การศึกษาพัฒนาเกมจาก Open-source engine

โดย

นาย จิรพนธ์ กันภัย

นาย ปฐากร สุขแสง

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยธรรมศาสตร์

ปีการศึกษา 2565

การศึกษาพัฒนาเกมจาก Open-source engine

โดย

นาย จิรพนธ์ กันภัย

นาย ปฐากร สุขแสง

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยธรรมศาสตร์

ปีการศึกษา 2565

Game development from Open-source engine

By

Mr. Jirapon Kanpai

Mr. Pathakorn Suksaeng

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF THE BACHELOR OF ENGINEERING

IN COMPUTER ENGINEERING

FACULTY OF ENGINEERING

ACADEMIC YEAR 2022

ชื่อโครงงาน การศึกษาพัฒนาเกมจาก Open-source engine

ชื่อโครงงาน Game development from Open-source engine

โดย 1. นายจิรพนธ์ กันภัย

2. นาย ปฐากร สุขแสง

ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยธรรมศาสตร์

อนุมัติให้โครงงานนี้เป็นส่วนหนึ่ง ของการศึกษาตามหลักสูตร วิศวกรรมศาสตรบัณฑิต

.....อาจารย์ที่ปรึกษาโครงงาน

( )

.....หัวหน้าภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

(

หัวข้อโครงงาน	การศึกษาพัฒนาเกมจาก Open-source engine
ชื่อผู้เขียน	นายจิรพนธ์ กันภัย
	นายปฐากร สุขแสง
ชื่อปริญญา	วิศวกรรมศาสตรบัณฑิต
สาขาวิชา/คณะ/มหาวิยาลัย	วิศวกรรมคอมพิวเตอร์
	คณะวิศวกรรมศาสตร์
	มหาวิทยาลัยธรรมศาสตร์
อาจารย์ที่ปรึกษาโครงงาน	อาจารย์ ชุมพล บุญมี
ปีการศึกษา	2565

บทคัดย่อ

โครงงานการศึกษาพัฒนาเกมจาก Open-source engine เป็นการริเริ่มออกแบบและสร้างเกม เพื่อนำไปเป็นต้นแบบของเกมที่ออกแบบโดย มหาวิทยาลัยธรรมศาสตร์ และนำไปใช้ให้เป็นประโยชน์หรือ เพิ่มพูนรูปแบบการสอนที่มาจากการออกแบบเกมใน คณะวิศวกรรมศาสตร์ สาขาไฟฟ้าและคอมพิวเตอร์

โครงงานการศึกษาพัฒนาเกมจาก Open-source engine ได้มีการพัฒนา Document ทั้งหมดในการสร้างตัวเกม Prototype ขึ้นมา และตัวเกม Prototype พร้อมใช้งานเพื่อนำไปเป็นต้นแบบการริเริ่มออกแบบเกม เพื่อนำไปเป็นตัวอย่างในการศึกษาในคณะวิศวกรรมศาสตร์ สาขาคอมพิวเตอร์ต่อไป

คำสำคัญ: Open-source engine, Game Engine, Game Design

Title	Game development from Open-source engine
Author	Mr. Jirapon Kanpai
	Mr. Pathakorn Suksaeng
Degree	Bachelor of Engineering
Major Field/Faculty/University	Computer of Engineering
	Faculty of Engineering
	Thammasat University
advisor	อาจารย์ ชุมพล บุญมี
Academic Year	2022

ABSTRACT

Open-source engine game development education project is an initiative to design and build games. To be used as a prototype of a game designed by Thammasat University and put to good use and enhance the teaching style derived from game design in the Faculty of Engineering electrical and computer branch

An open-source engine game development study project has developed all the documentation on creating a game prototype, and the game prototype is ready to be used as a prototype for game design initiatives. to be used as an example for education in the Faculty of Engineering's computer branch

keyword: Open-source engine, Game Engine, Game Design

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงไปได้ด้วยความกรุณาของอาจารย์ ชุมพล บุญมี อาจารย์ที่ปรึกษาโครงงานที่ได้ให้คำแนะนำและปรึกษา ตลอดจนช่วยเหลือในการแก้ไขปัญหาต่างๆจนโครงงานนี้สำเร็จลุล่วง

ขอขอบพระคุณ อาจารย์ และเพื่อนนักศึกษา ตลอดจนผู้เกี่ยวข้องทุกท่านที่ได้มีส่วนร่วมให้โครงงานชิ้นนี่เสร็จสิ้นไปได้ด้วยดี

คณะผู้จัดทำหวังเป็นอย่างยิ่งว่าโครงงานนี้จะเป็นประโยชน์สำหรับผู้อ่านทุกท่าน

จิรพนธ์ กันภัย

ปฐากร สุขแสง

2566

สารบัญ

สารบัญรูปภาพ

บทที่ 2

บทที่ 3

บทที่ 4

สารบัญตาราง

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

ปัจจุบันได้มีการผลิตเกมออกมาจำนวนมากในโลกออนไลน์ ซึ่งหลังจากที่ผู้จัดทำได้ทดลองเล่นเกมไปเป็นจำนวนมาก ผู้จัดทำเลยได้อยากริเริ่มทำโครงงานสร้างเกม โดยใช้ Open-Source Engine ในการพัฒนาเพื่อนำไปเป็นต้นแบบของเกมที่ออกแบบโดย มหาวิทยาลัยธรรมศาสตร์ และเพื่อนำไปใช้ให้เป็นประโยชน์หรือเพิ่มพูนรูปแบบการสอนที่มาจากการออกแบบเกมใน คณะวิศวกรรมศาสตร์ สาขาไฟฟ้าและคอมพิวเตอร์ โดยมีเป้าหมายของผู้จัดทำคือ การพัฒนาเกมต้นแบบ และนำความรู้จากการที่ได้ออกแบบและพัฒนาเกม เพื่อเพิ่มพูนความรู้อีกแขนงหนึ่งให้กับ คณะวิศวกรรมศาสตร์ โดยสิ่งที่จะจัดทำขึ้นในโครงงานได้แก่

1. Document ทั้งหมดในการสร้างตัวเกม Prototype ขึ้นมา
2. ตัวเกม Prototype พร้อมใช้งานเพื่อนำไปเป็นต้นแบบการริเริ่มออกแบบเกม

1.2 วัตถุประสงค์

เพื่อที่ผู้จัดทำจะได้ศึกษาองค์ความรู้เพิ่มเติมเกี่ยวกับการสร้างเกมคอมพิวเตอร์ด้วย open source engine ศึกษาภาษาคอมพิวเตอร์ที่เกี่ยวข้อง ลองใช้งาน Library ต่างๆประยุกต์เพื่อทำให้ element ต่างๆไม่ว่าจะเป็นทั้งด้าน Visual Graphic, Game Play, Sound มารวมกันเพื่อให้ตัวเกมสามารถเล่นได้อย่างราบรื่นมีมาตรฐาน และ มอบประสบการณ์สนุกๆแก่ผู้เล่น

1.3 ขอบเขตการดำเนินงาน

การศึกษาและสร้างเกมจาก engine Godot และจัดทำการสร้างเกมออกมาเป็น Prototype ให้สมบูรณ์

1.4 ขั้นตอนการดำเนินงาน

1. เริ่มศึกษาเกี่ยวกับตัวของ Engine ที่ต้องใช้ทำ
2. InterActive Design
3. Graphic Design
4. Game Play Design
5. Story/Artwork
6. Inteface/Level Design
7. Audio

1.5 ผลที่คาดว่าจะได้รับ

- ประสบการณ์ในการทำงานในด้าน Game-dev
- เกมพร้อมใช้งาน
- นำความรู้ไปต่อยอดได้
- เพิ่มทางเลือกสายงานในอนาคต

