

Unreal Game Jam: April 2016

Version: 1.0.1

Updated: All the time

[Idea Pitches](#)

[Overview](#)

[Story](#)

[Platform](#)

[Development Challenges](#)

[Project Specific Websites](#)

[Flow of the game](#)

[How to play](#)

[Controls](#)

[Gameplay Mechanics](#)

[UI Layout](#)

[Level Design](#)

[Entity Descriptions and Behaviors](#)

[Player Feedback](#)

[Assets](#)

[Refs/Inspiration](#)

[Sound design](#)

Idea Pitches:

Idea 01:

Futuristic garden.(Robotic plants)

You have a garden that you are trying to grow of robot plants during the spring (Roses/flowers etc)

you are trying to help you garden grow by allocating power to your garden for a certain period of time. However while doing so weeds grow into your garden, you must allocate resources to help prevent the garden from being over run with "weeds"

Weeds can be real life nature objects...or can be a virus that infects your garden and makes your plants grow uncontrollably. The idea is to allocate resources of your garden to fully grow in a period of time and have it flourish past a certain percent.

Example Garden must grow in 3 days, to 60% of the area you have.

We can have fun making cool robotic plants and taking natural things and turning them robotic. We can have buffs and boosts that are "programs" and "energy" you allocate to grow your garden further.

Idea 02:

you must reach back to your time period of 20XX while springing through time. The idea is you are on a futuristic version of moon shoes and must Spring forward to your proper time. Each Step you make changes the color of the moon shoes and you must "step" on the right color/timing to spring ahead to the proper timeline.

so example Ever 5 steps your shoes change a color, you are given a time period where you must go too that is shown at the start of the game.

In example Your Time period you must return too Nov 12, 2088

November is shown in Red Text

12 is yellow Text

2088 is purple text

So you must race while keeping an eye on the time and spring to your right time and jump to the next part of the time and spring to your future. Every time you make a jump, the time gets shorter and shorter. If you make a wrong jump (wrong color) you go back in time (game over?)

Idea 03:

You control a robotic rabbit hopping from platform to platform in a over the shoulder view in a 3D setting. As you jump from platform to platform you control the rabbits power and angle as it jumps. The goal is to reach the end of the stage black hole (future) or end of stage. The stages will spawn random platforms that you can land on where you have physics take over as soon as you land on them.

Overview:

Grabbity- (Spring into the future!)

In this game you control a ~~robotic~~ rabbit hopping from platform to platform in an over the shoulder view in a 3D setting. As you jump from platform to platform you control the rabbits power and angle as it jumps. The goal is to reach the end of the stage black hole. The stages will spawn random platforms that you can land on where you have physics take over as soon as you land on them as well as objects that collide or hinder the players progressions.

Grabbity!

Enjoy your cheerful journey through a floating platform world as the happy robot rabbit called Grabbit, who is trying to find the path back to his mother egg.

In this game you control a rabbit hopping from platform to platform in an over the shoulder view. Aim carefully and charge your jump, then release the button to take a leap of faith! Can you make it to the exit teleporter in one go?

Controls explained in game.



Story:

You are a time traveling Bunny from the future who is stuck in a world crumbling apart around him you have limited time to get back home! Only through the Time Hole can you breach back to the future! Spring towards your future!

Platform:

- PC
- Consoles

Development Challenges:

