**เอกสารรายละเอียดเกม**

**ชื่อเกม :** Pacman Slither

**ชื่อ-นามสกุลผู้พัฒนา :** นายบัณฑิต สีดาว

**รหัสนักศึกษา :** 59010759

**รายละเอียดของเกม :**

ตัวละครของเราจะได้รับบทบาทเป็น pacman ที่มีหน้าที่กินอาหารให้หมดทั้งแผนที่ ซึ่งแมพเป็นเส้นทางปิด ทุกเส้นทางที่สามารถเชื่อมต่อถึงกันได้ทั้งหมดทั้งแผนที่ นอกจากนั้น มันยังมีเงินที่ถูกสุ่มขึ้นมาที่เพื่อรับคะแนนพิเศษ

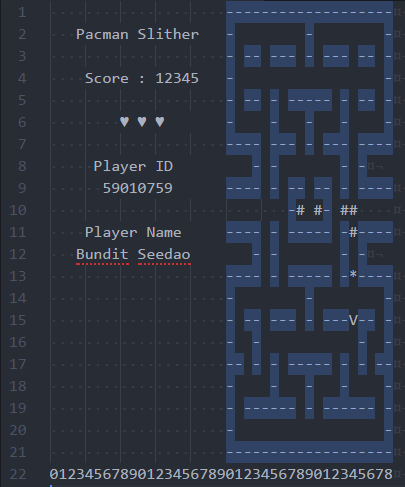
แพ็คแมนของเรานั้น จะถูกเหล่ามอนส์เตอร์ตามไล่ล่า โดยเราต้องหนีไม่ให้ถูกมอสเตอร์กิน ถึงถ้าหากเราถูกมอนส์เตอร์กินเมื่อไหร เจ้าแพ็คแมนของเราก็จะตายลงทันที แต่เราสามารถเก็บไอเท็มพิเศษที่สามารถแปลงร่าเหล่ามอนส์เตอร์ให้เป็นสัตว์ประหลาดธรรมดาที่ไม่มีพิษมีภัยอะไรเป็นช่างระยะเวลาหนึ่ง ซึ่งในห่วงนี้ เราสามารถกินเจ้ามอนศืเตอร์แล้วรับคะแนนพิเศษของพิเศษไปอีก

ในความท้าทาย หากเราถูกศัตรูกินเกมส์จะเริ่มชีวิตใหม่ให้ โดยเกมส์นี้จะมีจำนวนชีวิตให้ตามระดับความยาก 3 แบบ โดย

* ระดับง่าย จะมีชีวิตให้ 3 ชีวิต
* ระดับปานกลาง จะมีชีวิตให้ 2 ชีวิต
* และระดับยาก จะมีชีวิตให้ 1 ชีวิต

ซึ่งเกมส์นี้จะจบก็ต่อเมื่อกินอาหารได้หมดทั้งแผนที่ หรือเหลือชีวิตอย่างน้อย 1 ชีวิต