1.6 ตารางการดำเนินงาน

ตาราง 1.1:ตารางการดำเนินงาน

24 ส.ค. 65	พูดคุยถึงเรื่องหัวข้อโครงงาน
29 ส.ค. 65	จัดทำข้อเสนอ
31 ส.ค. 65	นำส่งข้อเสนอโครงงาน
ก.ย. 65	เริ่มต้นศึกษาค้นคว้าการใช้งาน/รายงานการค้นคว้าเบื้องต้น
ต.ค. 65	ศึกษาค้นคว้า/จัดทำความคืบหน้าโครงงานครั้งที่ 1
พ.ย. 65	อับเดตความคืบหน้า/ดำเนินการจัดทำโครงงาน
ธ.ค. 65	อับเดตความคืบหน้า/ดำเนินการจัดทำโครงงาน
ม.ค. 66	อับเดตความคืบหน้า/จัดทำความคืบหน้าโครงงานครั้งที่ 2
ก.พ. 66	อับเดตความคืบหน้า/ดำเนินการจัดทำโครงงาน
มี.ย. 66	ดำเนินการจัดทำโครงงานให้สำเร็จ
เม.ย. 66	ตรวจสอบความเรียบร้อย/จัดทำปริญญานิพนธ์
พ.ค. 66	สอบโครงงานครั้งที่ 2 และส่งปริญญานิพนธ์

บทที่ 2

ทฤษฎีหรืองานที่เกี่ยวข้อง

2.1 Game Engine

Game Engine คือ Software ที่ถูกออกแบบและพัฒนาขึ้นมาเพื่อใช้ในการผลิตผลงานเกม เป็นโปรแกรมที่จะทำให้การพัฒนาเกมเกิดความสะดวกมากขึ้นในขั้นตอนของการผลิต โดยมักจะรวบรวมเอาชุดคำสั่งหรือเครื่องมือที่จำเป็นมารวมกันไว้เพื่อให้ง่ายต่อการหยิบยกเอามาใช้งานในการผลิตผลงานเกมโดย Core หลักๆ ที่เกม Engine มักจะมีใส่มาให้ก็เช่นการ Rendering ไม่ว่าจะเป็นในรูปแบบ 2มิติ หรือ 3มิติ , physics engine หรือระบบจัดการค่าทาง physics ต่างๆ ภายในเกม ไม่ว่าจะเป็นการคำนวณค่าแรงโน้มถ่วงของวัตถุ การชนกันของวัตถุ และ หลักการทางฟิสิกส์ อื่นๆ collision detection คือ ระบบการสร้างพื้นผิวสัมผัสของวัตถุ ของกำแพง รวมถึงระบบคำนวณสิ่งที่จะเกิดขึ้นเมื่อชนกับกำแพงต่างๆ , นอกจากนั้นยังมีระบบ sound, scripting, animation, artificial intelligence, networking, streaming, memory management, threading, localization support, scene graph และ อื่นๆ อีกหลายระบบที่มักจะถูกจัดเตรียมมาไว้ใน Game Engine เพื่อให้ผู้ใช้สามารถนำมาใช้งานได้เลยโดยที่ไม่ต้องสร้างขึ้นมาใหม่ซึ่ง Game Engine ของแต่ละเจ้าก็จะมีความสามารถและการใช้งานที่แตกต่างกันไปตามวัตถุประสงค์ของการใช้ [1]

รูปที่ 2.1 Game Engine

แหล่งที่มา: https://www.beartai.com/tgs/article-game/209630

2.2 Unity 3D

รูปที่ 2.2 Unity 3D

แหล่งที่มา: https://en.wikipedia.org/wiki/Unity_%28game_engine%29

เป็นโปรแกรมที่มีความหลากหลายสามารถสร้างเกม 2 มิติ 3 มิติ AR VR สามารถใช้ได้ในเกือบทุกๆ แพลตฟอร์ม Windows, IOS และ Android

ข้อดี - ฟรีมีตัวอย่าง source code ต่างๆมากมายทั้ง official และ community

ข้อเสีย - ใช้เนื้อที่ในการจัดเก็บไฟล์เยอะ และมีการรันไฟล์ที่นาน ไม่ได้ฟรีหมดทุก option

ข้อเปรียบเทียบระหว่างแบบเสียเงินและแบบไม่เสียเงิน [2]

2.2.1 บุคคลทั่วไป (ฟรี) สำหรับผู้มีรายได้หรือเงินทุนน้อยกว่า 1 แสนดอลลาร์ (ประมาณ 3 ล้านบาท) ในช่วง 12 เดือนที่ผ่านมา ฟังก์ชันที่สามารถใช้งานได้คือ

- ใช้งาน Unity เวอร์ชั่นล่าสุด
- ชุดเครื่องมือทั่วไป
- สามารถเข้าถึงแหล่งเรียนรู้เกี่ยวกับ Unity สำหรับผู้เริ่มต้นใช้งาน

2.2.2 Plus (เสียค่าใช้จ่ายประมาณ 1,267 บาท/เดือน/คน หรือ 12,639 บาท/ปี/คน) เหมาะสำหรับผู้ที่มีรายได้หรือเงินทุนน้อยกว่า 2 แสนดอลลาร์ (ราว 6 ล้านบาท) ในช่วง 12 เดือนที่ผ่านมา

- ใช้งาน Unity เวอร์ชั่นล่าสุด

- ใช้งานหน้า UI ในโหมด Dark

- ปรับแต่ง Splash Screen

- การวิเคราะห์ความเสถียรของเกม

- ทำงานบนระบบคลาวด์แบบเรียลไทม์

2.2.3 Pro (เสียค่าใช้จ่าย ประมาณ 4,751 บาท/เดือน/คน และ 56,997 บาท/ปี/คน) เหมาะสำหรับผู้ที่มีรายได้หรือเงินทุนมากกว่า 2 แสนดอลลาร์ (ราว 6 ล้านบาท) ในช่วง 12 เดือนที่ผ่านมา และทีมนักพัฒนาตั้งแต่ 3 คนขึ้นไป

- ทำทุกอย่างได้เหมือน Plus

- ทำงานร่วมกันได้ง่ายขึ้น

- สอบถามพูดคุยกับผู้เชียวชาญของ Unity

- กำหนดตัวเลือกหรือบริการเอง

- การสนับสนุนทางเทคนิค

- เข้าถึงซอสโค้ด (Source Code)

2.2.4 Enterprise (เสียค่าใช้จ่ายประมาณ 6,333 บาท/เดือน/คน) เหมาะสำหรับองค์กรที่มีขนาดใหญ่ที่มีทีมนักพัฒนา 20 คนขึ้นไป และผู้ที่มีรายได้หรือเงินทุนมากกว่า 2 แสนดอลลาร์ (ราว 6 ล้านบาท) ในช่วง 12 เดือนที่ผ่านมา

- ทำทุกอย่างได้เหมือน Pro

- ได้รับการสนับสนุนทางเทคนิค

- พูดคุยปรึกษากับผู้เชี่ยวชาญเพื่อพัฒนา Skill โดยรวม

- สามารถเรียนรู้ และประชุมแบบ Live ได้

- สร้าง AR และ VR ไว้ใช้ในการออกแบบด้านวิศวกรรม และสถาปัตยกรรม

- สร้างอนิเมชัน 3 มิติ (3D Animation) และภาพยนตร์

2.3 UNREAL ENGINE

รูปที่ 2.3 UNREAL ENGINE

แหล่งที่มา: https://www.pcgamesn.com/unreal-engine-5-demo

พัฒนาโดย Epic Game เป็นเกม Engine ที่ประสบความสำเร็จสูงสุดตัวหนึ่ง โดย engine ตัวนี้ใช้ภาษา C++ ใช้ได้กับทุกแพลตฟอร์ม และเป็น engine ของเกมส่วนใหญ่

ข้อดี - ใช้งานได้ฟรี เหมาะสำหรับทำเกม 3 มิติ แล้ว มีกราฟฟิกสวยงาม มีฟีเจอร์มต่างๆ พร้อมให้ใช้โดยไม่ต้องลง package

ข้อเสีย - มี learning curved ที่สูง user interface ไม่เหมาะกับมือใหม่ จำเป็นต้องใช้ฮาร์ดแวร์ที่มีประสิทธิภาพสูง อาจจะต้องเสียรายได้บางส่วนให้กับ Unreal engine

2.4 AMAZON LUMBERYARD

รูปที่ 2.4 Amazon lumberyard

แหล่งที่มา: https://en.wikipedia.org/wiki/Amazon_Lumberyard

เป็นโปรแกรมสร้างเกมแบบโอเพนซอร์สข้ามแพลตฟอร์มที่มีความสามารถระดับ AAA และพร้อมใช้งานภายใต้ใบอนุญาต Apache 2.0

