Treasure Island GDD

Mete Ruso - M00774805

Gam2002 Advanced Creative Technology

Module Leader – Penda Tomlinson

High Level Design

This is the Game Design Document for a top-down, grid based, mobile game. The aim of the game is to head to different dig sites and unearth treasure by digging it up. These treasures are sold, and this currency can be used to upgrade a player's base and upgrade their tools to make the process more efficient.

Design Pillars

Design Pillars are "Rules that support your goal and vision of the ideal game you are creating" **(Ross Metcalf, 2018).** Creating a set of pillars early informed my decisions for the project such that they were inline with the pillars I had set up.

The main pillars that will guide my design process in making this game are:

- Making a relaxing/non-stressful game
- Making a game that requires the player to engage with the systems and think for themselves
- Making a game that attempts to maximise the play time of the end user

For example, there was an idea for energy to also deplete slowly on a timer such that there was pressure on the player to make a decision on a timer. This was scrapped to be in line with a relaxing game and also to encourage forethought of actions.

Gameplay Loop

The player will select the desk in their home, and this will take them to the map which displays the different levels. They will select one and be placed into a top-down grid. They can move around by tapping and dig to find treasure. This requires energy which is replenished over time. Once they are done digging in the map, they will be given the option to sell their treasure or keep it to display. This will impart bonuses for the player. They will select another map to redo the process.

Loops can be broken down into four stages: Action, rules, feedback, model **(Daniel Cook, 2012).**

Action is to dig

The rule is to search for items and take away energy

The feedback is to return items and information

The model is the new state of the game which is used to inform the next action/dig area

The hint system can be described as a gameplay arc since in only occurs once per level with a single pay-out per hint.

Moment to Moment Gameplay

When put into the map, the player will be able to move by tapping on their desired destination. There will be a starting maximum energy of 100 and moving 3 squares will deplete energy by one. There will be a prompt on the screen with a shovel. Tapping this dig in the area. Digging will deplete energy by 4. When the grid square is dug up, it will display any found treasure along with a number which will tell the player how many pieces of treasure there are in the 8 adjacent tiles. They will use this information to dig around the initial dig site to find more treasure.

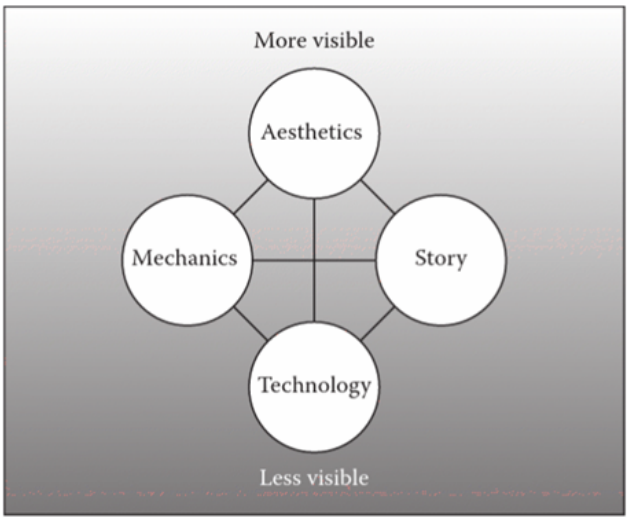
This was the original idea for energy consumption. Instead, energy was set to 500 and all actions (moving and digging) cost 3 energy each with the potential later for modifiers and bonuses to conserve more.

Theory

Games like solitaire demand nothing from the player in terms of reaction time and energy. It is the perfect break. Casual games are used by many players to zone out **(Ian Bogost, 2011)**. I tried to emulate a “lean back” experience by removing any timer or need to make a quick decision and offer an abstract puzzle.

Using arbitrary relationships between tiles and numbers and shapes/tiles the game is removed from any real-world connection. This makes the game stand on its own as a relaxing experience

I think that any game can be examined using the Elemental Tetrad. I wanted to use the elemental tetrad to solidify my themes which can be done by looking at each element and how that element can be used to serve the themes of relaxation and detachment.



