Daily log

# Monday 11/02

Today I’ve been mostly busy with setting up the basics that every unreal project need. Setting different settings which are needed for my project, and which aren’t.

I’ve also created the base game mode, game state, Hud, pawn, and player controller which unreal needs for any game to run, and given it some basic camera movement.

The 2nd half of the day I’ve been setting up basics for a single tile, making a base c++ and blueprint file for it, where I added a PaperSpriteComponent to so it shows a sprite in game. And I made a first step into an actual tile map, which spawns a grid of the tiles (all of them the same for now). Note: directly saw that 100x100 of the actors with just a PaperSpriteComponent does not work fps wise

# Tuesday 12/02

After finishing a basic tile map on the first day, I already found that just having a 100 x 100 grid of actors with PaperSpriteComponent in the game does not work for FPS (I got 30 fps at this point, with still an mostly empty game), so I needed to chop them up in chunks, and turn the rendering off the offscreen ones. I did this by making a “TileMapChunkActor” to keep the tiles under, and easily call 1 function to turn off a larger amount of tile renderers.

The reason this is an actual actor is that in unreal, anything that you want to spawn and have in the world outliner has to be an actor, but you can turn almost everything off to make it basically a holder object. So, for what I could find this should not impact fps.

With those chunks in place I moved the spawning of tiles into the chunk, so it is responsible for spawning its own tiles for readable code.

After this was done, I had enough time to add culling to the chunks, where if the camera moves, there will be a check to see if there are any chunks that need to be enabled/disabled to keep draws to a minimum. (ref: 43281)

This did make my FPS go up a bit again (~60, while I have the settings capped at 120) but was still far from enough, and I found that this was because there was no batching capabilities with the PaperSpriteComponent, so I’ll have to find another way to make batching happen to reduce the draw calls to a small amount instead of 1 for every tile (making it still ~ 300 draw calls with 5x5 chunks, and only 1 layer).

# Wednesday 13/02

First part of the day I had to find out what would be a good option to draw a Tile map in unreal, where in the end I came to using the Paper2DTilemap. This object has only drawing functionality, and not a way to add different functionality to different tiles, which was the initial reason for not using it, and making my own tile map.

Not having used this before besides looking into what it could do with custom tiles, I had to take my time in finding how this exactly works, and took most of the day to make the tile actors I made draw onto the tile map.

# Thursday 14/02

Today I started with adding different layers to the tilemap, which now contain 3 different layers where objects can be set onto.

I also written code to get all 3 objects from the different layers at a specific location.

After talk with Jeremiah I went onto write more into the 4th year plan, and make the goals more specific to be able to plan them in better, and plan the thing I need to do after the basics

# Friday 15/02

Today I have been writing more detail into my plan for the whole day.

With this I went almost from the start to the finish again, adding tasks to hack’n’plan for what I think I need to do for the different steps, and how much time I think I will need to make them.

# Monday 25/02

Meeting:

Citethisforme.com

- Correct way for referencing sources that I use

Plan looks better now

Likes use of hacknplan

Added a lot of functions to make editing the map doable without having to start playmode, making it that I don’t have to make a way to safe while playing yet.

These include:

Selecting a whole layer

Selecting a chunk from all layers

Selecting a chunk in a specific layer

# Tuesday 26/02

Made functions to select a room of the map:

Selecting the middle bit of a chunk

Selecting the walls of a chunk

Building a room with wall and floor

Added texture variants for the grass and concrete:

Give a tile multiple possible textures and it switches between them when the tile is created

# Wednesday 27/02

Made a first map which is a full grass ground and a small room in the bottom left corner.

Started on tiles being bigger than 1x1 and sprite-based tile

Sprite tile has a separate sprite attached instead of drawing its sprite on the tile map, this is for things that aren’t full-tile sized or move during gameplay, example of this is a door that moves to the side when opening.

For tiles that aren’t 1x1 I needed to add checks to see if it could be build on the other tiles than you click on, and set the tile map to draw the tile on all off the places.

# Thursday 28/02

More work on making the non-1x1 tiles work correctly.

Work on checking other tiles for them not being built on already.

# Friday 01/03

Started working on GUI, made a base and a popup system.

Started with basic info about what the player is hovering over with his pointer

# Monday 04/03

Added a building select pop-up menu based on a “file” where you can go through different tabs each with different object types.

