class which has a list of tiles, a list of fires, a reference to the tile that has the door to the other rooms on it and some parameters for later calculations for the fire. After that I created a method that finds out how many rooms there are in the layout by using Yosif’s Fill algorithm and some extra code and adds these rooms to a list. After that class Instead of having fires separately in the form I moved them to the room class. For the spreading part I made it choose a random fire from a random room and spread out to a random room – we will change this later on in the project. I added the signature to the Flashover method which is what happens when a room explodes from its temperature.

**Time elapsed: 6 hours**

**Week 7**

**Fire Spreading**

**Assignee:**

Atanas Marchev, Filippo Nardocii

**Task description**

Atanas:

Elaborating more with Filippo we created a more realistic fire spreading algorithm. It made use of a room class for spreading consistently and a floor class for overall control and connection with the main Form for drawing. We added some events to automatically draw on the form when the fire spreads, delete fires from lists and a Flashover event to spread the fire to other rooms. Extinguishing the fires also used events and a new tile was introduced: Charred tile – it’s like a regular one but fire cannot spread on it anymore and it is used to show where there was a fire.

**Time elapsed: 4 hours**

**People Spawning and Behavior**

**Assignee:**

Nikolay Ganev

**Task description**

For the first week of the iteration, I did not do anything for the people, since their behavior relies heavily on the other classes, which did not exist yet. For week 2, I created and templated the Person and Extinguisher classes, minor adjustments to Tile class, and the PERSONALITY enumeration. I added the basic implementation of the methods, excluding the moving, which uses the pathfinding. In week 3, I did the pathfinding (see below for details) and the spawning and animation of the people. I adjusted some of the methods in the behavior. I added button to spawn people, methods which move the person from the Form class, and animating the moving person.

**Time elapsed: Total combined time is approximately 5 hours (1.5 adjusting the classes, 3.5 spawning and animating)**

**People Pathfinding**

**Assignee:**

Nikolay Ganev

**Task description**

Done during week 3 of iteration 1. First, I spent time researching and determining the best way to implement the A\* search algorithm. This involved watching videos and reading pseudocode. Then, I created the Pathfinding class and made the full implementation. Created a pathfinding object in Person class and adjusted all moving methods. For iteration 1, only personalities Shaggy and pussy are working. Other personalities have partially made moving methods, but are incomplete. Tested different scenarios and made adjustments accordingly. Currently on spawn only pussy personality is created.

**Time elapsed: Total combined time is approximately 6 hours (4.5 implementation and research, 1.5 testing and adjusting)**

**GUI Design**

**Assignee:**

Stef van den Tempel

**Task description:**

To show and interact with the application, the application needs a GUI. I made a GUI in windows form which supports the following functionality: A map where you can see the floor layout, an editor where you can save / load the map and an option to spawn fires and/or people. I started by choosing the style, than made the buttons. Placed the group boxes for the buttons to separate different parts. I made it so when you want to activate the editor, you have to press a button “create” to enable the creating part. I adjusted the positions of the different components so the GUI would look nice.

**Time elapsed: 3 hours**

**Unity POC**

**Assignee:**

Atanas Marchev, Yosif Kiradzhiev

**Task description**

Yosif: I researched different methods that could accomplish the communication between Unity and the Forms app. I found a few ways like piping and WCF services. I decided to use services due to the fact that is seemed easier that pipes and I have some experience with services. So I created a basic WCF service in out app, I only wanted it to send a string (a custom object with only a string as an attribute). Finding a way to send it to Unity took a bit more time (not a lot of people are doing it, so there is not that much info online), but in the end I found a way. I created a custom class that can be loaded into Unity with the references to the service to act as the proxy object for the client (Unity is the client). With this class attached to a game object in my Scene I could call Methods from the service in a MonoBehavior Script and everything worked fine after that

**Time elapsed:**

**~3-4 hours**

**Iteration 2**

**Week 1**

**Assignee:**

Yosif

**Task description**

Created models and animations for the Unity translation of the simulation

**Time elapsed:**

**9-10 hours**

**Air Flow**

**Assignee:**

Atanas

**Task description**

Created tiles for air currents, vents and windows. I used Nikolay’s pathfinding algorithm with a bit of tweaks to it so they can create themselves dynamically from window to vent. They we’re not drawn on the bitmap and had problems when people walked on them because they were deleting them, but I made them refresh on a tick. I also added an option of a fire spawning on a random tile in the floor.

**Time elapsed:**

**5-6 hours**

**Week 2**

**Dijkstra and Line of Sight**

**Assignee:**

Nikolay Ganev

**Task description:**

I implemented the functionality for the Hero and the Selfish personalities, and them getting an extinguisher. I implemented a basic version of Dijkstra algorithm, which I improved as well, used to locate the nearest tile on fire whenever someone gets an extinguisher. Implemented extinguishing the fires using said extinguisher. Slightly improved the hero and the pussy personalities. Added a line of sight to the people, as well as implemented realistic running to the exit/extinguisher, and checking for other possibilities when the path is blocked. The new movement system works with the line of sight.

**Time elapsed: 12 hours**

**Fire Spread logics**

**Assignee:**

Atanas

**Task description:**

Because fire doesn’t spread randomly, I had to think of ways to make it look and feel more real and have some sense behind it. The fire had to spread to the place with most air in the room, it could also spread through doors and the air flows of the room would definitely help it spread faster and easier. I divided the spread method in three and had the fire spread from a queue of tiles that this method filled out. The first part checked for doors and if they were open, if so they would get added to the queue. The second part was the main fire part where the fire had a list of tiles for each direction and it spread to the one with the most tiles. The third part was the off-fire spread – a random fire from the entire list could spread out from the main path and if there were more than 5 or 10 fires more fires would actually spread out. And the last part was spreading from air currents – the fire checks if there is an air current on its tile and if there is it adds the neighboring air current tile to the queue.