The aesthetics are fairly simple. An isolated island. Golden beaches and azure waves lapping on the sand conjure images of holiday and relaxation. It is a locale that some may even greet with wonder with tropical jungles and ruins. Though it is only tile based and drawn by me, the colour palette serves the themes well.

The mechanics are simple. As stated before, they are abstract and arbitrary, so they remove themselves from the world that the player is used to. This is also to reinforce the theme of escape.

The story element is made by the emergent gameplay choices the player makes. Due to the random nature of treasure generation and the lack of any narrative, the story becomes the player’s personal journey through the game.

The technology element is served by simple user interface and buttons that have one simple function each. It removes any need for the player to learn how to play the game which encourages a relaxing environment.

Low Level Design

Digging & Treasure

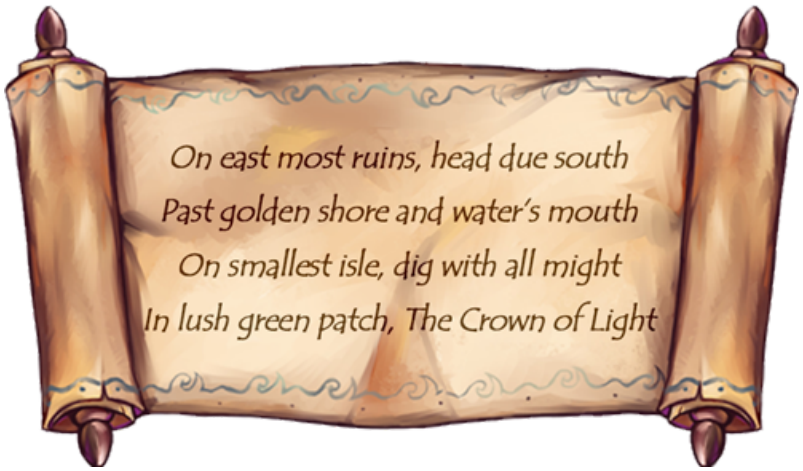
The level is created via inputting a text document into a script which makes it easier to create many levels. The text document is read, and it creates the tiles one by one. When the tiles are generated, they have a 1 in 8 chance of being a treasure tile.

When a tile is dug up, it reads if there is treasure in that tile. If there isn't, it creates a crater in the tile so it cannot be dug up again. If there is treasure in the tile, it will create a slightly modified tile to show treasure was dug up from the tile and give the player the item.

Regardless of if there was treasure in the tile, it will search the 8 adjacent tiles to see if there is treasure in any of them and display the number of treasure tiles. This will not change if any of these treasure tiles are dug up, so the player is expected to count them all. This is to create a sense of deliberate and considered gameplay such that the player thinks about their next dig to conserve energy and be as efficient as possible.

Hints

The player will be able to dig up secret messages in a bottle which give hints to larger and more unique treasures. These will use the unique tiles on the map to point the player in the direction of unique and valuable treasure. This will be through riddles and worded directions. These unique treasures will be pre-set in the maps but in harder to reach locations, such that it is less likely but not impossible to be found without help.



This was the original idea for the hints, but I decided to make them available as soon as the player loads into the level since a random chance to find a riddle seemed unfair to the player and unnecessarily gated off a fun challenge.

Home Base

The home base is where the player will be able to spend their hard-earned money. Without any upgrades, the home will have a desk and a cabinet. The desk, when tapped, will display the map to get into gameplay. The cabinet is where the player can store up to three pieces of treasure that will confer bonuses to help them dig more treasure. Like higher maximum energy, lower energy cost for digging or increased money earned.

The player will be able to buy furniture for their home which would increase the energy regeneration rate which allows the player to play more often. This may include a couch, bed, TV etc.

This was not implemented in the final build due to time restrictions, however, this was the intended usage for the player's money. Intended to give the player a sense of progression and also allow them to become more efficient in their digging as they not only got better at finding treasure but also spend less resources.

Advanced Technology Research

Data Driven Design

Data driven design is the practice of using to externally created data to influence interactions and code in the game. Data Driven Design is effective to be easily able to change things about the game without fiddling with the game engine or inspector. It is best to have planned out your game before coding it since the scope is already clear.