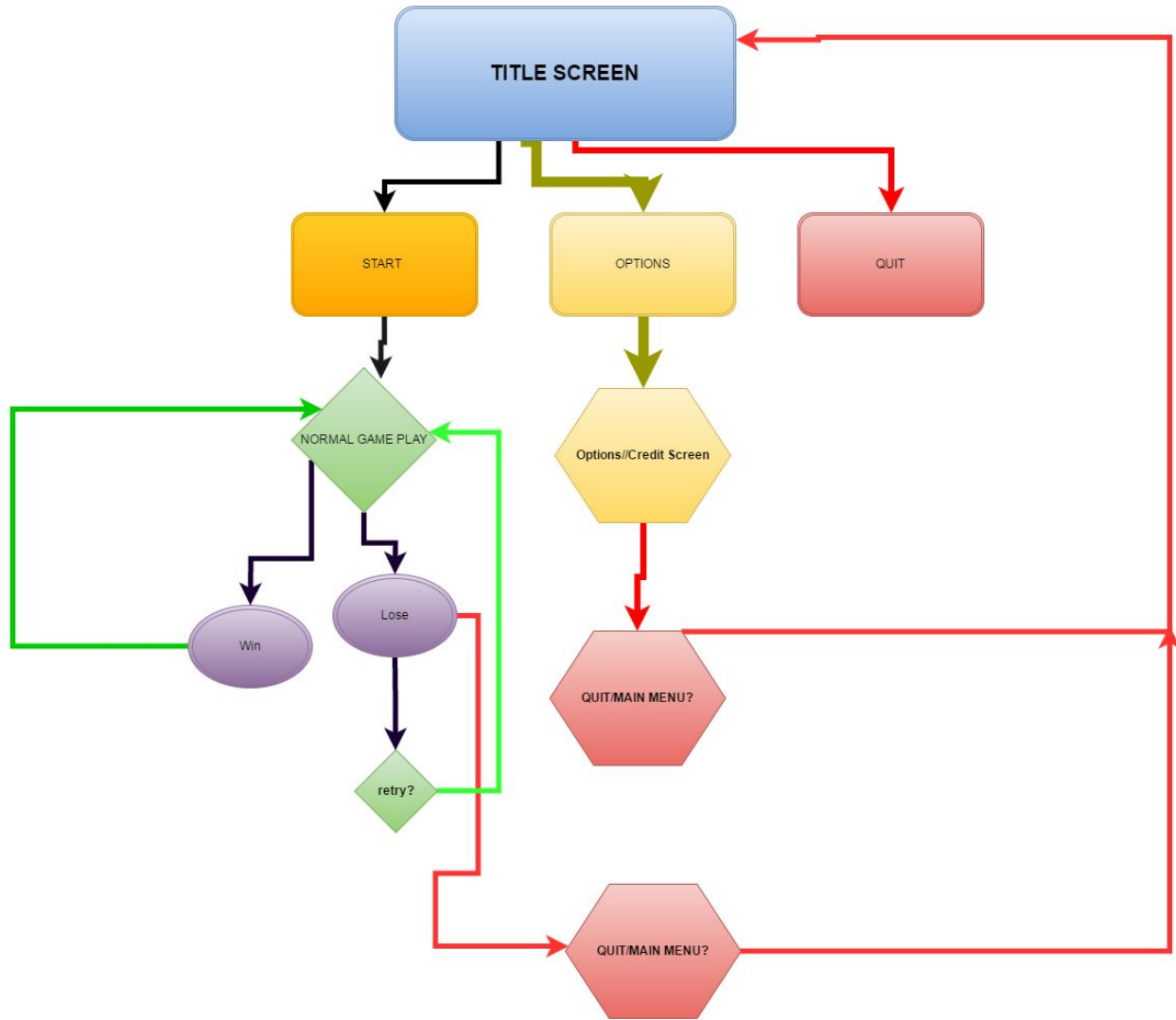
- Physics
- Time
- Aesthetics
- Blood pressure

Project Specific Websites:

- <https://trello.com/b/IH25qyhR>
- https://plus.google.com/hangouts/_/event/gamejamforrealsyo

Flow of the game:

<https://www.draw.io/>



How to play:

You control the angle of your rabbit (360 degrees) as you aim towards a platform and then charge with your power meter to go towards that direction.

Controls:

Xbox controller analog to aim, right or left trigger to charge your power.

Game Mechanics & Entity Behaviors:

-Player Controls

Power Jumping

- You aim in a direction you wish by rotating a object around a player to indicate the direction it will launch too (360 degrees)
- Once you aim in a direction you charge your Power meter to indicate how much force you gain before jumping in that direction.