ข้อดี - เป็น engine ที่ฟรี , ใช้งานกับ Amazon cloud services ได้ดี , กราฟฟิกที่สูง

ข้อเสีย – เนื่องจากยังใหม่และยังไม่ได้รับความนิยมเท่าที่ควรทำให้การเข้าถึงค่อนข้างยาก

และในปัจจุบัน Open 3D Foundation และ Open 3D Engine (O3DE) ซึ่งเป็นโปรแกรมสร้างเกมแบบโอเพนซอร์สข้ามแพลตฟอร์มที่มีความสามารถระดับ AAA และพร้อมใช้งานภายใต้ใบอนุญาต Apache 2.0 เราต้องการให้นักพัฒนาเกมและการจำลองมีทางเลือกมากขึ้นในการทำงานร่วมกัน ปรับแต่ง และควบคุมไปป์ไลน์การผลิตของตน และเรายังขยายชุมชนโอเพนซอร์สด้วย Linux Foundation และคู่ค้าในอุตสาหกรรมอีกด้วย O3DE ซึ่งมาแทน Lumberyard [3]

2.5 Godot

รูปที่ 2.5 Godot engine

แหล่งที่มา: https://en.wikipedia.org/wiki/Godot_(game_engine)

เป็นเกมเอนจินแบบข้ามแพลตฟอร์มสำหรับใช้ในการพัฒนาวิดีโอเกมทั้งแบบ 2 มิติ และ 3 มิติ Godot engine ถูกเผยแพร่ออกมาสู่สาธารณะในรูปแบบที่ฟรีไม่มีค่าใช้จ่ายและเปิดเผยซอร์ซโค้ด (Open-source) ภายใต้ MIT License ซึ่งไม่มีพันธะข้อผูกมัดหรือค่าสิทธิ (Royalty) ในการใช้งานแต่อย่างใด ผู้ใช้งานจะเป็นเจ้าของเกมส์ที่ตนเองสร้างตลอดจนโค้ดทั้งหมดของตัวเอนจินโดยสมบูรณ์ โกโดถูกพัฒนาอย่างอิสระเนื่องด้วยความเป็น Open-source โดยความร่วมมือของชุมชน (Community) ผู้ซึ่งมีความสนใจที่จะเป็นส่วนหนึ่งในการพัฒนาขับเคลื่อนตัวเอนจินให้มีประสิทธิภาพดียิ่ง ๆ ขึ้นไป ทำให้ Godot ถูกพัฒนาไปในแนวทางที่ตอบโจทย์ความต้องการของผู้ใช้งานเนื่องจากชุมชนสาธารณะนี้อาจหมายรวมถึงผู้ใช้งานจริงร่วมด้วย นอกจากนี้โกโดยังได้รับการสนับสนุนจาก Software Freedom Conservancy ซึ่งเป็นองค์กรไม่แสวงหาผลกำไรอีกด้วย [4]

โดยที่ภาษาหลักๆที่ใช้งานจะประกอบไปด้วย C#,C++, GDScript และ ยังมี ตัว Visual Scripting ของตัว engine เอง

ข้อดี - เป็น open source ที่ฟรีไม่มีค่าใช่จ่าย, ติดตั้งง่าย , ขนาดของไฟล์โปรแกรมที่เล็ก, ทำงานได้รวดเร็ว ตัว script language ยึดภาษา python เป็นหลักทำให้มีใช้งานได้หลากหลาย

ข้อเสีย - ตัว engine ไม่มี asset ที่สวยงามแถมมา ดั้งนั้นผู้ใช้จึงต้องหา model หรือสร้าง asset อื่นๆมาใช้งานเอง , ขาดฟีเจอร์สำหรับ open world และการจัดการสภาพแวดล้อม ไม่เหมาะสำหรับผู้ที่เริ่มต้นโดยไม่มีความรู้เกี่ยวกับการเขียนโค้ด

2.6 Game Design Document

รูปที่ 2.6 game design document

แหล่งที่มา: https://gamedevbeginner.com/

Game Design Document เป็นเอกสารที่บอกรายละเอียดการออกแบบด้านต่างๆ ของเกม จะทำหน้าที่เป็นแผนที่ให้ทุกคนในทีมได้ไปเจอกันที่เป้าหมายเดียวกัน เป็นเอกสารที่อธิบาย วิธีเล่นเกม ตัวละคร เนื้อเรื่อง แนวกราฟิก กลุ่มตลาดเป้าหมาย และข้อมูลทางเทคนิคสำหรับนักพัฒนา

Game Design Document จะมีหน้าในการทำให้ทุกคนในทีมมีความเข้าใจและมุมมองไปในทิศทางเดียวกัน เบื้องต้น Game Design Document จะเป็นเพียงโครงสร้างหลวมๆ ให้คนที่รับผิดชอบในแต่ล่ะด้านสามารถมาปรับแก้ได้ระหว่างพัฒนาเกม ดั้งนั้น Game Design Document จะไม่ใช่เอกสารตายตัว และในหลายๆกรณีจะไม่ใช่เอกสารที่เขียนด้วยคนคนเดียวแต่เป็นการรวมมือกันหลายๆฝ่าย [5]

2.7 Aseprite

รูปที่ 2.7 Aseprite

แหล่งที่มา: https://devahoy.com/blog/2020/04/setup-aseprite-on-mac-os

Aseprite เป็นโปรแกรมแบบพิกเซลอาร์ตที่ผู้ใช้สามารถใช้สร้างแอนิเมชั่น 2 มิติสำหรับวิดีโอเกม เป็นซอฟต์แวร์ที่ต้องชำระเงิน และผู้ใช้สามารถซื้อผลิตภัณฑ์ผ่านเว็บไซต์ทางการ อย่างไรก็ตาม ซอร์สโค้ดยังมีให้ใช้ฟรีอีกด้วย วิธีการติดตั้ง Aseprite บน Linux

โดยที่เราจะใช้ตัวโปรแกรมนี้ในการออกแบบพัฒนา sprite รูปแบบต่างๆให้มีความสวยงาม

บทที่ 3

วิธีการวิจัย

3.1 ศึกษาการใช้งาน Godot Engine

ผู้จัดทำได้เลือก Engine ที่จะใช้พัฒนาเกมนี้ซึ่งก็คือ Godot เพื่อให้ตรงตามจุดประสงค์สำคัญ ซึ่ง Godot นั้นเป็น open-source engine ที่มี UI ที่ใช้งานได้ง่าย โดยภาษาหลักๆที่ใช้งานประกอบไปด้วย C#,C++GDScript และตัว Visual Scripting ของตัว engine เอง

ข้อดี – เป็น open source ที่ฟรีไม่มีค่าใช่จ่าย, ติดตั้งง่าย , ขนาดของไฟล์โปรแกรมที่เล็ก, ทำงานได้รวดเร็ว ตัว script language ยึดภาษาคล้าย python เป็นหลักทำให้มีใช้งานได้หลากหลาย

ข้อเสีย – ตัว engine ไม่มี asset ที่สวยงามแถมมา ดั้งนั้นผู้ใช้จึงต้องหา model หรือสร้าง asset อื่นๆมาใช้งานเอง , ขาดฟีเจอร์สำหรับ open world และการจัดการสภาพแวดล้อม ไม่เหมาะสำหรับผู้ที่เริ่มต้นโดยไม่มีความรู้เกี่ยวกับการเขียนโค้ดต้องใช้พัฒนาเกม

3.1.1 Node องค์ประกอบเบื้องต้นที่นำมาสร้างเกม 2 มิติของ Godot

องค์ประกอบสำเร็จรูปต่างๆใน Godot ที่ผู้จัดทำได้นำมาใช้งานจะถูกเรียกว่า Node ซึ่งจะสามารถเพิ่ม Script ที่สามารถเขียน code ซึ่งเป็นภาษา GDScript เพื่อแก้ไขคุณสมบัติรวมไปถึง event ต่างๆที่จะเกิดขึ้นเกี่ยวกับ Node นั้นๆ ซึ่งการใช้งาน Node นั้นจะมีการสร้าง Node หลักมาเป็น scene ที่เราจะใช้แสดงผลแล้วจึงเพิ่ม child Node เชื่อมเข้ากับ Node หลักที่เป็น parent Node เพื่อให้ทำงานอยู่ใน scene ทางผู้จัดทำได้มีการใช้งาน Node ต่างๆดังนี้

