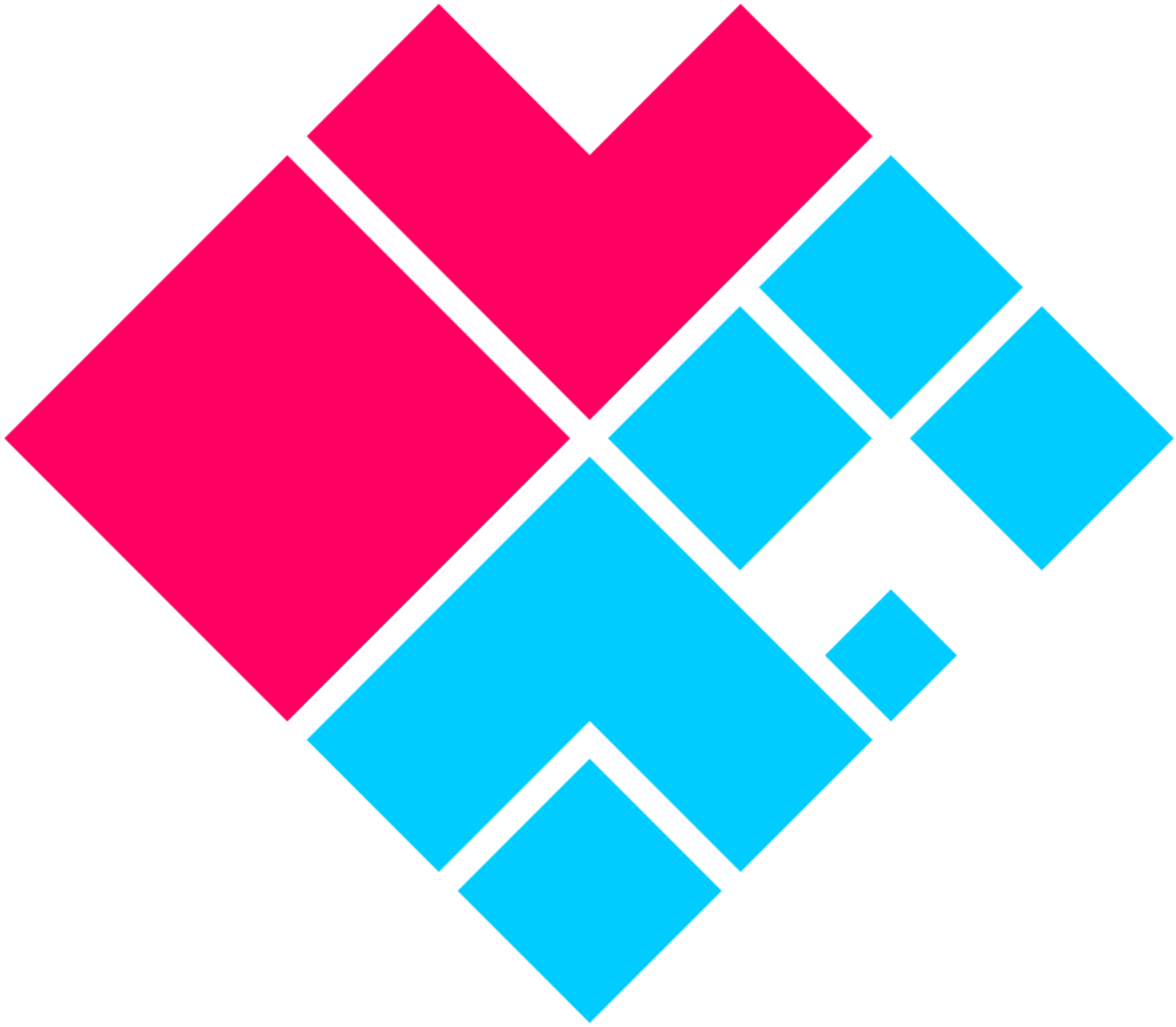
Alice in Wonderland



By:

Alexia Soucy: Art direction & assets
Earl Steven Aromin: Project management & programming
Jesse Tremblay: VFX implementation & QA
Marco Tropiano: Sound design & composition
Tan-Phat Pham: Programming & QA
Thomas Backs: Programming & QA

Table of Contents

1. Executive summary....2
2. Overview....2
3. Related games....2
    3.1 Enter the Gungeon....2
    3.2 The Binding of Isaac....3
    3.3 Hyper Light Drifter....4
4. Player composite....5
5. Game World....6
6. Characters....6
    6.1 Alice....6
    6.2 Pawns....6
    6.3 Rooks....6
    6.4 Knights....7
    6.5 Queen of Hearts....7
7. Art direction....7
8. UI Storyboard....9
    8.1 Title Screen....9
    8.2 Game Screen....9
9. Technology plan....10
10. Controls....11
11. Level design....12
    11.1 Level Generation....12
    11.2 Level Design....14
12. Mechanics analysis....14
13. Schedule....17
14. Music direction....18
15. Change log....19

1. Executive summary

A roguelike hack-and-slash game, the protagonist name is Alice and the story takes us in a near future where Alice, a prominent hacker meddles and attract attention of some dangerous mobster. Alice has no other choice to hack and slash her way in the mobster database to uncovers their schemes and saves her life... Will you be the one who will help Alice succeed?

2. Overview

The player controls Alice movement and actions, the character has four primary actions available in combat, which are melee attack, ranged attack, and evasive ability. She also has an ability that is charged by Blackjack cards game system, the closer the player is to 21 the stronger Alice ability is. The game is a top-down view games and it plays like “The Binding of Isaac” meets “Hyper Light Drifter” with the “Blackjack” card game, it also take some inspiration from Enter the Gungeon as well. This game is a fast-paced action, it will please dungeon crawler fans as well as roguelike game style fans. We also appeal to cyberpunk world to reflect the future aspect of the game. The art is minimalism and stay true to cyberpunk environment and art style, we focus a lot on the combat mechanism to allows player to experiment with wide possible approach to each fight and to rationalize and take quick decision/reaction in combat as well, granting the player 30 seconds grace between actions allows the game to remains fast-paced and more interactive.