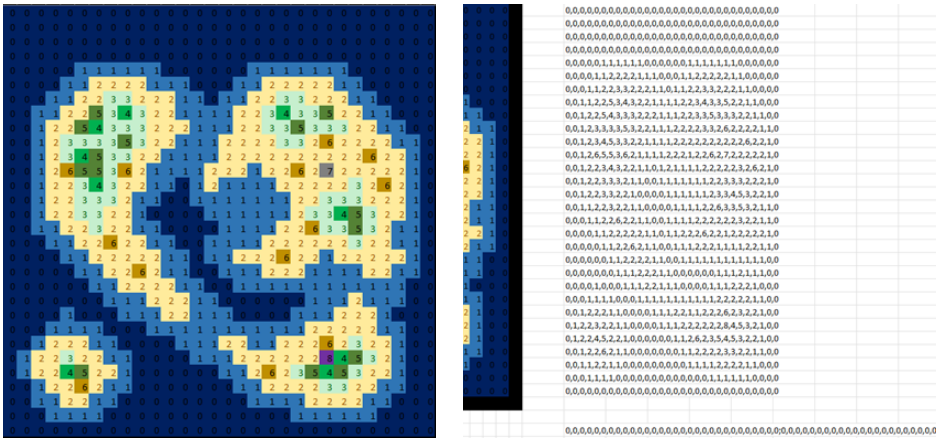
“Projects frequently fail due to flaws in the development process” **(Scott Shumaker, 2001).**

To avoid these problems, a system where all of the data is in accessible external forms and important variables are already written out can be a powerful tool to accelerate the development process and minimise game specific code.

“Once a game has shipped, its code tends to be reused for subsequent titles.” **(Kyle Wilson, 2002)**.

This game will use principles of data driven design which will use strings of characters to form the grid map. As well as an economy of in game currency that will need to be balanced using spreadsheets. Each character in the comma delimited string will represent a different tile with different properties. These strings will be loaded into an 2d array to create a grid map.

Using a excel sheet made by Penda Tomlinson, I inputted characters into a grid that represented different grid tiles. I added conditional formatting to add colour to help visualise the level before the level was made.



Each line’s characters were recorded and combined to give a single string with each tile delimited by a comma and each line delimited by a semi colon. This string was put into a txt file to be taken as a reference for the map maker in that scene.

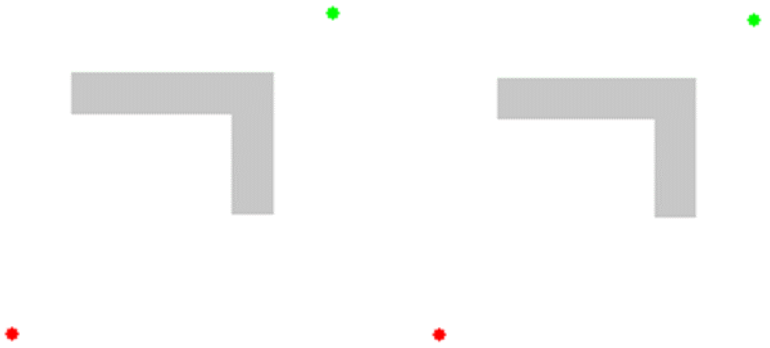
The other instance of data driven design is the loot items. I created a table in excel before implementing these items to work out their values and chances to spawn so it would be easier to make an if statement encapsulating these values. It may also have been an opportunity to create a system where the game actually reads from a table to get these values instead of just using the outside made data as a reference to create an if statement.