**รูปภาพ ตัวอย่างการทำงานในเกม :**

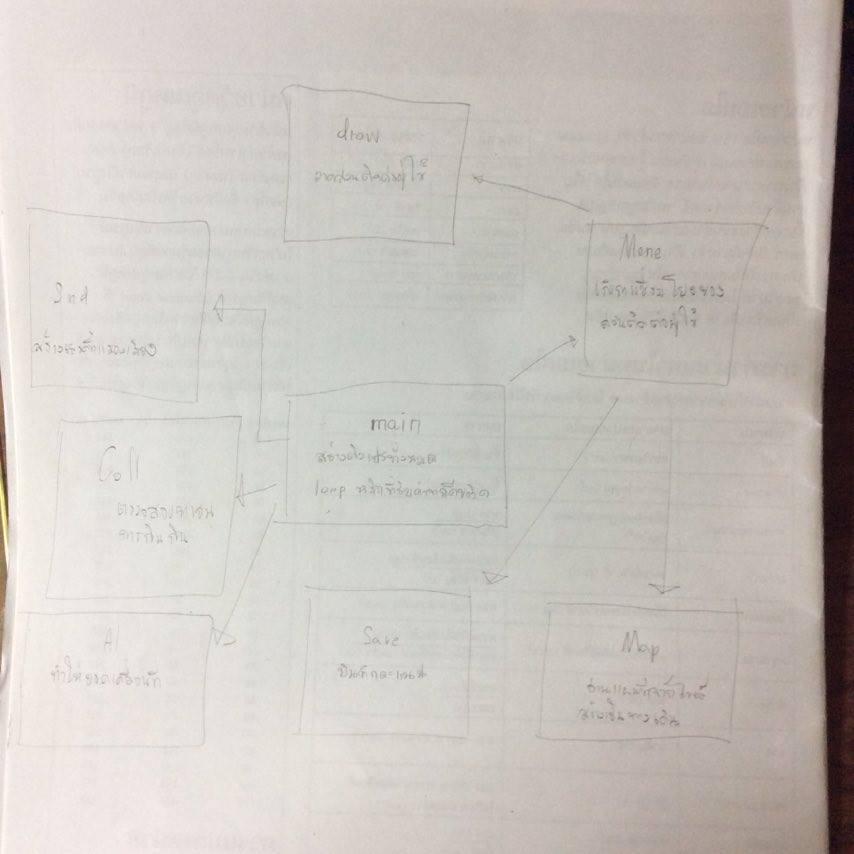


โครงร่างรูปแบบเกมส์คร่าวๆ



เกมส์ที่สร้างเส็จแล้ว

**องค์ประกอบของเกม :**



**การควบคุมภายในเกม :**

|  |  |
| --- | --- |
| **กดปุ่ม** | **ผลลัพธ์ที่ได้** |
| **Q** | บันทึกคะแนน แล้วปิดเกมส์ |
| **P** | หยุดเกมส์ชั่วคราว หรือเล่นต่อ |
| W | ตัวละครเดินย้ายตำแหน่งไปด้านหน้าเรื่อย ๆ |
| S | ตัวละครเดินย้ายตำแหน่งไปด้านหลังเรื่อย ๆ |
| A | ตัวละครเดินย้ายตำแหน่งไปด้านซ้ายเรื่อย ๆ |
| D | ตัวละครเดินย้ายตำแหน่งไปด้านขวาเรื่อย ๆ |

**การคิดคะแนนในเกม :**

* กินอาหารบนทางเดิน 1 จุด ได้ 10 คะแนน
* เก็บเงินบนทางเดิน 1 จุด ได้ 100 คะแนน
* กินศัตรูปกติ 1 ตัว ได้ 200 คะแนน

**เหตุการณ์ที่จะเกิดขึ้นในเกมโดยอัตโนมัติ :**

* เมื่อเรากดปุ่ม w, a, s, d ตัวละครเราจะเดินไปเรื่อย จนกว่าจะชนขอบแผนที่ หรือโดนศัตรูกิน
* เมื่อเราเดินผ่านอาหาร อาหารจะถูกกินอัตโนมัติ
* คะแนนที่แสดงผลจะอัพเดตตามเหตุการณ์
* ศัตรูจะสุ่มเดินอัตโนมัติ
* เมื่อเราชนศัตรู หรือเมื่อศัตรูชนเรา กรณีที่เราใช้ไอเท็มพิเศษจะมีการลบศัตรู แล้วเพิ่มคะแนน
* เมื่อเราชนศัตรู หรือเมื่อศัตรูชนเรา กรณีปกติจะมีการลบตัวเรา แล้วเริ่มชีวติใหม่ (ถ้าชีวิตเรายยังไม่หมด)
* เมื่อชีวิตเราหมด จะมีบันทึกคะแนน แล้วกลับไปหน้าเมนูหลัก