I also added a info item on the left side that shows which item is selected, this disappears when no object is selected

# Tuesday 05/03

Wrote the first 2 weeks into the big learning journal

Started on mechanism to build onto the map

# Wednesday 06/03

Finished the ability to build objects onto the map, every item is able to be selected in the build menu, and placed into the world if the location Is free

Started on creating a default room in 2 clicks

# Thursday 07/03

Finished building a room in 2 clicks

Made a preview object, that shows an preview at the location where you want to build, and turns red if that location is obstructed (color single object only)

# Friday 08/03

Started on pathfinding

Remembered most things of how pathfinding works with a 30 min video instead of a day’s worth of research that I planned.

Made a thing that calculates a path from one point to the other. And can show squares on the path

# Monday 11/03

Feedback with jeremiah:

Happy with progress

Next week: try having ai’s walking around, offload the pathfinding to different thread

Started today on doing pathfinding async, getting delegates to work is very difficult, I can’t get it to work to have a function with the same parameters as the unreal delegate bind functions

# Tuesday 12/03

Been working on making the pathfinding async, looks like it’s almost working, started on a component to add to any pawn for it to follow a path

# Wednesday 13/03

Fixing async pathfinding, when there’s a lot of characters, there happens null ref errors when the game is stopped before paths are calculated, additionaly randomly null ref errors when getting a new path during normal gameplay

# Thursday 14/03

Had to re-write the threading to a different type, as the “FAsyncTask” that I used before did only use 1 thread for all the tasks you make, so I had to fully make a threading object.

I used <https://wiki.unrealengine.com/Multi-Threading:_How_to_Create_Threads_in_UE4> as tutorial how to do this.

This took me some time to figure out, but in the end of the day I had it.  
biggest difficulty was to get reading of which places of the world where walkable in a threadsafe way. And to get closing of the thread safe, which I did not figure out yet.

# Friday 15/03

First half of the day I worked on making reading the “get walkable” threadsafe, which I now got for 99% and it still only crashes when you close the game at a wrong moment. Which I have to spend some more time with, or ask jeremiah on Monday.

2nd half of the day I’ve been finishing my journal entry for the first 4 weeks. Which I always dislike but goes oke with help of this log.

# Monday 18/03

Meeting ask about:

High Level Development Plan

Or

Documenting Development Process & Intentions

Meeting:

Make sure the delegates are really threadsafe

Race conditions

Mutex on shared data

Book: c++ concurrency in action

Outline your progress, with regard to competency development?

For each competency, try to relate what I’ve done to learning about the competency

Next week:

Pathfinding working in separate thread without crashes

Started researching different ways to safely return the path from thread to main thread

Might need to use a TQueue which is supposed to be even more thread safe than delegates

# Tuesday 19/03

Did more work on moving from delegate to TQueue

End of the day it looks like I figured it out, left it running for the last half hour of the day and it looks to be working without problems now!

# Wednesday 20/03

Multithreading is fully working now. Fixed the “crash on close” crashed that happened because of not correctly closing the threads.

Reason for this was that I did not correctly call the shutdown functions on the different threads.

# Thursday 21/03

With multithreading working I now wanted to add some kind of visuals to the characters, to that I can make multiple different ones which also are visual different.

Made a base pawn for this.

# Friday 22/03

Went further with the base pawn and added a controller, lot of default stuff that has to be done for organizing the different AI’s later on.

Nothing difficult, mostly time consuming, but it has to be done to make the project work.

# Monday 25/03

The day went okeish for how I felt, been half sick at home w/ bowel problems making me having to take a lot of toilet breaks which made it difficult to concentrate.

Even with these problems I made some progress in making a singleton ai manager actor, which will give errors if you put more than 1 in the level.

I also started on the base tasks, which are data assets which will be overridden for the different task specific things

# Tuesday 26/03

Very very tired, did not work to well.

Managed a first draft of task queue and base tasks.

# Wednesday 27/03

Had a lot better day than yesterday, managed a almost working queue, it only has some problems with overriding itself.

# Thursday 28/03

Got the task queue working fully, without it crashing. Had to write my own linked list mechanism for this but it works very nice

Starting on the implementation of building tasks, where an builder goes to the location and wait’s for a few sec then changes the tile.

# Friday 29/03

Finished the building task, made the destroy task, both work without problem now.