Treasure	Value	Discovery%
Bronze		70
Coins	150	15
Circlet	300	15
Goblet	400	13
Coins Full	450	10
Circlet Gem	650	9
Goblet Gem	850	8
Silver		25
Coins	600	7
Tiara	850	5
Goblet	800	5
Coins Full	800	4
Tiara Gem	1350	2
Goblet Gem	1200	2
Gold		5
Coins	1500	1
Crown	2500	1
Goblet	2000	1
Coins Full	1900	1
Crown Gem	3500	0.5
Goblet Gem	2500	0.5
Special		Set Placement
Crown Of Light	7500	
Cursed Goblet	7000	
Runic Gem Pouch	8250	

Pathfinding

I attempted to implement a pathfinding algorithm where the player could click anywhere on the map and the algorithm would find the shortest path to walk there. Since moving consumed a resource, it would be unfair to do anything but move the player at the utmost efficiency so to not waste their resource of energy.

After some research on pathfinding algorithms, I settled on two algorithms that could serve my purposes: A star and Dijkstra's Algorithm. These could both work but they were differently suited for different scenarios.



Dijkstra's Algorithm checks nodes radially outwards from the player until it reaches the goal. This is most effective to find a path through mazes and where there are long dead ends as it can effectively check every pathway at the same time. This however makes it less efficient in more straightforward scenarios since if the algorithm will be checking a large number of unnecessary nodes.

A star algorithm works in the opposite way. Instead of checking every node, it searches the most likely node to reach the goal by giving nodes 2 cost values. It uses these heuristics to create lists of checked nodes until the goal is reached. **(Daniel, 2018)** It has 2 main

heuristics Gcost and Hcost which are combined in the end to make the Fcost. G cost is the distance from the starting node. Hcost is the Manhattan distance from the end node where Manhattan distance is the distance strictly using vertical distance + horizontal distance.

The adjacent nodes from the starts calculate the Fcost and the lowest cost gets added to the Open list with all the nodes to be searched and a Closed list with all of the nodes to not be searched again. After a path is found, the Open list is reversed, and the best path is found. Since it is always making the most informed decision at any given search and anything wrong is removed from the open list, the algorithm will find the best path simply by removing any inefficient paths and finding the first solution it can.

I was not able to successfully implement the A Star algorithm in my timeframe for the project, so I settled to give the player freedom to move tile by tile so that energy wasn't wasted by something outside of the player's power.

Inspiration

Minimum Viable Product

The minimum viable product which is the smallest amount of functionality that the product could be made with to demonstrate the concept, would be: a grid-based map, created by a string of characters representing tiles, the digging system and storing treasure earned, energy system that takes energy to move and dig.

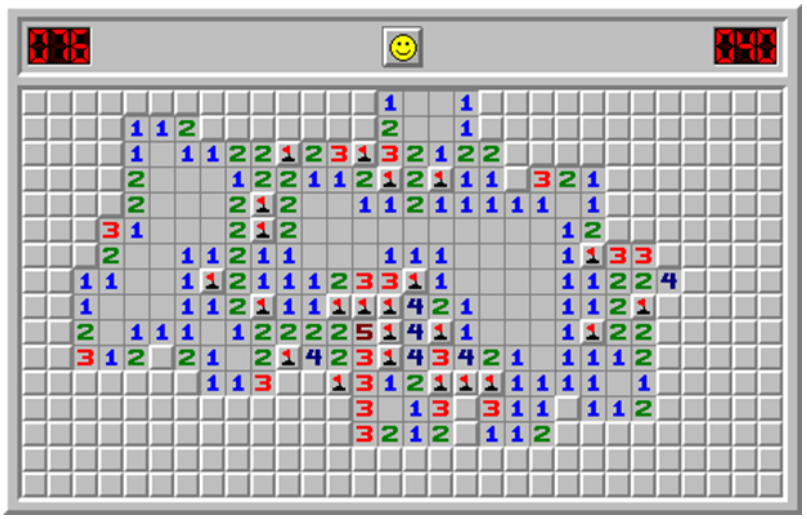
The treasure selling and home building, while important to the game economy, are not essential to creating a proof of concept for the core gameplay loop as they only serve to enhance it and make it more efficient.

Potential Expansion

The potential for expansion for this game is large and can delve into procedurally generated maps and hints similarly to Sea of Thieves. Different tools and material types could create a further layer of depth to the gameplay where the player would invest in mining with pickaxes in stone instead of digging with shovels in sand and soil. Some online sharing functionality to show off to other people or compete with scoreboards could be possible to create a social aspect of the game.