3. Related games

3.1 Enter the Gungeon

By Source (WP:NFCC#4), Fair use, https://en.wikipedia.org/w/index.php?curid=50064987

Enter the Gungeon is a bullet hell dungeon crawler games, this game is developed by Dodge Roll that run on Unity Engine. The game is published by Devolver Digital, the initial game release was on April 5, 2016 on PC. This game is now ported on every console with the most recent release on Switch. They sold over a million of copy of this game, it is considered a huge success. Metacritics website rate this game 84% based on 26 critics while it reach 77% from User score based on 192 ratings.

We will take inspiration from this game for our map and level layout for our game as shown in *Illustration 1* below, Enter the Gungeon uses simple layout, we also take the twin-stick shooting style from this game and it is heavily favoured and loved based on critics from Metacritics website. This is considered a really challenging game which appeal to some fans as well, in our game we will try to provide a good challenge to our player while still maintaining the appeal to more casual player. We will not take the bullet hell mechanism from this game because we feel that it will be too crowded for our game and it doesn't feel like a good fit for this. The bullet hell approach is not appealing nor an interesting implementation into our games. Bullet Hell has been viewed as a downside for this game due to its high difficulty as well, in our game we will reduce the bullet hell amount.

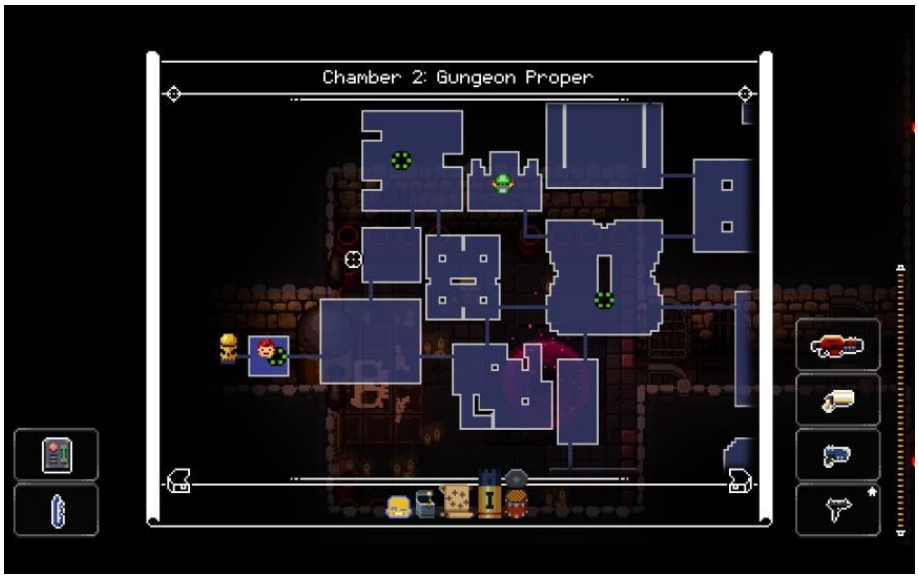


Illustration 1: Map and level Layout

3.2 The Binding of Isaac

Illustration 2: By Source, Fair use, https://en.wikipedia.org/w/index.php?curid=38131772

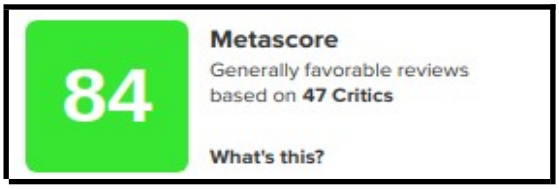
The Binding of Isaac is a roguelike video game, its plot is taken from the Bible. The game is developed by Edmund McMillen, an indie developer, he is also the publisher and the designer as well. This game has been released through steam on September 28, 2011. it runs on various PC platform, it uses Adobe Flash engine. By July 2014, it has sold over 3 millions of

copy, this game contributes heavily in renewal interest in the roguelike genre. Based on Metacritic website, it got a Metascore of 84% based on 30 critics, and a User score of 84% as well based on 1035 ratings!

The game uses 2d sprites just like our project, it uses top-down view as well. We incorporate many feature from this game in our project, such as level design, twin-stick shooter style again. The concept of clearing the room before going into another room, and small inspiration for the power-up, the main difference between Binding of Isaac power-up and ours, is how to use them, we will use these during the fight via the Blackjack card mechanism, and we would purchase upgrade between level.

Illustration 3: http://images.pushsquare.com/screenshots/63467/large.jpg

3.3 Hyper Light Drifter



Hyper Light Drifter is a 2D action RPG, this game is a homage to 8-bit and 16-bit video games, it is a combination of The Legend of Zelda: A Link to the Past and Diablo. It has been developed by Heart Machine, the publisher is Playism. It is available on all platform from PC to last-gen console. The initial release date is March 31, 2016. It use GameMaker Studio engine for its development. The criticism are mixed for this game about minimalism of the game storytelling method, we will try to improve the game storytelling in order to improve player experience and immersion to this game. Metacritic gave a Metascore of 84% based on 47 critics while the user score is 83% based on 277 ratings. It has won two awards at Independent Games Festival in 2016, one for Excellence in Visual Art and the other for Audience Award.

Our game is heavily inspired from Hyper Light Drifter, our protagonist shares many common aspect to this game, such as energy sword, as well the dash mechanism. Our main inspiration is without doubt form Hyper Light Drifter for the character movement and actions.

Illustration 4: https://www.mobygames.com/game/windows/hyper-light-drifter/screenshots/gameShotId,856270/

4. Player composite

John Doe, a 27 years old male, enjoy playing some video games after long day at his office works. He is in a happy relationship with his girlfriend, live with her in a small apartment downtown. He enjoys game that bring some nostalgia from his childhood game, he remembers spending countless hours on ChronoTrigger trying to beat, a big fan of top-down single-player game. He only plays games that has controller capacity. Simply put, after a long hard day of work, he comes home, have a nice meal with his girlfriend, crack open a beer, hit the couch, take controller and play some quick games.