3.1.1.1 Node2D

เป็น node ที่เป็นพื้นฐานของทุกๆ Node ที่นำมาใช้ในการสร้าง element เกี่ยวกับการสร้างเกมส์ 2 มิติ มักใช้เป็นตัว Node หลักของ scene เพื่อเชื่อม child Node อื่นๆ

3.1.1.2 TileMap

เป็น Node ที่ใช้ในการสร้างสภาพแวดล้อมต่างๆภายในฉากโดยสามารถนำภาพ asset ต่างๆมา import เข้าไปใน node เพื่อนำมาใช้สร้างพื้นดินสำหรับให้ตัวละครเดินหรือสิ่งกีดขวางภายในฉากที่สามารถตั้งให้ตัวละครผ่านไม่ได้ ผ่านการสร้าง collision

3.1.1.3 CharacterBody2D

เป็น Node ที่ถูกสร้างไว้สำหรับตัวละคร 2 มิติ โดยจะมี script สำเร็จรูปสำหรับจัดการเกี่ยวกับฟิสิกส์การเคลื่อนที่ต่างๆในแนว 2 มิติ

3.1.1.4 Sprite2D

เป็น Node ที่ใช้สำหรับแสดง Texture รูปภาพต่างๆโดยสามารถนำรูปภาพมาใส่โดยตรง หรือใช้งานร่วมกับ Node อื่นๆ เพื่อเปลี่ยนแปลงภาพตาม script ของ Node อื่นๆ

3.1.1.5 AnimationPlayer2D

เป็น Node ที่ใช้สำหรับจัดการการแสดงภาพเคลื่อนไหวต่าง โดยสามารถทำให้เป็น Frame ของภาพตามเวลา และสามารถสั่งใช้งาน method ใน GDScript ในแต่ละ Frame การแสดงผลของ AnimationPlayer2D จำเป็นต้องแสดงผลใน Node อื่นเช่น Spite2D

3.1.1.6 RayCast2D

เป็น Node ที่ใช้สำหรับตรวจจับ Object ที่มีการ collide กับตัว Node โดยจะมีทิศทางเป็นลูกศรออกมาจากตัว Node ที่เป็น parent Node

3.1.1.7 Area2D

เป็น Node ที่มักถูกใช้สำหรับเป็น parent Node สำหรับสร้างพื้นที่ ตัวอย่างการใช้งานคือใช้เป็น area เพื่อตรวจจับ หรือเพิ่ม code script ที่จะทำงานเมื่อมี object มา collide

3.1.1.8 CollisionShape2D

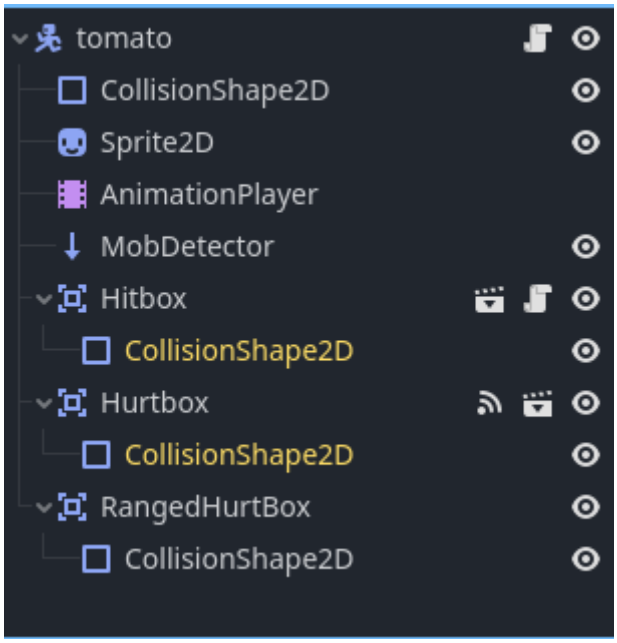
เป็น Node ที่เพิ่ม Collision Layer ให้กับ parent Node ของตัวเอง

3.1.1.9 TextureButton

เป็น Node ที่นำมาใช้ในการสร้างปุ่มสำหรับกด โดยสามารถใส่รูปภาพ ในแต่ละสถานะของปุ่มกดได้ สามารถเพิ่ม script เพื่อเพิ่มวัตถุประสงค์ในการทำงาน

3.1.1.10 ParallaxBackground และ ParallaxLayer

เป็น Node ที่ใช้สำหรับสร้างภาพพื้นหลังของฉาก สามารถปรับให้เคลื่อนไหวได้



รูปที่ 3.1 ตัวอย่าง Node Tree ของตัวละคร tomato

3.1.2 GDScript

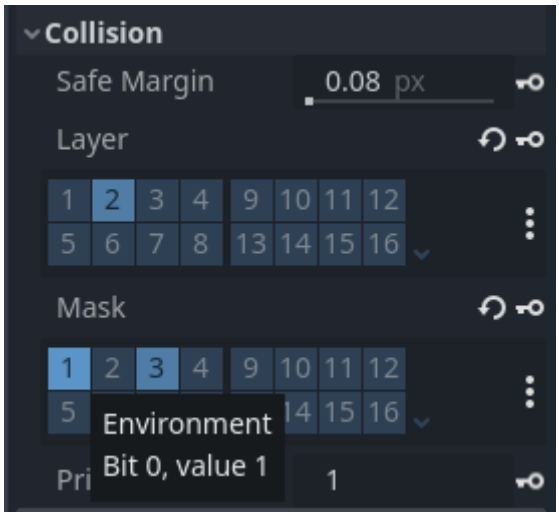
เป็นภาษาที่มีการใช้งานในการแก้ไขสถานะหรือสร้าง event ใน Node มี Syntax การใช้งานคล้ายคลึงกับภาษา python โดยจะมีการ extend มาจาก Node ที่เป็น parent

```
1   extends CharacterBody2D
2   @export var hp =50
3   @export var speed = 30.0
4   var direction = 1
5   const JUMP_VELOCITY = -400.0
6
7   # Get the gravity from the project settings to be synced with RigidBody nodes.
8   var gravity = ProjectSettings.get_setting("physics/2d/default_gravity")
9
10
11  func _physics_process(delta):
12      # Add the gravity.
13      if not is_on_floor():
14          velocity.y += gravity * delta
15
16      # Handle Jump.
17      if Input.is_action_just_pressed("ui_accept") and is_on_floor():
18          velocity.y = JUMP_VELOCITY
19
20      if $AnimationPlayer.current_animation == "attack":
21          return
22      detect_player()
23
24      # Get the input direction and handle the movement/deceleration.
25      # As good practice, you should replace UI actions with custom gameplay action
26      if direction:
27          velocity.x = direction * speed
28      else:
```

รูปที่ 3.2 ตัวอย่าง GDScript ของตัวละคร tomato

3.1.3 CollisionLayer และ CollisonMask

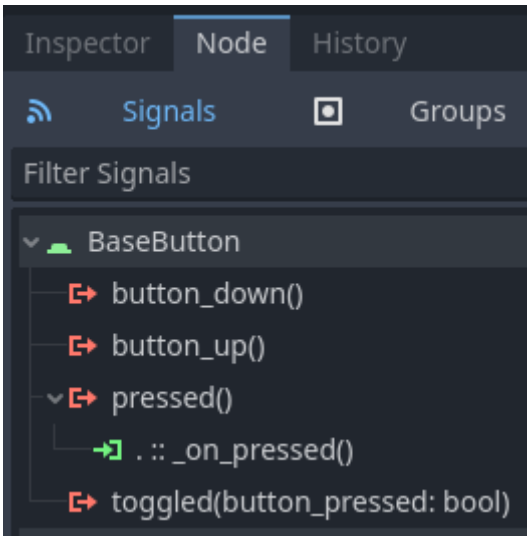
Collision เป็นสถานะที่ทำให้ object ต่างๆใน project สามารถชนกันได้ ซึ่งแต่ละ Node ที่ต้องการให้เกิดการชนกันจำเป็นที่จะต้องมีการตั้งค่า Collision ให้ถูกต้อง โดยจะมีแยกเป็น Layer และ Mask โดย Layer จะบอกสถานะที่ object นั้นเป็น และ Mask จะบอกสิ่งที่ object นั้นสามารถมองเห็นได้ เช่น เมื่อ tomato อยู่ layer ที่ 2 จะนับว่าเป็น layer 2 มี mask เป็น 1 แสดงว่าสามารถ มองเห็นหรือ ชนกับ layer 1 ได้