Games Inspiration

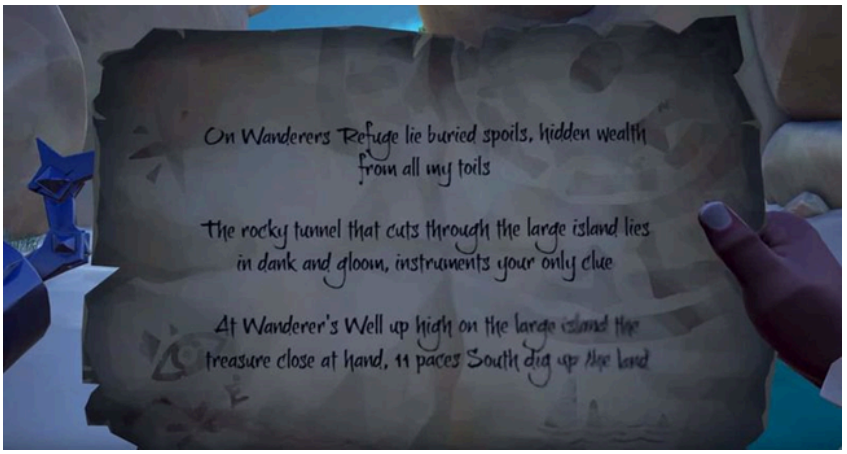
Minesweeper's numbering system is a large inspiration for the game. A number system like this was chosen because it created a sense of intentionality to one's actions. Instead of randomly digging up tiles, a player could calculate chances and more likely tiles to find treasure in. So, thinking first before digging always yields greater results.



Legend of the Golden Robot is an old flash game that this is mostly based on that has time in the day and different material and treasure digging. This game is the base inspiration that first exposed me to the reverse minesweeper grid system but on such a small grid, the concept seemed wasted and random digging seemed to be just as effective as a measured approach.

Sea of Thieves is the main inspiration of island treasure hunting and the secret hints to buried treasure. This serves to add complexity to the game by adding another layer to randomly generated loot. It is a simple puzzle with a rhyming set of instructions which

challenges the player to solve a different style of puzzle to find a large reward. Instead of a numerical puzzle, it is a written one, so it diversifies the types of challenge in the game to add depth.



The aesthetics of the game would be inspired by another flash game called Castaway. This game uses a grid system and also pulls off a tropical Caribbean Island feel which is what I am trying to replicate through the scenery choice, obstacles, and colour palette. Shallow waters coloured turquoise instead of a darker shade of blue, golden beaches and lush green palm trees paint a vivid picture that I attempt to emulate.

References

Amit Patel. (2010). *Pathfinding.* Available: http://theory.stanford.edu/~amitp/GameProgramming/index.html. Last accessed 24th Dec 2021.

Daniel. (2018). *Unity - A Star Pathfinding Tutorial.* Available: https://youtu.be/AKKpPmxx07w?t=81. Last accessed 24th Dec 2021.

Daniel Cook. (2012). *Loops and Arcs.* Available: https://lostgarden.home.blog/2012/04/30/loops-and-arcs/. Last accessed 24th Dec 2021.

Ian Bogost. (2011). Relaxation. In: *How to Do Things with Videogames*. Page 92.

Jesse Schell. (2008). The Game Consists Of Elements. In: *The Art of Game Design : A Book of Lenses*. Pages 54-57.

Kyle Wilson. (2002). *Data-Driven Design.* Available: http://gamearchitect.net/Articles/DataDrivenDesign.html. Last accessed 24th Dec 2021.

Ross Metcalf. (2018). *Design Pillars.* Available: https://www.rossmetcalf.com/post/design-pillars. Last accessed 24th Dec 2021.

Scott Shumaker. (2001). *Techniques and Strategies for Data-driven design in Game Development.* Available: https://web.eecs.umich.edu/~soar/Classes/494/talks/Schumaker.pdf. Last accessed 24th Dec 2021.