Mike Smith, a 23 years old male that is a student and a part-time worker at a retail store. He is someone that enjoys video game with or without company. While his priority is mostly school and work, he uses his free time to play games such as super Mario, Mega man and many more. The genre that he prefers playing would be action/platformer since his parents gave him an SNES as a Christmas gift when he was younger. After a long week of work and school, he would spend the weekend relaxing by either going to a friends place or stay at home playing the video games that he likes.

Most of our target audience are people who own a computer and want to feel the nostalgia and playing fast-paced game, the resurgence of roguelike allows to merge these two feeling, the players mostly enjoy single-player game and wants to immerse in good storytelling with a rich combination of attacks

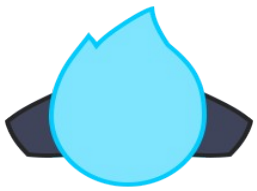
5. Game World

A young talented hacker named Alice who enjoys doing penetration testing, has attracted the attention of the wrong kind of people, a dangerous criminal organization who launders money through encrypted online bank transfers between their casinos. They noticed Alice's skills which allow her to de-encrypt and uncover their scheme. The crime organization made an offer to her, which she refused. The crime organization doesn't take 'No' for an answer, Alice has no other choice, she must save her life, she must hack into the shady organization database via their network, to uncover their schemes to the world, in order to achieve her goals, no one must stand in her way.

6. Characters

6.1 Alice

A young talented hacker, Alice loves doing penetration testing and trying to access various server, she has strong moral, she only does that for challenge purpose, Alice would never uses her skills for bad thing, this is why she denied the mobster offers to work for them because she knows the mobster will uses her for their money laundry.



Alice

- VPL Blade, a energy sword which will be used for her melee attack
- psi bolt, a ranged attack, Alice launches a bolt in the direction she is facing
- Dash, this ability allows Alice to dashes toward the direction she is facing to avoid enemy attack, she is also invincible for the dashing duration.

6.2 Pawns

Pawns are simple programs that run once, it is execute to run toward the target and self-destruct in order to eliminate the intrusion. The AI is quasi non-existent, they cannot perform any tasks, the Queen of Hearts disposes of them without blinking, they are cheap programs, with no other purpose to self-destruct themselves on intrusion.

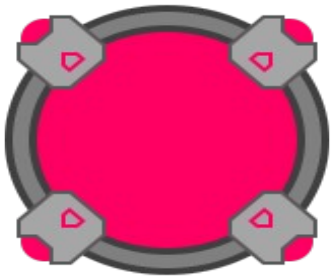


Illustration 5: Enemy Pawns

6.3 Rooks

Rooks are programs, a basic defence AI system remotely controlled by the Queen of Hearts, not a very advanced one, Rooks just shoot bullet in one direction when the alerts is triggered whenever a intrusion is detected. It is stationary. Its sole purpose is to defend the servers and prevent intrusion.

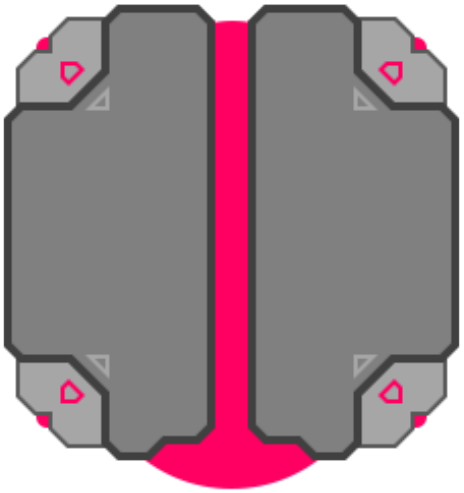


Illustration 6: Enemy Rooks

6.4 Knights

Knights are the most advanced intrusion-prevention programs, its AI mechanism is the best of these options, it can detect any intrusion, roam around and do some range attack toward the target and move away from it. It is the strongest AI in the game, it runs really efficiently and quickly, they are the Queen of Hearts favourites program.

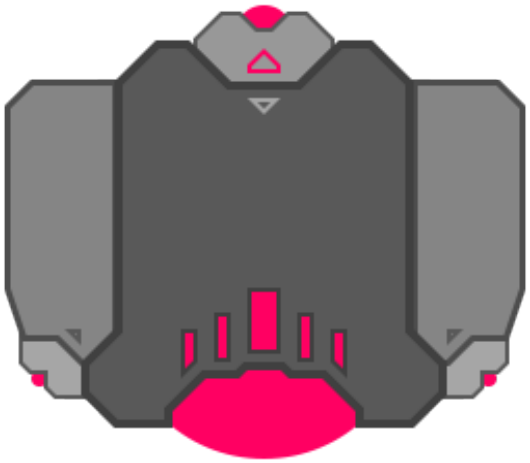
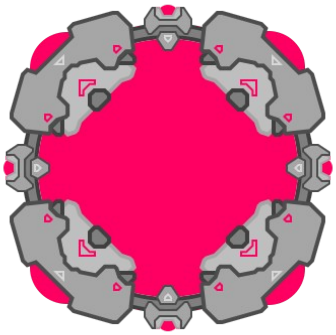


Illustration 7: Enemy Knights

6.5 Queen of Hearts

The Queen of Hearts is the boss of the game, she is Alice foe, her objective is to prevent Alice to reach casino server to fulfill her mission, hence protecting the money laundry server. The Queen of Hearts is an overseer AI that control all intrusion-prevention programs that runs on the mobster server. She only appear in the boss stage fight when Alice managed to beat all programs that stood in her way. The boss has heart beating in the middle, it will beat faster when her HP has reduced in halves, and it will enter in the second phase of combat.



7. Art direction

Alice in Cyberland's art direction is primarily minimalist, representing the game world through shades of grey and key focal points in either cyan (#00CCFF) or magenta (#FF0062), based on whether they are beneficial or not. This approach makes it easier for the player to spot important information at a glance, which is necessary because of the game's fast pace and mechanical complexity (i.e. focusing on both combat and blackjack at any given time). All text is represented with Lucida Console to match the square art style of the interface and to suit the digital setting.