รูปที่ 3.3 ตัวอย่าง Collision ของ tomato

3.1.4 Signal

ทำหน้าที่ส่งการทำงานของ Node ที่ interactive ได้ไปเป็น code ใน GDscript



รูปที่ 3.4 ตัวอย่างของ Signal จาก Node Button

3.2 ศึกษาเกี่ยวกับ Game Design

ในปัจจุบันได้ดำเนินการในส่วนของการเขียน GDD หรือ Game Design Document และ ได้ออกแบบ UI หรือ User Interface เบื้องต้นสำหรับตัวเกม โดย Game Design Document คือขั้นตอนแรกในการพัฒนาเกมโดยต้องการให้เห็นถึงภาพรวม และสิ่งต่างๆที่จะเกิดขึ้นในเกมที่จะถูกพัฒนาขึ้นมา โดยจากการที่ศึกษาวิธีสร้างเกมตัวเอกสารตัวนี้จะเป็นส่วนสำคัญที่จะเป็นจุดเริ่มต้นของเกมที่กำลังจะถูกพัฒนาขึ้นมา

3.2.1 Game Design Document

Project Description

เอกสารสำหรับออกแบบเกมนี้แสดงถึงรายละเอียดของเกม ซึ่งจะเป็นเกมบน Platform PC โดยตัวเกมจะเป็นเกม 2D ที่มีรูปแบบการเล่นแบบ Real-time strategy tower defense แบบมุมมองด้านข้างผสมผสานกับรูปแบบ Rougelike ที่มีตัวละครกับเนื้อเรื่องเป็นของตัวเอง

Characters

Veggie caravan - คาราวานผักที่รักการผจญภัย

Supreme Dark Candy King - ราชาปีศาจลูกอมที่คิดจะยึดครองโลก

Candy legion - ปีศาจลูกอมลูกน้องของราชาปีศาจลูกอมจากต่างโลก

Story

การครั้งหนึ่งในโลกอันสงบสุขของเหล่าผักผลไม้ คุณได้รับบทกลุ่มผักนักผจญภัยที่กำลังใช้ชีวิตเป็นทหารรับจ้าง แต่ทว่ากลับมีประตูมิติจากดาวลูกกวาดชั่วร้ายผุดขึ้นมา ลักพาตัวสมาชิกผักและชาวบ้านผลไม้ เพื่อไปเป็นเชื้อเพลิงสำหรับอัญเชิญราชาปีศาจลูกอมมายึดครองโลกผักผลไม้

คุณจึงต้องไล่ล่าและช่วยเหลือเหล่าผัก รวบรวมกองกำลังของคุณเพื่อต่อต้านการรุกราน

Gameplay

Goals

Overall: เอาชนะศัตรู เพื่อปลดล็อคและพัฒนาตัวละคร และ ไปสู่ระดับถัดไปให้ได้มากที่สุด

User Skills

1. การวางแผน

2. การจัดการทรัพยากร

3. ปุ่มควบคุมการเคลื่อนไหวของตัวละคร Hero

3.1 ปุ่มลูกศรสำหรับเคลื่อนที่

3.2 ปุ่มสำหรับโจมตีเบา

3.3 ปุ่มสำหรับโจมตีหนัก

3.4 ปุ่มสำหรับสกิลพิเศษของตัวละคร

4. ปุ่มสำหรับการเรียกตัวละครมิเนี่ยนทั้ง 4 ตัว

Game Mechanics

เกมจะนำเสนอ การเล่นแบบที่เป็น tower defense ผสม ความเป็น Roguelike โดยที่

- ผู้เล่นจะได้เล่นเป็นรอบเมื่อผู้เล่นแพ้จะต้องเริ่มใหม่จากศูนย์เสมอ โดยในเกมจะปล่อยศัตรูออกมาเป็นรอบย่อยๆโดยเมื่อเล่นจบแต่ละรอบย่อยจะได้รับรางวัลเสมอ

- เมื่อเล่นจบทุกๆรอบย่อยผู้เล่นจะได้เลือกรับรางวัลเป็นทหารผักที่ถูกจับตัวไว้ 1 จาก 3 เพื่อมาเสริมให้ทีมของคุณแข็งแกร่งขึ้น โดยจะสุ่มประเภทและความหายากของรางวัล

- หากคุณได้รับรางวัลตัวละครซ้ำ ตัวละครตัวนั้นจะแข็งแกร่งขึ้นเช่นคุณเลือกทหารผักกาด โดยที่คุณมีทหารผักกาดอยู่แล้ว ทหารผักกาดของคุณจะกลายเป็น ทหารผักกาด lvl.2

- เมื่อเริ่มและทุกๆการเล่น Wave ที่ กำหนดคุณจะได้เลือก Relic พิเศษที่จะทำให้คุณสามารถเลือก Hero ได้ โดยหลังจาก Wave ที่ กำหนดจะเป็นการเพิ่มค่าสถานะให้กับ Hero

- หากคุณแพ้คุณต้องกลับไปเริ่มเล่นใหม่จากศูนย์

Item and power-ups

ไอเทมในเกมจะเป็น Artifact และ Unit ฝ่ายพันธมิตรต่างๆ ที่หาได้จากรอบการเล่นและสามารถนำมาอัพเกรดแล้วนำมาใช้ในรอบๆถัดๆไปได้

Unit หมายถึงตัวละครที่ผู้เล่นสามารถเรียกออกมาเพื่อป้องกันคาราวาน โดยจะแบ่งออกเป็น 3 คลาสแบ่งระดับ ตามระยะยืนและรูปแบบการทำงานของตัวละครโดยแบ่งออกเป็น 3 สี ดังนี้

1. Unit สีแดงจะเป็นประเภทโจมตีประชิดจะยืนอยู่ช่องทางด้านหน้าสุด

2. Unit สีเหลืองจะเป็นประเภทโจมตีกระยะกลางจะอยู่ที่ช่องตรงกลาง

3. Unit สีเขียวจะเป็นโจมตีระยะไกลและสนับสนุนจะอยู่ที่ช่องหลังสุด

Artifact หมายถึงอุปกรณ์ที่จะทำให้เราสามารถเรียกตัว Hero ได้โดยตัว Hero นั้นสามารถมีได้เพียงตัวเดียวบน Field และมีคูลดาวน์ในการเรียกที่นาน

Progression and challenge

ยิ่งเล่นไปไกลในรอบศัตรูก็จะยิ่งยากขึ้น โดยความที่เป็น roguelike ทำให้เกิดความหลากหลายในการเล่นต้องมีการลองผิดลองถูกในการเลือกยูนิทต่างๆ คุณอาจต้องเล่นหลายๆรอบเพื่อให้สามารถพิชิตด่านยากๆได้ หรือ คุณอาจจะสามารถเอาชนะได้ในครั้งเดียวหากคุณมีการวางแผนที่ดี

Losing

คุณจะแพ้ในรอบนั้นก็ต่อเมื่อ รถม้าของคุณพังทะลาย

Art style

โดยเกมนี้จะเป็น 2d pixel art side-scrolling โดย sprite 2d จะเป็น original character เน้นสีสันที่สดใสพร้อมกับพื้นหลังที่เน้นให้ดูมีชีวิตชีวา