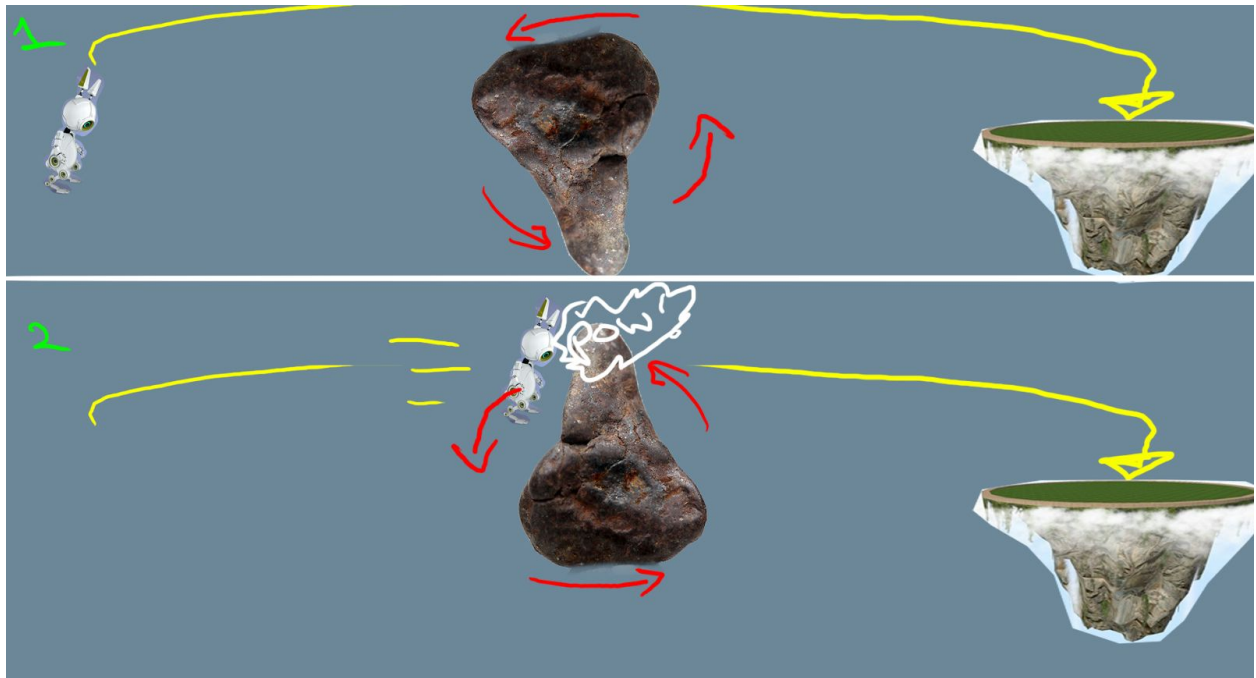
-Obstacles

Flying Carrots

- These objects move across screen in a direction causing obstruction to the player
- Objects will respawn across screen constantly

Rotating Objects

- Large platforms that rotate that can block an area until they rotate over once more (think of a wall that obstructs until it rotates completely over).



-Platforms

Solid Base

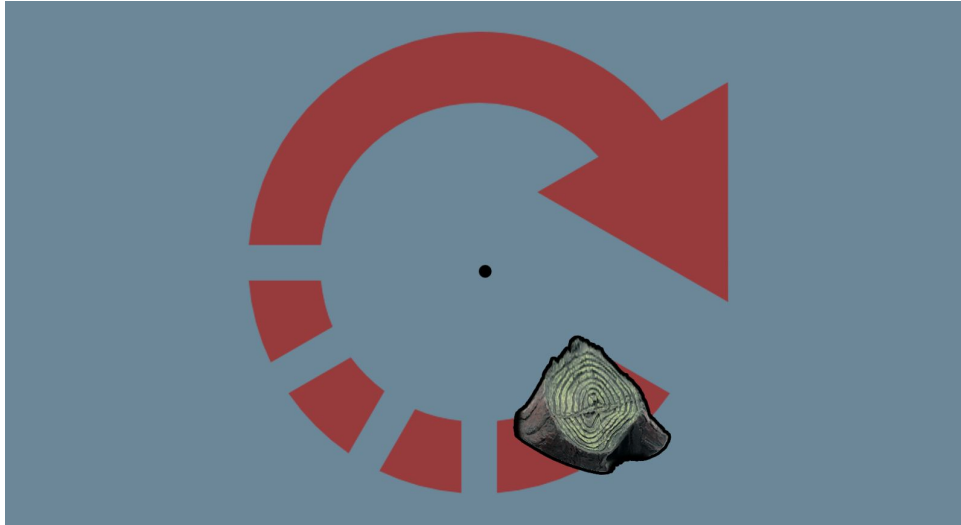
- These platforms stay in place and do not move

Crumble Base

- These platforms stay in place but if you do not immediately move to another platform and jump away this platform crumbles away and the player will fall.