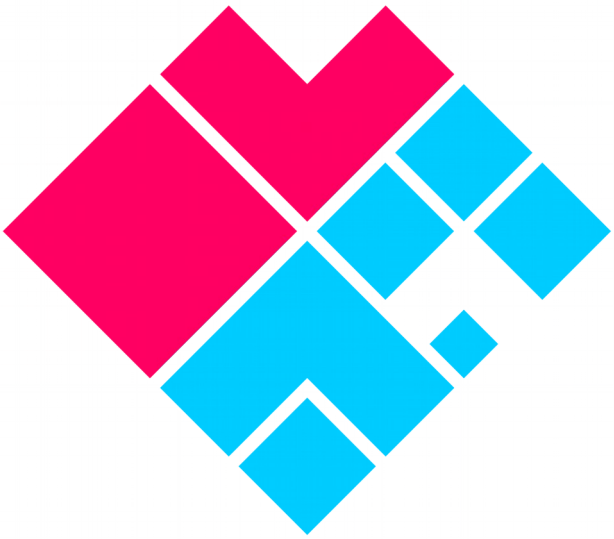


Illustration 8: Game Logo

The game is played in an isometric top-down view. This is because Alice's omni-directional movement in the game would require several times more assets to be created for each action if for example an isometric 3/4 view was used. Because the perspective is top-down, character designs must be distinct

from the top. To that end, Alice's primary characteristic is her neon hair against her otherwise relatively dark and desaturated suit. The enemies use block grey shapes with magenta neon accents to differentiate them and make them pop.

Illustration 9: Alice art

Entities that move in the world, like Alice, enemies, projectiles, and the Vorpal Blade, are followed by light trails to emphasize movement. This makes it easier to represent movement while using the isometric top-down perspective. Alice's light trail in particular represents her hair and gets longer as she grows in power.

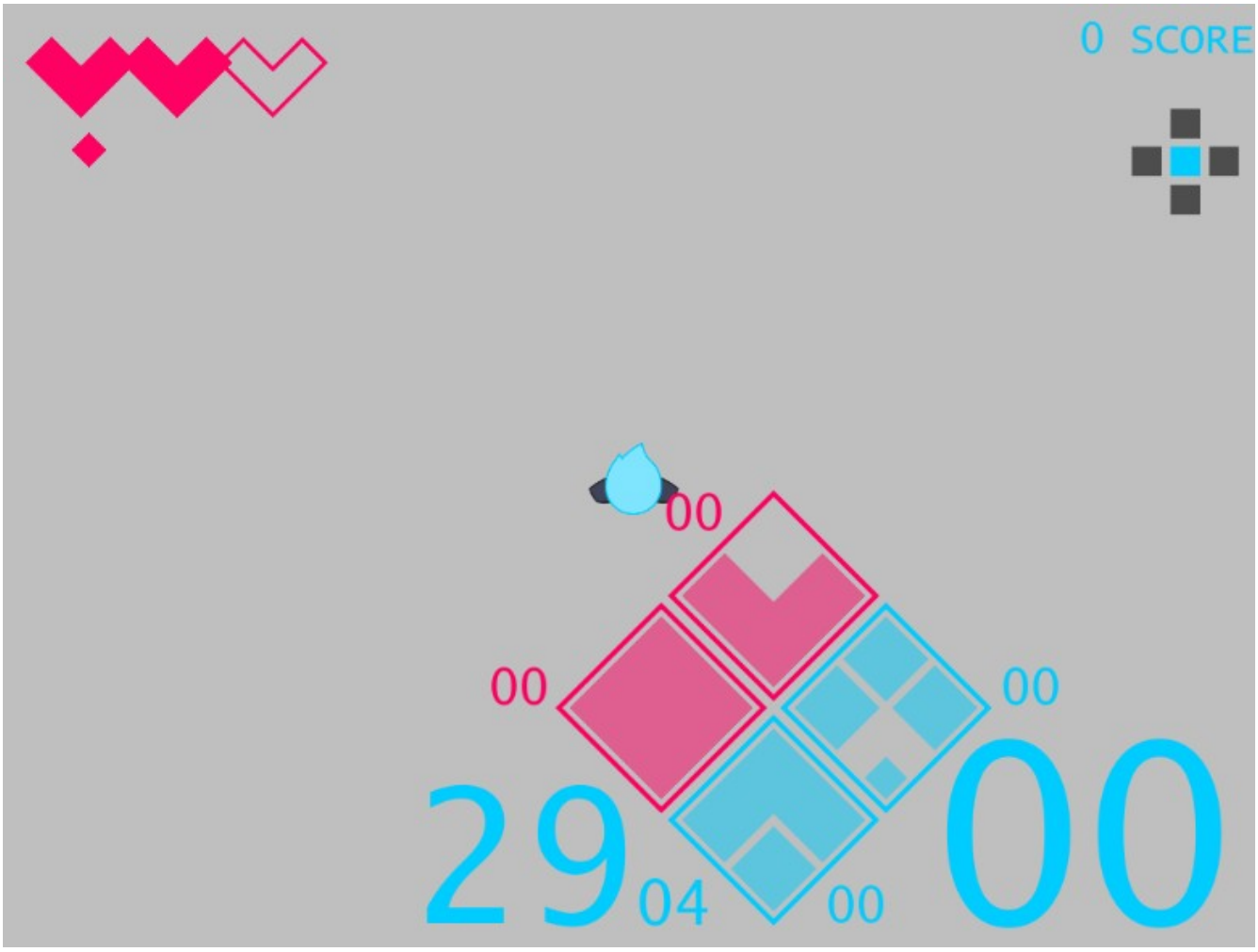
8. UI Storyboard

8.1 Title Screen

Alice in Cyberland Main Menu screen is minimalism, a black background with the game logo with few button provided to the player, Start Game, Controls, Credits and Exit Game. The buttons are magenta (#FF0062) and become cyan (#00CCFF) when it's highlighted. The font used is called *lucon*. Whenever the game is over the player will go back to the Main Menu. The player also can go back to main menu by pressing "start" button on his controller. There is a small animation playing at the bottom of screen to give a more lively looks to the Main Menu.