Music and Sounds

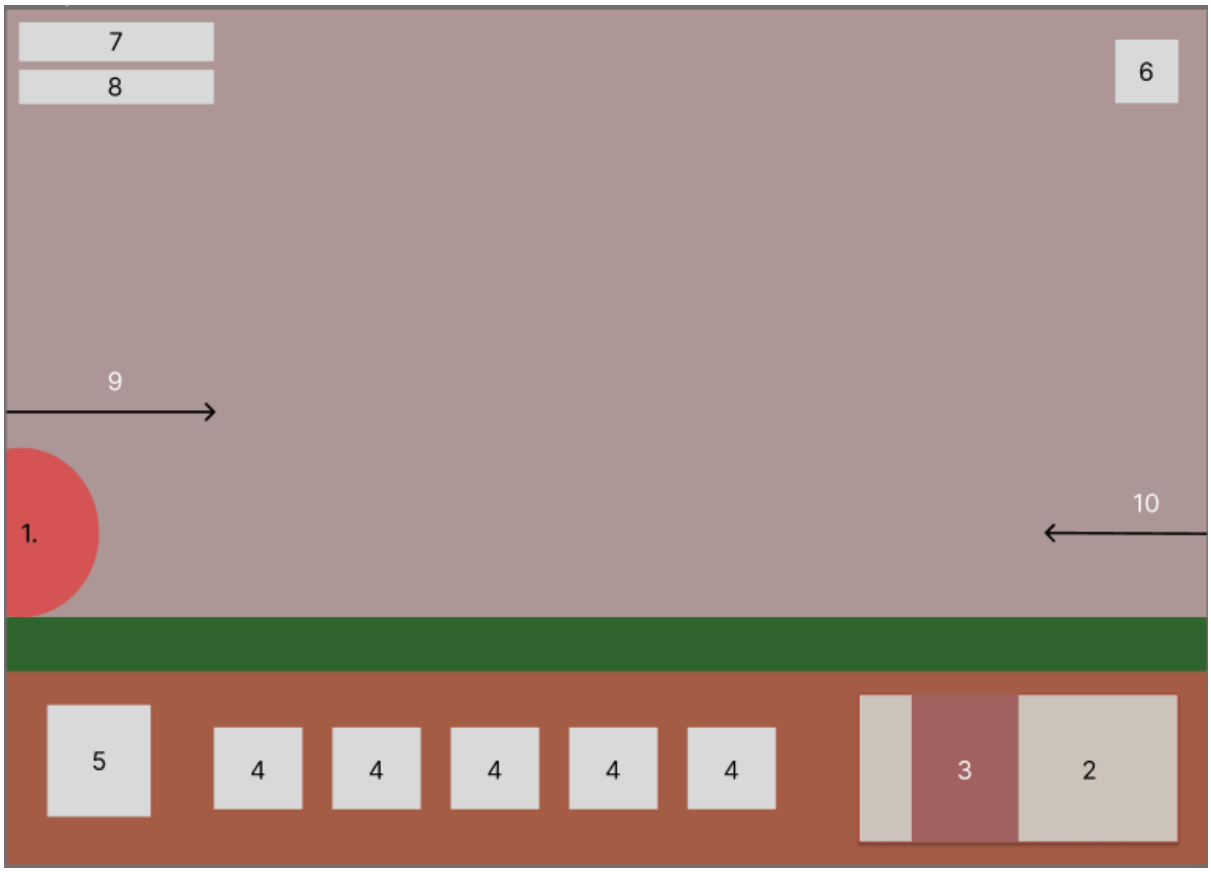
ส่วนของเพลงจะใช้เป็นเพลง 8-bit โดยเน้นไปที่ความเป็น fantasy และการผจญภัย โดยเพลงส่วนใหญ่จะใช้เป็นเพลงที่ให้ความรู้สึก Happy Relax โดยเฉพาะตอนที่ปราสาทใกล้พัง เราจะเร่งจังหวะเพลงให้ดูตื่นเต้นมาขึ้น

Technical description

เกมนี้มีแผนจะทำลง Window pc เท่านั้น

โดยใช้ engine Godot ซึ่งใช้ภาษา GDScript ซึ่งมี base เป็น Python

3.2.2 การออกแบบ User Interface



รูปที่ 3.5 ตัวอย่าง user interface

จากภาพจะเป็น User Interface เบื้องต้นของเกมเพลย์โดยมีรายละเอียดที่แสดงตามตัวเลขดังนี้

1.แสดงถึงตำแหน่งของ คาราวานที่เราต้องป้องกัน

2.แสดงถึงแผนที่ของด่านนั้นๆ

3.แสดงถึงตำแหน่งของแผนที่ที่กำลังแสดงอยู่ในหน้าจอหลัก

4.ปุ่มสำหรับเรียกใช้งาน Unit ทั้ง 5 ชนิด

5.ปุ่มสำหรับเรียกใช้งานตัวละคร Hero

6.ปุ่มสำหรับ Pause และแสดง Menu

7 และ 8.แสดงถึงหมายเลขประจำด่านนั้นๆ รวมไปถึงเวลาที่เหลืออยู่

9.แสดงถึงทิศทางที่ยูนิตฝ่ายเราจะเข้ามาป้องกัน

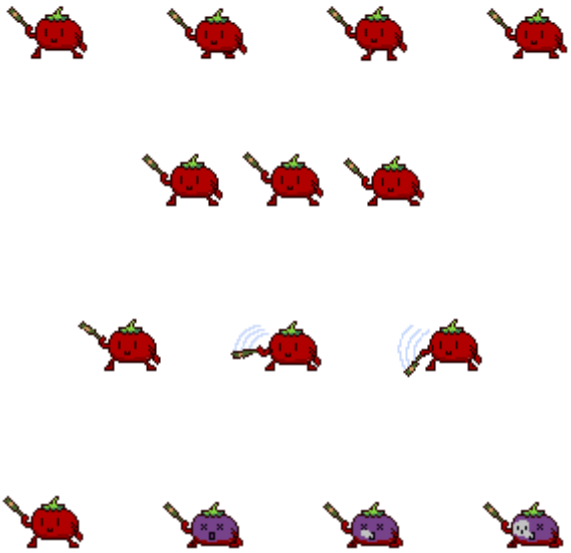
10.แสดงถึงทิศทางที่มอนสเตอร์ฝ่ายศัตรูจะเข้ามาโจมตี

บทที่ 4

ผลของการวิจัย

4.1 องค์ประกอบของเกม

Sprite



รูปที่ 4.1 Tomato Sprite

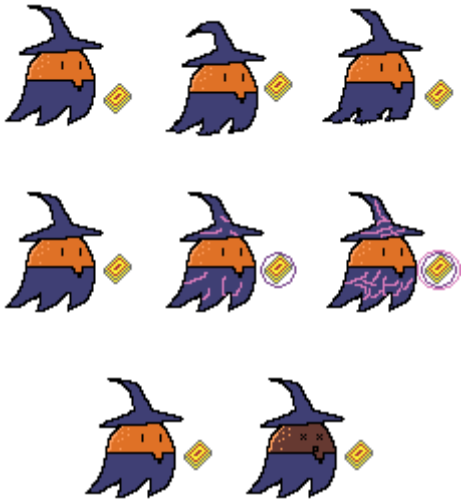
โดยจะประกอบไปด้วย 4 state

- Walk Animation
- Idle Animation
- Attack Animation
- Die Animation

รูปที่ 4.2 Cabbage Sprite

โดยจะประกอบไปด้วย 4 state

- Walk Animation & Idle Animation จะใช้ภาพเดียวกัน
- Attack Animation
- Die Animation



รูปที่ 4.3 Orange Sprite

โดยจะประกอบไปด้วย 4 state

- Walk Animation & Idle Animation จะใช้ภาพเดียวกัน
- Attack Animation
- Die Animation