Had a small problem with the tasks all having a shared timer, making it only take 1 time the build time, and all tasks after that will be instantly build.

Written the start of an ai interactable tile registry, which also works fully, only crashes if there is nothing in there yet and something is added by c++ code instead of editor menu’s

# Monday 1/04

Very bad day.

Started of almost sleeping in because of the summer time change. And things didn’t go better from there.  
I did a bit of work on the registry system, changing it from classes to strings of the name of the class, which isn’t working correctly yet.

From 11 am everything went even further downhill though, as my laptop over-corrected for summer time, and said to me it was 12 already. So, after an hour break, I had to go to work again instead of having a meeting with Jeremiah directly after lunch.

When trying to start back from this annoyance, I checked my mail only to see that jeremiah was not in because of illness.

The combination of this sleepy-ness, bad luck, and mishaps made me not believe in the day anymore, and leave for home. I’ll be making up for the hours lost on Saturday.

# Tuesday 2/04

Very good day, I’ve been working hard on getting a location register working, and it now has a list of tiles where ai can spawn from.

Also changed up the input a bit, to have clearer/easier changeable modes. this is so I could add a “prioritize” mode which when enabled moves the task you’re clicking to the priority queue, which is also all implemented.

# Wednesday 3/04

Today was mostly reading about how to implement a GOAP system, and watching some videos.

Added the really useful links of it to the hack’n’plan design model element: [#7 GOAP](https://app.hacknplan.com/p/84596/gamemodel?nodeId=7)

# Thursday 4/04

Started the day off with more research into GOAP, added a last few links to the design model.

After lunch I started on writing the state nodes, which will be the key/value pairs for testing state.

# Friday 5/04

This day was good, did a lot of work on the state nodes that the GOAP planner will use, had some problems with “object” state, which holds a reference to an object. Example of this usage would be an inventory slot having a ref to building tools.

For not having to write duplicate code for every type of object that can be in such a state, I have made one with a ref to unreals UObject which is the base class for any class that is in unreal.

The biggest problem I had with this is being able to find which class it is, as when you set it from the content browser it would always say it’s just a generic “blueprint” or “class” and doesn’t give the correct c++/blueprint class.

# Monday 08/04

Meeting notes:

Building stuff over ai path’s does not recalculate it -> fix

Make sure sources etc. are in learning log

Use word reference tool for references

In the morning I mostly finished the state nodes, with writing the last things for object, and full code for float and string nodes.

After meeting I started on fixing the AI pathfinding where they don’t care if a wall is being build on top of their path.

# Tuesday 09/04

Finished the ai pathfinding not caring about an object being built on top of the path, and they will now calculate a new path if they find out something is built on their path.

Started working on the GOAP actions, comparing pre-conditions and adding the effects to a existing state.

# Wednesday 10/04

In the morning finished most of the base action structs. In the afternoon I started on the way that these actions are actually done.

# Thursday 11/04

Started writing the planner that plans the actions to be taken  
think I have it mostly done, need to go and write tests for it starting tomorrow.

# Friday 12/04

Wrote a lot of debug stuff for GOAP plan creation. It almost all works now, but I made an oversight with a thing in creating new states that isn’t threadsafe.

# Monday 15/04

Spent the morning writing the learning journal from the week, and completing the “4-week period."

After having done all that I started with adding all the sources that I’ve used to the document, downloading Zotero and going through all my internet history to add the sources I’ve used.

# Tuesday 16/04

Today I spent the day writing debug code and do debugging on the planner.

Started the day of with making the planner work on threads, and after I had to add stuff to the state checks that test if a action can be done and is usefull.

At the end of the day I got the planner fully working, meaning I can create plans from any starting state till the wanted end state.

# Wednesday 17/04

Started on doing the flow of the actions.

The GOAPWorkerComponent will be doing every action the plan returns in order, and finishing with the task.

This last part isn’t working yet, and the actions don’t walk to a place where it has to happen yet

# Thursday 18/04

Finished the GOAPWorkerComponent flow through the plan, it all works now including it doing the end task and going to the location of this end task.

Fixed everything which has broken during some refactoring of old code (making set functions for transforming from tile to world coordinates).

# Tuesday 23/04

Finished the GOAP worker, where they now also go to locations where they can get their tools/materials.

This finishes the goals for the GOAP milestone and will mean I’ll start on the next part next.