**Time elapsed:**

**7-8 hours**

**Assignee:**

Yosif

**Task description**

Fixed and improved the service responsible for transferring the info from the Form to Unity

**Time elapsed:**

**10-12 hours**

**Week 3**

**Person, Pathfinding, Dijkstra and Extinguisher Unit Tests**

**Assignee:**

Nikolay Ganev

**Task description:**

Created unit tests for each of the classes I made (Person, Pathfinding, Dijkstra and Extinguisher). It took me so much time to make, mostly due to the Person class. Since the whole person spawning is an automated process, done by the form, in each test method I had to simulate a small grid, which took the most time. Also, I made small readjustments to the classes, in order for them to pass the tests.

Note: RunForTheExtinguisherTest method sometimes might be failed. This is due to randomization in the Person’s constructor. When ran again, the test usually succeeds.

**Time elapsed: 4.5 hours**

**Random People Spawning and Removing Extinguisher from Wall**

**Assignee:**

Nikolay Ganev

**Task description:**

I made some small improvement so now the hero runs for the exit when there are no available extinguishers for him. Reimplemented Dijkstra from scratch, as the previous implementation was different and had some problems. Hero now checks whether there is an extinguisher on the wall for him to take, preventing infinite extinguishers in one place. Added the removal of an extinguisher from the wall when a person takes it. Finally, I added the option to randomly spawn a select number of people, and created a sequence diagram for the GetExtinguisher method.

**Time elapsed: 4.5 hours**

**Assignee:**

Yosif

**Task description**

Created the methods and functionality responsible for sending the different commands from the Form to Unity

**Time elapsed:**

**5 hours**

**Room Unit tests and cleaning bugs**

**Assignee:**

Atanas

**Task description:**

Most of the tests were successful but some had issues – I had to clean up my code and add some checks here and there. The actual testing was a nuisance because I had to simulate a grid and use it without seeing what’s happening on it. I did the tests, cleaned up some bugs, documentation on my work – sequence diagram and the plan for iteration 3.

**Time elapsed:**

**5-6 hours**

**Iteration 3**

**Week 1**

**(Unity) Character movement**

**Assignee:**

Yosif Kiradzhiev

**Task description:**

First created a general movement, jumping from point to point like in the Forms app, after that convert the movements to be smooth. That was achieved with linear interpolation. The characters update their path every few seconds (when the service pulls) so sometimes they jump to a location (not smoothly) kind of like the methods used in games.

**Time elapsed: 12~14 hours**

**Start Stop Buttons**

**Assignee:**

Atanas Marchev

**Task Description:**

The application needed methods that would start the timers of the app running and pause them when needed. I added buttons so it can be a functionality in the application. The idea of them was to have them start and stop the multiple running of simulations.

**Time elapsed:**

**~3 hours**

**Week 2**

**Composite Colors**

**Assignee:**

Atanas Marchev

**Task Description:**

Drawing on top of other colors in the bitmap overloaded them so I had the task to create a way to store more than one tile in a color. The idea of the teacher was to have a bitshifting operation that would check the bits and give the color a value for one of its main colors – ARGB. I couldn’t understand it very good and just created two methods that encode and decode colors in certain situations – if the colors A value was 254 that meant it was a composite color and every other color value meant a tile by itself. It was a bunch of checks and I re-did the drawTile methods in order to have it work. I had problems with the persons drawing on top of doors and exits.

**Time elapsed:**

**~7 hours**

**People Behaviour**

**Assignee:**

Nikolay Ganev

**Task Description:**

Added the option to choose which personality to spawn, or to spawn one random. Improved the random number of people spawning functionality. Fixed people colliding with each other. Worked on solving a problem where 1 hero does nothing and waits for the other to extinguish a fire.

**Time elapsed:**

**~3 hours**

**Week 3**

**(Unity) Bug fixing**

**Assignee:**

Yosif Kiradzhiev

**Task description:**

Fix as much bugs occurring in the unity implementation and the c# Forms app

**Time elapsed: 5 hours**

**(Unity) Modeling and design**

**Assignee:**

Yosif Kiradzhiev

**Task description:**

Fix some errors with the animations of the character and creation of a particle effect to represent the fire instead of the previously used block.

**Time elapsed: 4~5 hours**

**Bugs**

**Assignee:**

Atanas Marchev

**Task Description:**

The previous weeks changes had a lot of problems with the people drawing on top of other stuff when moving. First of all they couldn’t store the colors properly and some values were lost. Then they started saving everything and they did not delete themselves when moving away from a tile. After that they couldn’t draw doors properly because of the CloseDoor method because some values were missing in the decode method I added them. Last thing I changed was that the exit door was not checked in the decode method and people just drew white when moving away. Almost everything was fixed – I did not have enough time to bug hunt absolutely everything so some stuff might occur.

**Time elapsed:**

**~6 hours**

**People Behaviour (Continued)**

**Assignee:**

Nikolay Ganev

**Task description:**

Further continue working on the 2 heroes problem. Managed to figure out a solution, which led to slightly modifying the Dijkstra algorithm and the person’s move method. It works 95% of the time now, meaning that all the scenarios I tried it worked, but I couldn’t think of others and I might have missed some. Additionally, fixed some minor bugs which were left over.

**Time elapsed: ~4 hours**