**The Untitled Rogue-lite Game**

**Agenda**

[I. Summary](#Summary)

[II. Story](#Story)

[III. Gameplay](#Gameplay)

[IV. Level Design](#LevelDesign)

[V. Art](#Art)

[VI. UI / Game Control](#UI)

[VII. Audio](#Audio)

[VIII. Development Plan](#Plan)

**Summary**

*The Untitled Rogue-lite Game* is a top-down action rogue-lite game on mobile. The game will take players into a fantasy world to explore the map, collect loot, defeating enemies while dodging attacks.

With the intention to encourage exploration, *The Untitled Rogue-lite Game* wants to combine a typical roguelike action game with a slot-based inventory system and looting. Players will be challenged to choose what weapons, gears and items to carry with them in the next battle, and what to discard.

* Randomly generated levels and permanent death.
* Replacing virtual joystick with gyroscope & accelerometer for more intuitive movement control.
* A large variety of weapons to explore.
* Auto-aim mechanism for intuitive control.
* Dodge & roll to enable players with better control of the character.

*Minimal Viable Product*

* Combat system implemented
* Core loop completed
* One weapon
* 2 ~ 3 types of enemies

*Engine Feature Needed*

* Multiple touch input
* Gyroscope support
* Export project for Mobile
* Improved Pathfinding system (low priority)

**Story**

Setting

Narrative

**Gameplay**

Core Loop

BOSS?

YES

Player Enters a Level

NO

Exit to Next Level

Mechanics / System

i. Combat System

*ATTACK*

Attack will aim automatically. Player will only need to tap the phone screen to launch attack to the most-nearby enemy. All weapons have two attack mode: light attack and heavy attack. (reference: Soul Knight <https://www.youtube.com/watch?v=ExoeMjEq4Go&t=1s>)

Light Attack: Players tap on right side of the screen to perform light attack.

Heavy Attack: Player tap and hold on the right side of the screen to perform heavy attack. Heavy attack consumes Stamina.

*DODGE & ROLL*

Dodge and rolling will improve character’s mobility and grant IFrames (Invincibility Frames) at the startup and active animation, but not at the recovery animation. Dodge and rolling also consumes Stamina.

Players tap on the left side of the screen to perform dodge & roll.

*STAMINA*

Performing heavy attack and dodge & roll will consume Stamina. Considering Stamina as an action point. One Stamina bar will allow player to perform either one heavy attack or one dodge & roll.

Stamina will start to regenerate one second after player stop performing actions that consume Stamina. One Stamina bar will be filled after one second.

Max Stamina Bar can be upgraded from 3 bars to 6 bars max.

( reference: Curse of the Dead Gods <https://youtu.be/g2AfGcGvOfI?t=107> )

ii. Items

*WEAPONS*

Weapons grant character different light and heavy attack.

*GEARS*

Gears grant character different Stats Upgrade.

*CONSUMABLE ITEM*

Usable items

iii. Slot Inventory System

Gear slots allow players to carry/equip/use items. Players swipe left/right to select the current active item.

If active item is a weapon, players tap/tap & hold to perform light attack/heavy attack.

If active item is a consumable item, players tap to use the item.

Click an item on the map to pick up. If inventory is full, replace the current active item with pickup item.

Inventory slots can be upgraded from 2 to 4 max slots.

Character Progression

Enemies

**Hobgoblin** Attack Mode: Melee

Max Health: 50

Move Speed: 1

Turning Speed: 900

Search Range: 5

Reset Radius: 2

Move towards and follow the player once player moved in the search range.

Attack when move close enough to the player.

**Goblin Chief** Attack Mode: Ranged

Max Health: 30

Move Speed: 1

Turning Speed: 800

Search Range: 6

Reset Radius: 3

Move towards the player if player is out of attack range; move away from the player if player is closer than the attack range.

Attack by firing one projectile three times consecutively (one volley of shots). Attack will have 3 seconds cooldown.

Cannot move while attacking. Move towards or away from the player after finish one volley of shots.

**Level Design**

We are closely following how has *Spelunky* procedurally generated its level with both hand-crafted templates and randomly generated layout. The link below explains in detail about how do *Spelunky* and we approach procedurally generated levels: <http://tinysubversions.com/spelunkyGen/>.

TOP

1

TOP

2

TOP

3

MID

4

MID

5

MID

6

BOTTOM

7

BOTTOM

8

BOTTOM

9

Each level is made of 9 rooms in a 3x3 grid. There are 4 different basic room types:

1. Secret Room (SR): a side room that is not on the solution path. A secret room can be spawn in TOP, MID and BOTTOM row. A secret room will have only one guaranteed exit randomly chosen between the sides that adjacent to another room, if there is not already an exit connect to it. Example, if ROOM 4 is chosen to be a secret room, it will randomly generate one exit to connect either ROOM 1, ROOM 5, or ROOM 7, if there is not already an exit for ROOM 4.

2. Pathway Room (PR): a room that is guaranteed to have only a left exit and a right exit. A pathway room can be spawn in TOP, MID, and BOTTOM row.

3. Connector Up Room (CUP): a room that is guaranteed to have only a left exit, a right exit, and a top exit. If there is another Connector Up room below it, then it also is guaranteed to have a down bottom exit (becoming a Crossroad Room). A connector up room can be spawn in only MID and BOTTOM row.

4. Connector Down Room (CDR): a room that is guaranteed to have only a left exit, a right exit, and a bottom exit. If there is another Connector Down room above it, then it also is guaranteed to have a top exit (becoming a Crossroads Room). A connector down room can be spawn in only MID and TOP row.

The room generation follow the guide below:

1. Randomly place a start room in the top row.

2. Every time a room is placed, at first it is always a PR.

3. The algorithm will look for where to place the next room: Right (40%), Left (40%), Down (20%).

4. If the algorithm decides to place next room Down, check whether that room is outside of the map boundary. If yes, change the current room as an Exit Room. If no, check whether the current room is CUP. If yes, change the current room as a Crossroad Room (with both up exit and bottom exit). If no, change the current to CDR, then place next room Down as CUP.

5. If the algorithm decides to place next room either Right or Left, first, check whether that room is outside of the map boundary. If yes, place the next room Down instead and follow the same placement rule of room type above. If no, place either a CUP (50%) or PR (50%) in that direction.

6. After the entire solution path generated, fill the empty grid with SR. Check whether the top, down, right, left already has an exit. If yes and has multiple exits pointing to it, randomly select an exit to connect the SR room with the template with the correct exit orientation; not multiple, just connect the SR room with the template with the correct exit orientation. If no, check the adjacent grids that have an existing room. Randomly pick one target room to connect the SR with.

Change the target room type to PR if the target room is to either the right or left of the SR.

Change the target room type to CUP if the target room is below the SR.

Change the target room type to CDR if the target room is above the SR.

If the target room is already either a SR, CUP, or CDR change the target room to Crossroad Room.

Connect the SR template with the correct exit orientation to the target room.

(*Room Template* see additional text document)

**Art**

**UI / Game Control**

Control Scheme

****

Gyroscope for Character Movement

Dodge

Attack

Action

Use Item

**Audio**

**Development Plan**