As explained above, there is 4 different buttons choices offered to player, the highlighted one will be cyan and the non-highlighted will be magenta.

8.2 Game Screen



As shown above, the game UI, Alice is in the middle of the screen and will always be in the middle, on bottom right corner you have the Burst Meter with the timer (on the left) and the total value shown in cyan. On top left we have 2 thing. The first one is the heart shaped icon which represents health that Alice has. Right next to it, there is a magenta diamond icon (it will increase up to 4 with buff), which are our number of charge available for dash. You can see below for more information about the UI.

- Represent burst meter of each suits of cards
    - Top position: Heart
    - Left position: Diamond
    - Right position: Club
    - Bottom position: Spade

As player draw cards, corresponding suit bar will increase until it reaches the maximum then apply the burst overhead, buff player.

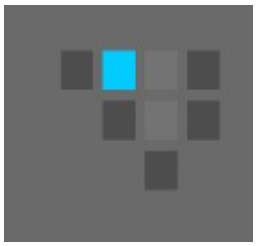
The sums of current card that is out on the UI is right next the the burst meter on its right in cyan.

There is a player health bar that display the current health point of the player, it is displayed using heart icon like this:

Right next to the HP bar, there is some diamond shaped icon that represent dash.

There is also a timer display that will let player knows how long he is allowed to keep the Blackjack card states without action before losing the buff. Once the timer will go below 10 seconds, it will beep and turn to magenta to attract player attention to it. The timer will reset after each action performed or when the timer reaches zero and all buff is lost, and the player receives some damage. It is shown right next to the Burst Meter on its left.

On top Right you have Score, it keeps track your current score based on enemies eliminated, and right under it you have your minimap, the minimap is a really minimalism, the blue square indicates the current player position, the dark grey square indicates unexplored room while the light grey square means that room has been explored and cleared as well. The minimap is generated each time the game restarts and the score will be kept as long as you don’t leave or die.



Minimap

9. Technology plan

This game is an in-house project, meaning there will be no outsourcing work. The arts are done entirely by Alexia Soucy, she is in charges of art direction and assets. The technology used for art creation is Adobe Photoshop. The game development will be entirely done using Unity engine, which is a game-oriented engine that provides rich and powerful features to help us making our game. Unity also provides strong re-use features called Prefabs. We will use GitHub which is a web-based hosting service to centralize and for version control with the use of Git command. It offers a distributed version control as well as source code management. We will also use the Issues features provided by GitHub to keep track of any issue and problems we encounter during the game creation process. We also be able to keep track of all changes happened of our project with the commits tracking that GitHub provide. We will use C# for our scripting development, the game will be developed on both Linux and Windows environment, so we give the freedom of IDE to our programmer as long as it supports C# and Unity. The hardware used for development will be workstation such as laptop or PC, Internet connection to be able to synchronize and pull the project repository on GitHub, we will also need an Xbox controller to test game control input.

10. Controls

For the player inputs, the game will use a game controller for user inputs, we choose controller because it is easy to reproduce the twin-stick feeling with a controller rather than with keyboard. The picture reference shown below use an Xbox 360 controller which is primarily used during development, but any controller that is compatible with computer should be fine.

Left stick: Player's movement

Right stick: Player's rotation

Left Trigger (LT): Player's ranged attack

Left bumper (LB): Player's dash ability

RB button: Player's melee attack

Y button: player's draw card

B button: Vent player's deck

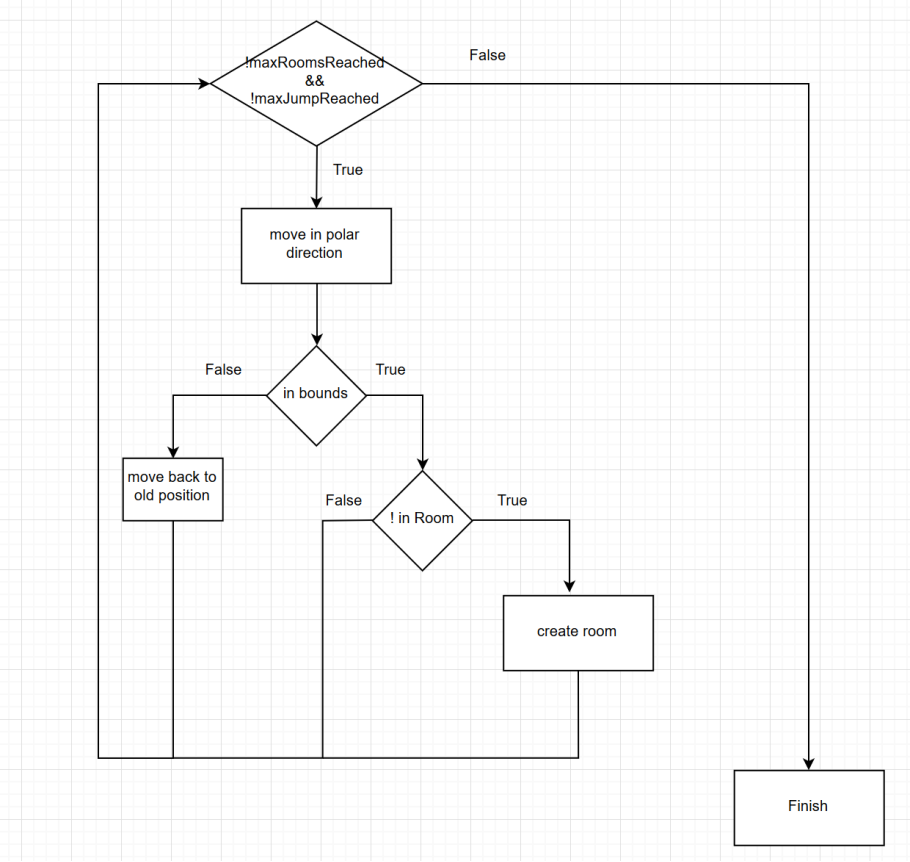
The left stick records player's input on horizontal and vertical axis and execute the appropriate movement direction toward where the left stick is being pushed. The right stick records player's input on a left or right facing direction. The dash ability is execute only when the player pressed the left bumper button. When the player press 'X' button, Alice delivers a strong melee attack with her VPL blade. There is also two button to interact with the Blackjack card system, 'Y' and 'B' button. The 'Y'

button, when pressed, will make the player draw a card from the existing deck that will be shown on screen, perform some addition and display current value. Then the player could Vent the card he currently hold by pressing the ‘B’ button which will discard *all* cards that player have in his deck and in return, grant a burst to fill her burst meter to apply some bonus depending what type of cards being discarded.