Next part: management.

I’ve started on a day/night cycle which will make the world lighter/darker

Next up a way for staff to leave/enter at specified times of day

# Wednesday 24/04

Finished the day/night cycle thing.

Made buttons and things for speed management in the ui

# Thursday 25/04

Finished the Speed management ui and usage

Added a stat’s manager which keeps track of the player’s money, ai count, and some other stuff

# Monday 06/05

Meeting Jeremiah:

Had a great meeting, he was very happy with the progress I made.

Agrees that the foundation is here now.

Liked to learn about Zotero as a source tracking software.

In the morning I’ve worked on the menu where you manage staff

Created a template with sliders to change the time ai’s leave/enter

Sending this data to the ai still has to be made + a mechanism that the ai’s leave.

In the afternoon I’ve gone home ill.

# Tuesday 07/05

Today I’ve been trying to fix a bug found during the meeting with Jeremiah, where the ai’s stop to do nothing once they get any action that can’t be done.

They now stop their task if they can’t reach their goal, and try to get a new one.

# Wednesday 08/05

Today I’ve been trying to get the data from the staff manage UI send trough to the AI’s, and other way around on creation of the UI elements.

The first part was quite easy to figure out, but getting it from the AI to the UI was more difficult, as it had to scale back from the range 0-24 to 0-1. Which I couldn’t figure out.

This ended up being a problem in casting, where it instantly casted everything to integers, ending up with the slider always being on either 0 or 1, and nowhere in the middle.

After having fixed that I implemented the ai’s leaving and entering the world outside of the times they need to, this was quite easy, as most of the logic was done, and I only needed to make the algorithm for when to leave/enter.

# Thurday 09/05

Today Today I’ve started the morning with adding indicators for if the builders have different items in their inventory, for which I added a callback when a AI’s state changes.

In the afternoon I’ve added more objects to the game, 4 floors, wooden walls, and a fence.

Last part I’ve been trying to package the game, which ended up that I have something wrong with my unreal install, and I’ve been trying to fix that.

# Friday 10/05

Today I’ve used all my time on fixing the packaging, there where quite some errors in there about functions that are allowed in engine, but not in a packaged build.

It was quite clear when this was the case, but it only showed 1 of these issues each build, meaning it costed me a lot of time.

In end of the day I got this all working, and made a first build which does work without the engine. <https://www.dropbox.com/s/4pp7wu83m1xwfmz/CasinoInc0_0_1.zip?dl=0>

A problem with this is that it crashes after some time when you actually build stuff, which I have to try and fix next.

# Monday 13/05

Ask about packaged builds crashlogs

CI\_GOAPWorkerComponent\_CPP.cpp[228]

Add close button to staff menu

In the morning I searched for some art for the casino games, and added tiles for them.

In the afternoon I’ve been fixing the crash that happens in the packaged builds. Seems to be something with memory being cleaned.

## Tuesday 14/05

Have spent the whole day trying to fix the crash, besides the knowledge that It is breaking in unreals garbage collection I have not found out why/what is being garbage collected differently between engine and packaged build.

# Wednesday 15/05

After a lot of work, I think I’ve found where it’s being garbage collected, which is the task list, where I duplicate udataassets which unreal doesn’t like to much. I’ve started to rewrite this so it doesn’t need to duplicate these objects anymore but just reads from them.

# Thursday 16/05

I’ve spent the morning finishing the fix for the crash in the packaged build. When I re-wrote everything to get rid of this duplicate it did not crash anymore.

# Friday 17/05

Today I started writing the guest stats, and how interactable objects (slotmachine, poker table, blackjack) change this.

Made a struct for this, with the stats: happiness, chips, money, hunger, thirst. Of these happiness, chips, and money, will be implemented first.

I added logic to “interactable object” that change these.

# Monday 20/05

Limit amount of builders etc.

Making of full casino, floors, walls, machines, etc.

Had a good feedback session with jeremiah, talked about problems I’ve had and solved last week, and documentation that needs to be done.

# Tuesday 21/05

Made the stats for guests and personalities

There stats are done in a struct attatched to the guests controllers.

The personalities are default values and multipliers in a database/excel sheet where I can simply add more types of personalities with the different values.

# Wednesday 22/05

Made implementation for the personalities to be used when a guest uses a interactable.

Made the beginning decision making for guests what they want to do