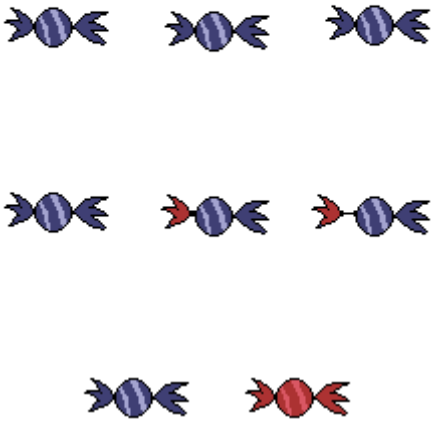
รูปที่ 4.4 Corn Sprite

โดยจะประกอบไปด้วย 4 state

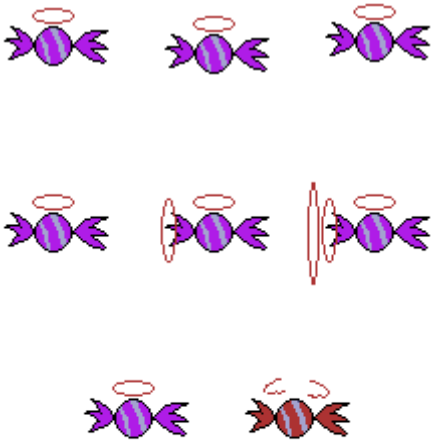
- Walk Animation
- Idle Animation
- Attack Animation
- Die Animation

รูปที่ 4.5 Projectile Sprite

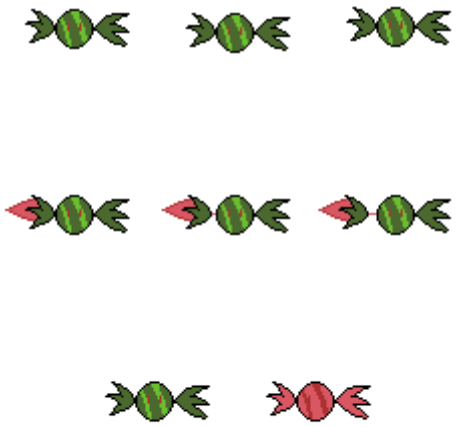
โดย Projectile sprite จะใช้กับการโจมตีระยะไกล



รูปที่ 4.6 Candy enemy melee Sprite



รูปที่ 4.7 Candy enemy ranged Sprite



รูปที่ 4.8 Big Candy Sprite

โดย enemy ทั้งสามประเภทนี้จะประกอบไปด้วย 4 state

- Walk Animation & Idle Animation จะใช้ภาพเดียวกัน
- Attack Animation
- Die Animation

รูปที่ 4.9 Sprite summon button

Sprite สำหรับใช้กดอัญเชิญตัวละครในเกมโดยที่จะประกอบไปด้วย

- summon button melee
- summon button range
- summon button tank

- summon button mage

รูปที่ 4.10 Menu Sprite

องค์ประกอบสำคัญของเมนูหลัก

ในส่วนของค่าสถานะจะเป็นไปตามตารางด้านล่างนี้

ตาราง 4.1: ยูนิทฝั่งพันธมิตร

	HEALTH POINT	DAMAGE
TOMATO	30	12
CABBAGE	50	8
ORANGE	24	25
CORN	24	12

ตาราง 4.2: ยูนิทฝั่งศัตรู

	HEALTH POINT	DAMAGE
EVIL CANDY	30	12
EVIL CANDY SHOOTER	25	15
BIG EVIL CANDY	50	24

บทที่ 5

สรุปและข้อเสนอแนะ

5.1 สรุปผล

1. พัฒนาตัวเกม Prototype โดยใช้ Godot engine ในการอำนวยการสร้างโดยที่ประกอบไปด้วย

    - ตัวเกมแบบ Prototype

    - การออกแบบ Original Character ในเกม

    - การออกแบบ Environment ในเกม

2. พัฒนา GDD หรือ game design document

    - Project Description

    - Character

    - Story

    - Gameplay

    - Game Mechanics

    - Item and power-ups

    - Progression and challenge

    - Losing

    - Art style

    - Music and Sounds

    - Technical description

โดยทั้งหมดนี้จะมีรายละเอียดทั้งหมดกล่าวอยู่ใน Game Design Document

5.2 ปัญหาและอุปสรรค

1. อุปสรรคด้านการใช้งาน script ของ Godot เนื่องจาก Godot ไม่ได้เป็น Game Engine ที่นิยมที่สุด ทำให้ในบางส่วนของโค้ดนั้น ยากต่อการแก้ไขและ หาข้อมูล ทำให้การแก้ไขบางครั้งล่าช้าและอาจจะต้องเปลี่ยนแปลงโค้ดต้นฉบับไปด้วย

2. อุปสรรคด้านการวาด Sprite เนื่องจาก การวาด Sprite หลายๆภาพมาต่อกันให้ดูเหมือนว่ามันขยับนั้นไม่ใช่เรื่องง่าย อาศัยจินตนาการสูง ทำให้ Sprite นั้นไม่ได้มีความ smooth อย่างที่คาดหวังไว้

3. อุปสรรคด้านเทคนิค บางส่วนที่เขียนไว้ใน GDD หรือ game design document นั้นไม่อาจทำออกมาได้เหมือน ครบถ้วนดังที่วางแผนไว้เนื่องด้วย อุปสรรคด้านการใช้งาน script

5.3 ข้อเสนอแนะในการพัฒนาต่อ

1. เพิ่มจำนวน stage ให้มากกว่านี้ จัดการ balance ตัว prototype ให้สมดุล

2. ทำการ port ตัวเกมเพื่อไปใช้งานในแพลตฟอร์มอื่นๆ

3. เพิ่มลูกเล่นใหม่ๆเข้าไปเพื่อเสริมช่องว่างของตัวเกม

บรรณานุกรม

[1] “GAME ENGINE คืออะไร ?,” [Online].

http://onlymeryo.blogspot.com/2018/03/game-engine.html

[2] “ข้อแตกต่างสำหรับแบบดาวน์โหลดฟรี และแบบเสียค่าใช้จ่าย,” [Online].

https://tips.thaiware.com/1334.html

[3] “Amazon Lumberyard,” [Online].

https://aws.amazon.com/th/lumberyard/

[4] “Godot engine,” [Online].

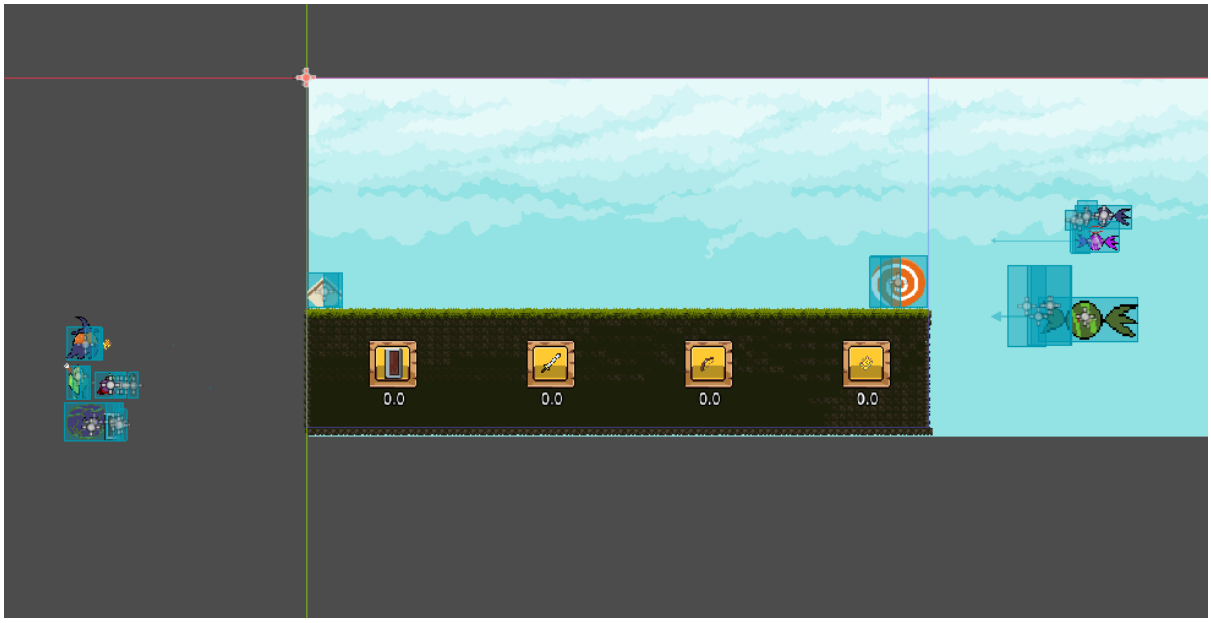
https://en.wikipedia.org/wiki/Godot_(game_engine)