11. Level design

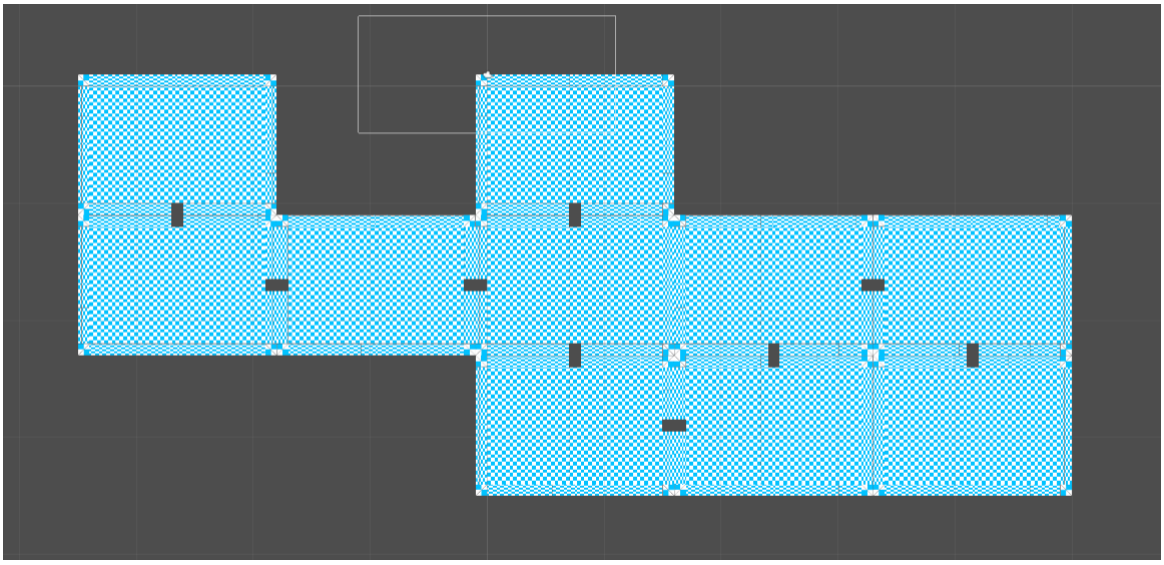
11.1 Level Generation

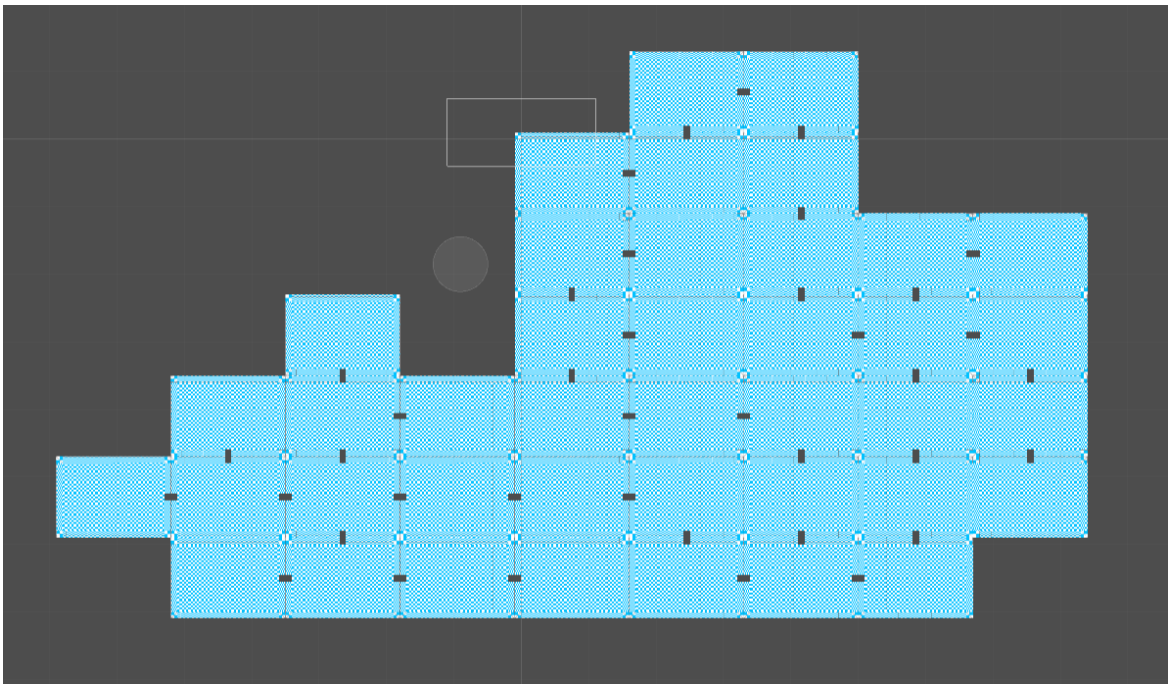
Alice in Wonderland’s game level consist of procedural generated floors that are created from a set of fixed dimension rectangles called rooms. Each of these rooms contain a layout such as a combat encounter that is picked from a pool of bespoke layouts. One problem with this is generating floors that have compelling maze-like layouts while also ensuring the level is completed and all rooms can be accessed. Our solution to this is a simple algorithm that can quickly create floor layouts as illustrated in the illustration below:



This approach ensures all rooms are accessible because the floor generation is essentially done on a linear path. It is comparable to tracing a line on paper while never lifting the pen, thus every point on t he line can be reached from any other point. Doors on rooms are created thanks to the factory design pattern allowing rooms to be created by connecting them to other rooms. This algorithm is bound by 2 values the total number of rooms which is self-explanatory and the total number of jumps which is how

many attempts can be made to create rooms. This is done to ensure that the algorithm does not loop endlessly due to its reliance on random values for where each room will be positioned. Example results of this algorithm can be seen in these two illustration below, the first one illustrate small floor and the second one large floor.