**แผนการพัฒนาเกม :**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| งาน/สัปดาห์ | 1  2ตุลา | 2  9ตุลา | 3  16ตุลา | 4  23ตุลา | 5  30ตุลา | 6  6พฤศ |
| เอกสารรายละเอียดเกม |  |  |  |  |  |  |
| ศึกษา C/C++ Style Guide  ศึกษา extended ascii  ศึกษาระบบ Multithreading  รับชื่อผู้เล่นแล้วเก็บคะแนนในไฟล์ได้  วาดส่วนติดต่อผู้ใช้คร่าวๆ  สร้างระดับความยาก |  |  |  |  |  |  |
| อ่านรูปแบบแผนที่จากเท็กไฟล์แล้ววาดบนหน้าต่าง dos  วาดตัวละครที่เคลื่อนไหวที่เดิมได้  สามรถบังคับตัวละครให้เดินตามทาง ไม่ชนกับแผนที่ได้ |  |  |  |  |  |  |
| สร้างอาหารในทางเดินได้  นับคะแนนจากการกินอาหารได้  สร้างการวาปในแผนที่  สร้างตัวละครศัตรูที่เดินด้วยโปรแกรม  สร้างให้ศัตรูสามรถกินตัวละครเราได้ และเราก็สามารถกินศัตรูได้เช่นกัน  นับคะแนนจากการกินศัตรูได้  เกมส์จบแล้วบันทึกคะแนน |  |  |  |  |  |  |
| สุ่มอาหารในทางเดินได้  แก้ไขการทับกันของตัวละคร  เก็บลายละเอียดงาน วาดส่วนติดต่อผู้ใช้ให้สวยๆ เก็บสี |  |  |  |  |  |  |
| ทดลองปล่อยให้เล่น เก็บงานให้เรียบร้อย |  |  |  |  |  |  |

**แผนการส่งความคืบหน้าของเกม :**

ความคืบหน้าครั้งที่ 1 :

1. หน้าส่วนติดต่อผู้ใช้คร่าวๆ
2. นำเสนอแผนที่
3. การเดินของตัวละคร

ความคืบหน้าครั้งที่ 2 :

1. หน้าส่วนติดต่อผู้ใช้ทั้งหมด
2. การเดินของศัตรู
3. การชน/ทับกันของตัวละคร และการคิดคะแนน
4. การบันทึกคะแนน

**เอกสารรายงานความคืบหน้า**

**ครั้งที่ 1**

**รายละเอียดการทำงานตามแผนการดำเนินการ**

|  |  |  |  |
| --- | --- | --- | --- |
| ลำดับที่ | รายละเอียด | สถานะ | |
| ตามแผน | ไม่ตามแผน |
| 1 | ศึกษา C/C++ Style Guide  ศึกษา extended ascii  ศึกษาระบบ Multithreading  รับชื่อผู้เล่นแล้วเก็บคะแนนในไฟล์ได้  วาดส่วนติดต่อผู้ใช้คร่าวๆ  สร้างระดับความยาก | **/** |  |
| 2 | ศึกษาระบบ Multithreading |  | **/** |
| 3 | อ่านรูปแบบแผนที่จากเท็กไฟล์แล้ววาดบนหน้าต่าง dos  วาดตัวละครที่เคลื่อนไหวที่เดิมได้  สามรถบังคับตัวละครให้เดินตามทาง ไม่ชนกับแผนที่ได้ | **/** |  |