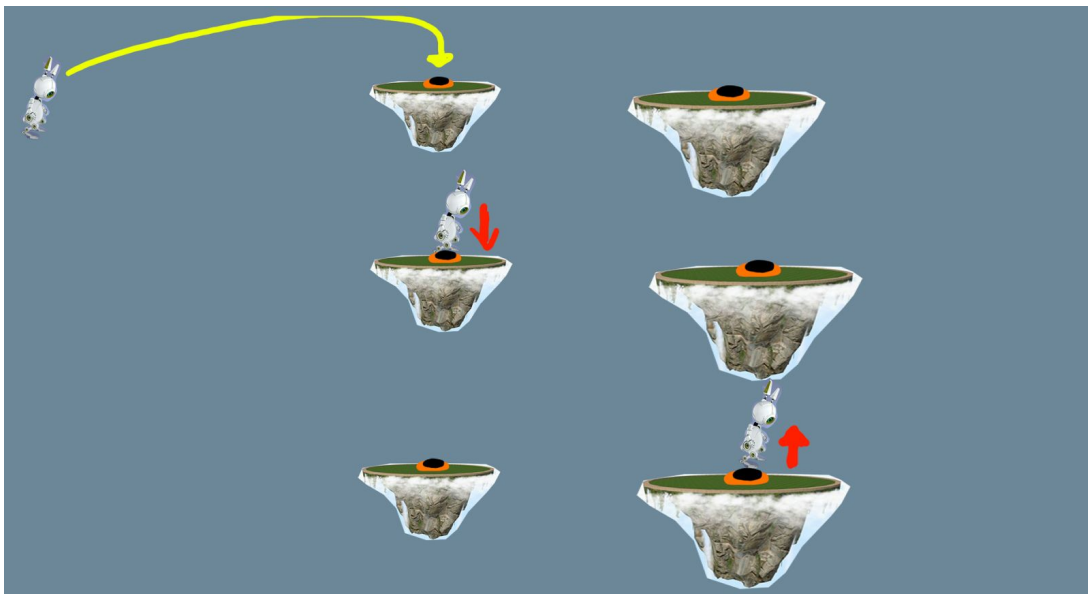
Rotating Base

- These platforms rotate in circles around a central location



Holes Base

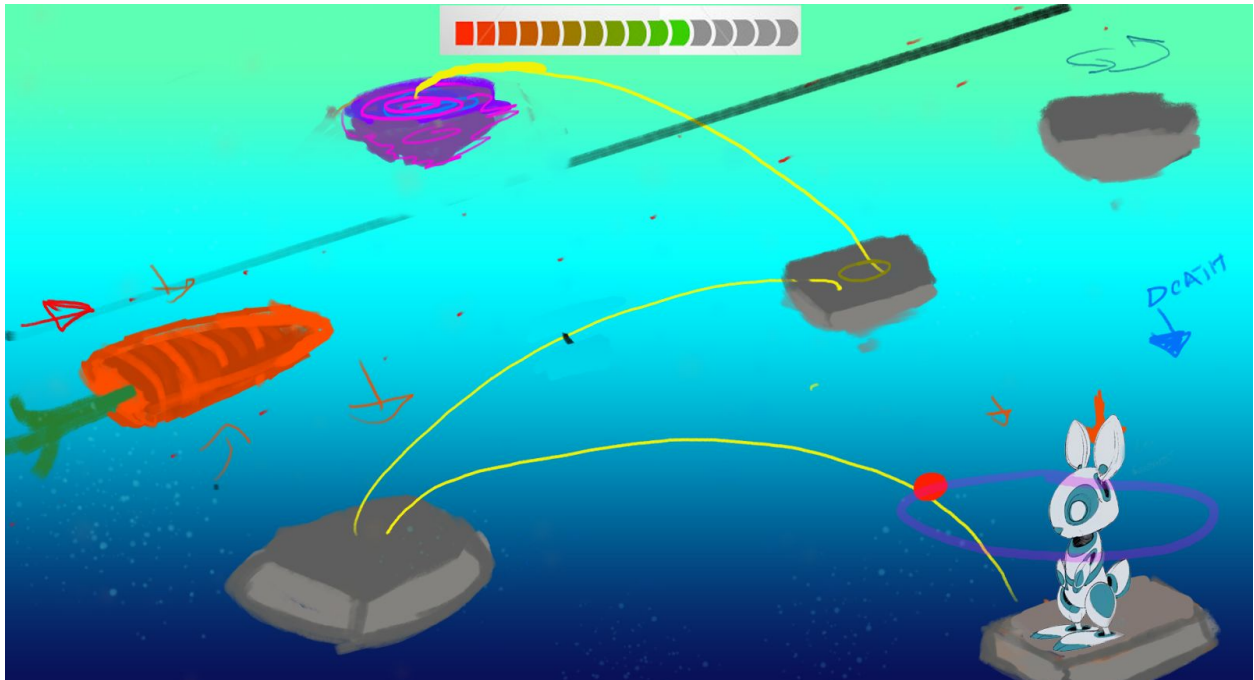
- These platforms stay in place but have rabbit holes that can end up in another location once you launch into them of the same rabbit hole color.



UI Layout:

Trajectory and Power

(purple and red orb represents how the orb can move 360 around the player, the red orb represents the direction the player will jump too)



Level Design:

Option A:

Premade platforms for level but object types and behavior is different for each one (Randomly spawns the type of platform it is, as well as obstacles in the level)

Option B:

Levels are purely dynamic with procedural layouts and platform spawning.

Player Feedback:

- Camera Shake
- RagDoll animation upon hit/collapse
- Controller Shaking?
- Motion blur/VFX lines on fast jumps or descents
-

Assets:

-Naming Conventions

EXAMPLE:

_MAT = Material
_T = Texture
_D = Diffuse
_RGB = Roughness/metalic/gloss/alpha/ao/cavity 4 in 1 texture
_N = normal
_Mesh = Mesh/fbx file

-2D Art



-3D Art

-HUD Art

-Animations

-Audio

- Music
- Ambient
- Sound Effects

References/Inspiration:

https://www.youtube.com/watch?v=hsMG829g_u0