The data structures involved in this generation are the Floor which runs the room generation algorithm, the RoomFactory which manages relative room positioning and door opening as well as a RoomMap which is a NxM matrix of rooms. This matrix has an iteration function that can be used to go through each room one by one and execute some form of logic on them such as assigning a suitable layout based on its position or toggling doors.

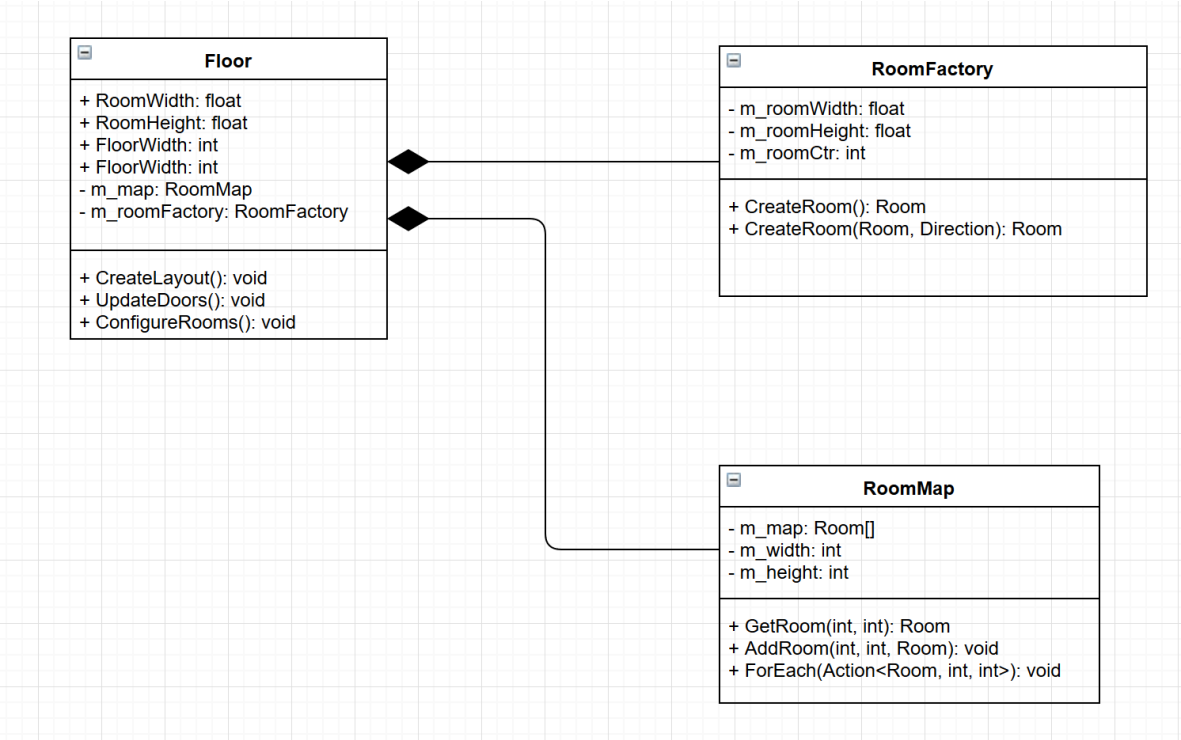


Illustration 10: Floor generation UML

11.2 Level Design

Since Levels in Alice in Cyberland and broken down into multiple rooms that are procedurally assembled. Our core level design is designing the specific rooms that compose a level. These rooms need to be self-contained fast paced combat encounters that each last from 30 seconds to 1 minute. The rooms will consist of props that acts as obstacles for the player and enemies as well as enemy spawn points.
Building rooms with these primitive building blocks will allow us to quickly create and iterate on room ideas. This allows us to easily make the game feel more varied simply by adding more room layouts into the floor generation process. Additionally, the floor algorithm will need to accommodate things like keys that will enhance the strategizing required to reach the end of a floor. Each floor will conclude with a boss that is a special combat encounter meant to be the ultimate test for the player.

12. Mechanics analysis

It is no secret that we heavily took our inspiration from roguelike game such as Enter the Gungeon and The Binding of Isaac, our shooting mechanic is heavily inspired from these two game, it is a twin-stick shooting game, meaning the player will use both left and right stick, the left for movement and the right for aiming. This provide a nice change of pace from the standard first and third person shooter games. This mechanism allows the player to move AND shoot at he same time which is one of best features of twin-stick shooting mechanism, it added more fluidity in the movement and improved shooting angle too. Our player will have various abilities to use along his twin-stick shooting action, such as using ranged attack, melee attack or simply dashing to evade attack. The balanced will be respected, such as certain enemy use shields to block ranged attack, then it forces the player to uses melee attack to break

shield to be able to use the ranged attack, the action combination is what improves the player's experience and provides rich set of mechanism. Another balance factor is the damage done by melee and ranged attack, a melee attack will be more powerful than ranged attack but ranged attack can be done in a safe ranged while melee must be close to the target which might not be always ideal against the Pawns who can self-destruct themselves on the player.

The player will be rewarded for its creativity and trying various combination to overcome enemy obstacle, the Blackjack mechanism will also provide some boost to help the player to beat more difficult enemy, using all these mechanism efficiently together is the keys to success.

We will also have the cards mechanism which will depend heavily on randomization to simulate the shuffling of deck. These cards will be drawn and added using the Blackjack card game mechanics which is basically assigning value from 1 to 11 based on cards value. All faces cards such as Jack, Queen, and King worth 10 and the aces worth 1 or 11. The objective is to not go over 21, but as close as possible for best output. On the left of the card deck there will be a meter that fill up with card values, the better cards are, the more quickly the meter fills up, below is an illustration of the empty meters, the sprite-sheets used for the meter and an example of partially filled meter.