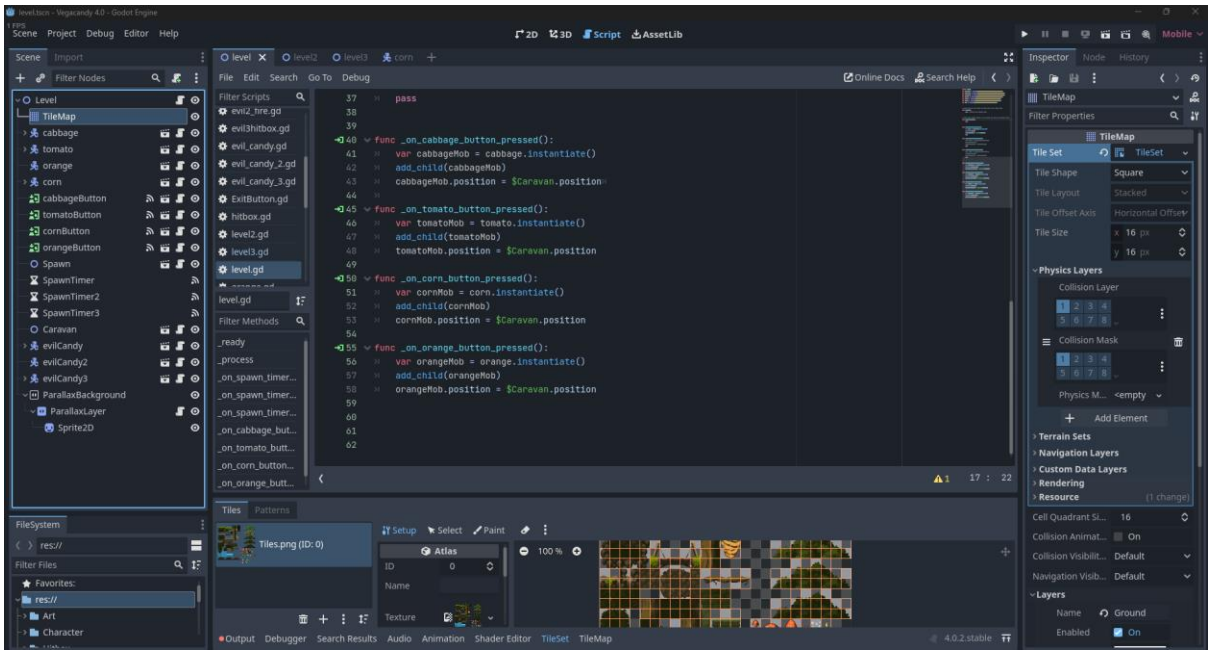
[5] “Game Design Doc,” [Online].

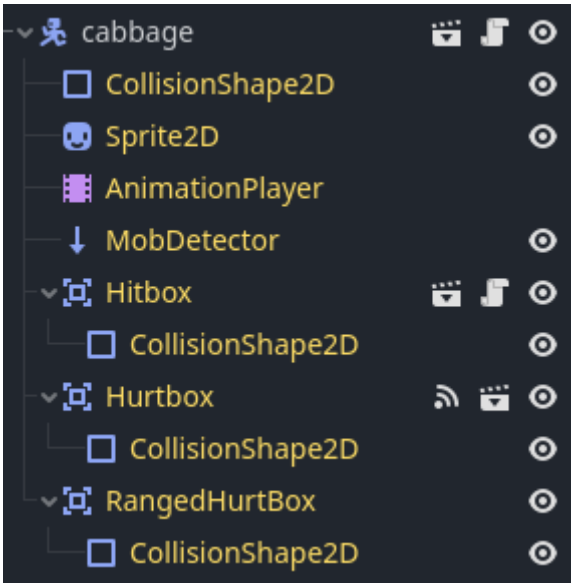
https://www.auntara.com/2022/02/23/gamedesigndocument/

ภาคผนวก

ภาพตัวอย่างเกมขณะอยู่ในระหว่างการพัฒนา



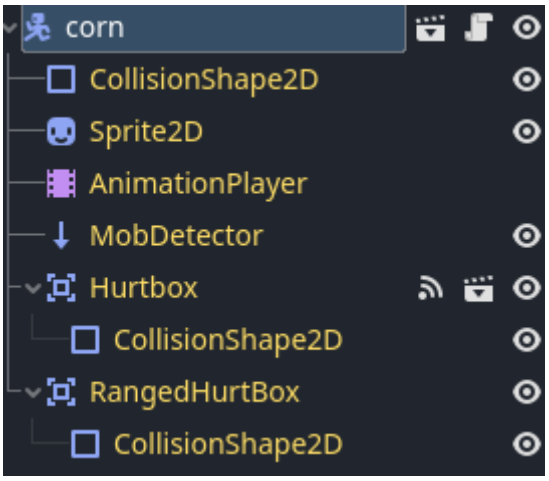




โดยที่สำหรับตัวที่โจมตีระยะประชิดจะมี Hitbox เอาไว้ใช้สร้างความเสียหายให้กับยูนิตอีกฝั่ง

```
55
56 func hit():
57     $Hitbox.monitorable = true
58     #print_debug("on")
59 func end_of_hit():
60     $Hitbox.monitorable = false
61
```

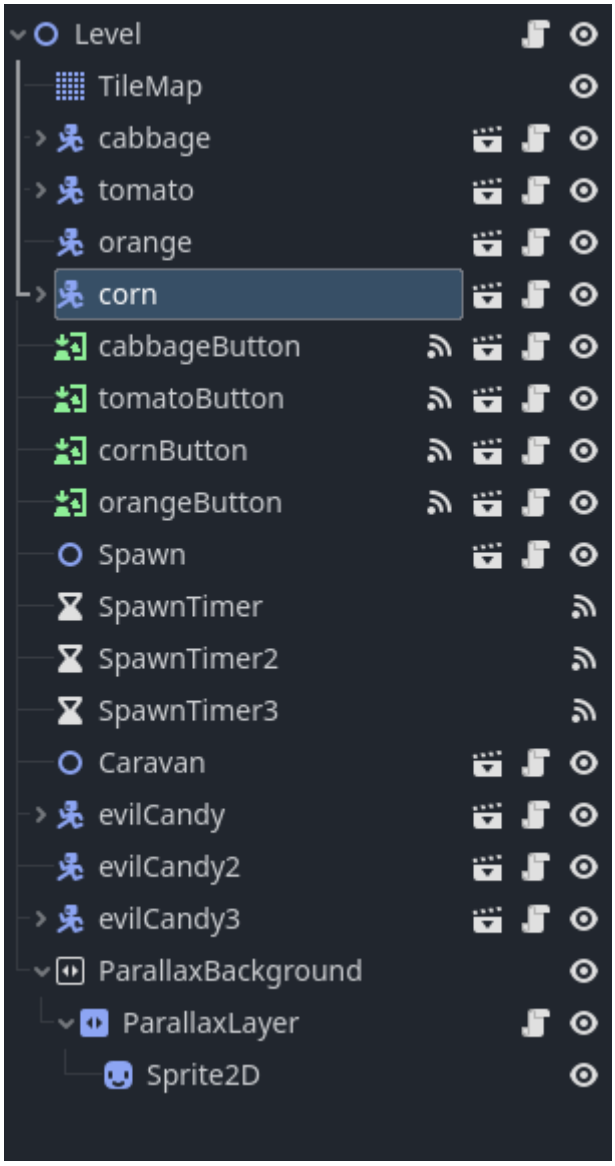
โดยจะทำงานควบคู่ไปกับ Animation player



โดยจะแตกต่างกับตัวที่ใช้การโจมตีระยะไกลซึ่งจะไม่มี Hitbox

```
4
5  func hit():
6      pass
7
8  func end_of_hit():
9      var arrowin = arrow.instantiate()
0      get_tree().current_scene.add_child(arrowin)
1      arrowin.global_position = self.global_position
```

โดยที่จะทำการ นำ projectile ระยะไกลไปแทน hitbox



องค์ประกอบนี้ทั้งหมดใน 1 เลเวลของเกม