**ปัญหาที่เกิดขึ้น และการแก้ปัญหา**

1. เมื่อเขียนเกมส์ไปได้สักพัก จะเริ่มรู้สึกมึนงง ว่าฟังก์ชั่นไหนใช้ตัวแปรอะไรบ้าง ตัวแปรนี้คือชื่อฟังก์ชั่น หรือชนิดตัวแปร ชื่อตัวแปรนี้เราสร้างมาไว้ทำอะไร แก้โดยการอ่าน C/C++ Style Guide
2. ปริ้นอักษรแปลกๆ ไม่ได้ แก้โดยการใช้ SetConsoleOutputCP(437);
3. การดีเลย์โดยตัวของฟังก์ชั่น Beep() ทำให้การทำงานทั้งหมดของเกมส์นั้นช้าอย่างมากเมื่อมีการเรียกใช้ซ้ำๆกัน แก้โดยการ สร้างเธรดสำหรับแยกสำหรับการาร้างเสียง Beep ต่างหาก
4. ฟังก์ชั่งบางฟังก์ชั่น เช่น time(NULL); ไม่สามารถสร้างนอกฟังก์ชั่น main() หรือเรียกใช้นอกฟังก์ชั่น main() ได้ แก้โดยการใส่ไว้ในฟังก์ชั่น main()
5. ปัญหาการโยนภาระทุกอย่างไว้ใน loop ใหญ่ ทำให้การอ่านโค้ดนั้นเป็นไปอย่างยากลำบาก แก้โดยทำให้ loop ใหญ่นั้นทำหน้าที่เพียงรับค่าจากคีย์บอร์ด โดยมีการเก็บโหมดการทำงานไว้ แล้วให้โหมดเหล่านี้โยนภาระให้แก่ฟังก์ชั่นต่างที่เกี่ยวข้องๆ
6. เมื่อเขียนโปรแกรมจนตัวแปรแกรมมีขนาดใหญ่ พบว่าการเขียนโปรแกรมใน 1 ไฟล์ นั้นทำให้การพัฒนาลำบาก แก้ได้โดยการแยกให้แต่ละไฟล์ทำเฉพาะหน้าที่ที่เกี่ยวข้อง
7. จากข้อที่ผ่านมา พบว่าการสร้างตัวแปรแบบ Global เมื่อแยกไฟล์แล้วทำให้เกิดข้อผิดพลาดที่เกิดจากการไม่พบตัวแปรเป็นจำนวนมาก แก้ได้โดยการสร้างไฟล์เก็บตัวแปรไว้ เก็บตัวแปรที่ใช้ร่วมกันแล้วเรียกใช้อันแรกสุด ส่วนตัวแปรที่มีการทำงานอิสระ จะแยกไว้ในแต่ละไฟล์ แต่คิดว่ายังเป็นการแก้ปัญหาที่ยังไม่ดีนัก
8. การวาดภาพบนหน้าต่าง จะพบว่าตำแหน่ง x, y ไม่สามารถสื่อความหมายว่าอยู่ตำแหน่งตรงไหน แก้โดยการใช้ define แล้วตั้งชื่อให้สอดคล้องที่สุด

**เอกสารรายงานความคืบหน้า**

**ครั้งที่ 2**

**รายละเอียดการทำงานตามแผนการดำเนินการ**

|  |  |  |  |
| --- | --- | --- | --- |
| ลำดับที่ | รายละเอียด | สถานะ | |
| ตามแผน | ไม่ตามแผน |
| 1 | สร้างอาหารในทางเดินได้  นับคะแนนจากการกินอาหารได้  สร้างการวาปในแผนที่ | **/** |  |
| 2 | สร้างตัวละครศัตรูที่เดินด้วยโปรแกรม  สร้างให้ศัตรูสามรถกินตัวละครเราได้ และเราก็สามารถกินศัตรูได้เช่นกัน  นับคะแนนจากการกินศัตรูได้ | **/** |  |
| 3 | การคิดคะแนน |  | **/** |
| 4 | การรวมเธรด |  | **/** |

**ปัญหาที่เกิดขึ้น และการแก้ปัญหา**

1. เมื่อเราใช้เธรด จะพบปัญหาการพิมพ์ภาพผ่านทางหน้าจอที่คาดเดาได้ยาก หรือไม่ว่าตัวแปรที่ถูกใช้งานมีค่าผิดปกติไป แก้โดยการล็อคให้เธรดใดเธรดหนึ่งทำงานเธรดเดียว โดยที่เธรดอื่นนี้เข้าคิวรอก่อน โดยใช้คำสั่ง pthread\_mutex\_lock() และ pthread\_mutex\_unlock()
2. การตรวจสอบการกินกันของ pacman กับ มอนส์เตอร์ นั้นถ้าเขียนเป็นกรณีนั้นอาจจะมีได้ถึง 16 กรณี ซึ่งหากเขียนอาจะทำให้เขียนได้ไม่ครบทุกกรณี จึงใช้ความรู้ของระยะห่างในวิชาคณิตศาสตร์มาใช้ช่วย

เมื่อเขียนเกมสืจนมีโครงสร้างที่มีขนาดใหญ่แล้ว จะพบว่าตัวแปรนั้นมีจำนวนนวนมากซึ่งสามารถสร้างความสับสนได้เช่น ตำแหน่ง x, y ของ pacman และ monster ซึ่งมีถึง 5 ตัวเลยที่เดียว ยิ่งทำให้การส่งผ่านเข้าฟังก์นั้นต้องระบบุอากิวเมนต์อีกเป็นจำนวนมาก ซึ่งในขั้นตอนนี้ได้ใช้ struct เข้ามาจัดหวดหมู่