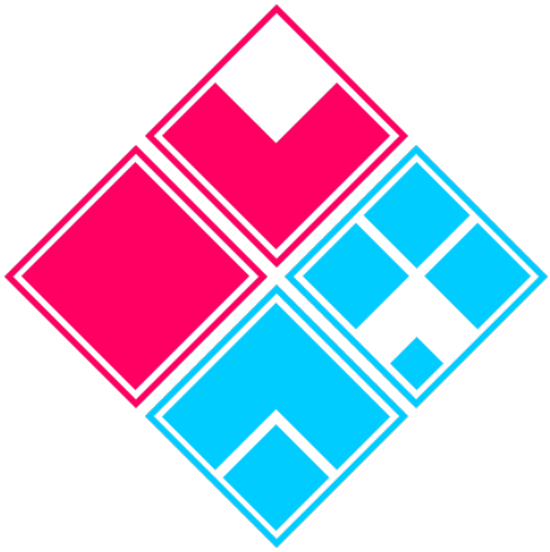


Illustration 11: What the meter looks like

Illustration 12: Partially filled meter

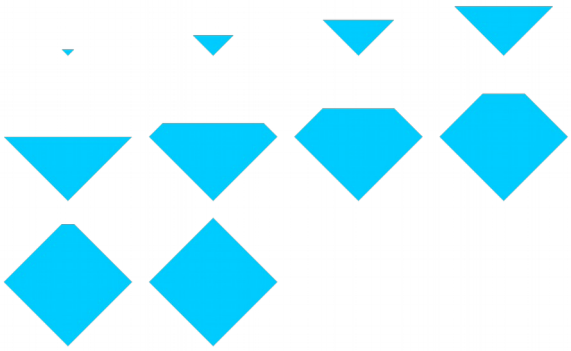


Illustration 14: Blue filling sprite sheet

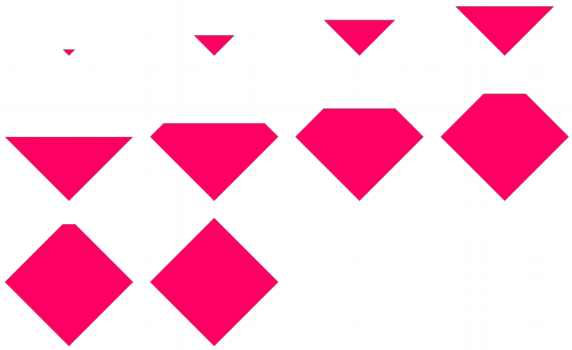


Illustration 13: Red filling sprite sheet

13. Schedule

	October				November			
Task	Week 1	Week 2	Week 3	Week 4	Week 1	Week 2	Week 3	Week 4
Min. Viable Product								
Core Gameplay(Alice)								
Core Gameplay(Alice) QA								
Map Generation								
Map Generation QA								
Enemies								
Enemies QA								
Boss Fight								
Boss Fight QA								
Menu Implementation								
Integration								
Integration QA								
Gameplay Feel (Tweaks from QA)								
Changes from original plan: Most features were stretched out past the first estimate. Separated small enemies from Core Gameplay for precision.								
-SFX and Visual asset creation are left out as they are created as needed, so they span the whole project lifecycle								

In terms of the schedule October week 1 through week 4 all has been completed except for the boss, menu and integration, which has been implemented and tested in the month of November as you can see in the illustration above. The tasks to merge everything together has been done roughly weekly, each branch has been merge at least once a week to GitHub repos as well some merge fixes and cleaning up.

Staffing Plan: Overtime since the beginning of the project, we have thought about the type of staff that we would need with different skill to achieve a proper product. As decision we have an artist which will deals with most of the visuals for the game, a music composer that would handle the ambient of the game and action sounds as we progress, a VFX and level designer that would take care of the level creations and effects needed, lastly we have programmers that will deal with the core component of the game and QA tester to help double check if the game has any problems. The current status of the project is final release, there might still be some minor bug that will be release in subsequent patch and fix, the testing is done thoroughly and all required component are added to our game-oriented

14. Music direction

The setting of the game is in the future, so it is only fitting that the background music will use many electronic synthesizers to produce futuristic sounding audio. Dance music it seen as the “music of tomorrow” so the tracks will follow a similar structure. Another aspect of the setting that brings inspiration to the background music is the underground vibe. Dark and deep melodies will play a part to enhance this.

When the main protagonist is facing a boss, the music will change to a more intense one. There is a technique used by sound engineers where they make melodies that are slightly out of key to make the listener uncomfortable and tense up. This will add extra depth to the world being shown.

15. Change log

[2018-10-5]: We decided to remove the shield mechanic after the proposal pitch to reduce the overflow of user input and actions to improve our game mechanism.

[2018-10-5]: Reduction of number of levels, initially we wanted multiple levels but instead we will have one big level that contains many areas that is locked by doors, which can be unlocked during the progression of the game.

[2018-10-25]: Complete removal of upgrades shops due to time constraint, it will take a lot of time to balance upgrades and implementing different kinds of upgrades and condition, etc. The upgrade shop is completely removed and we will rely solely on Blackjack system for our buffs. It will also clear some ambiguity that been shared from few customers.

[2018-11-12]: Final implementations of three kinds of enemies, change mechanism of Knight, will now be roaming around and if the target is in the range shoots at target while backing up.

[2018-11-14]: Added the capacity to give a list of prefabs to the floor to populate rooms with, and add more exits in room.

[2018-11-15]: Change the timer from a slider/bar to a numerical instead to be easier on the eyes

[2018-11-26]: Remove the Card Deck displayed on top-right corner, it is not really needed since the players only need to know the burst meter and the total.

[2018-11-27]: Finalization of the project, added sprites final arts, SFX, VFX and playtesting. Fix any minor bug issue