



เกม Hard Choices รุ่นเดสก์ทอป (ผู้เล่นหลายคน)

Desktop version of Hard Choices game (Multiple Human players)

นายเกรียงไกร กิจการ รหัส 51361926

เลขที่ลงทะเบียน
วันที่รับ.....	2.5/ธ.ย./2555/.....
เลขทะเบียน.....	16290402
เลขเรียกหนังสือ.....	นร.
มหาวิทยาลัยนเรศวร	๙๖๖ ๑

๑๕๕๕
ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2555




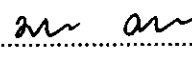
ใบรับรองโครงการวิศวกรรม

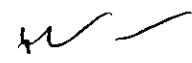
หัวข้อโครงการ เกม Hard Choices รุ่นเดสก์ท็อป (ผู้เล่นหลายคน)
ผู้ดำเนินโครงการ นายเกรียงไกร กิจการ รหัส 51361926
อาจารย์ที่ปรึกษา ดร.สุรเดช จิตประไพกุลศาล
สาขาวิชา วิศวกรรมคอมพิวเตอร์
ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา 2555

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยราชภัฏบรังษ อนุมัติให้โครงการฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

.....ที่ปรึกษาโครงการ
(ดร.สุรเดช จิตประไพกุลศาล)

.....กรรมการ
(ผศ.ดร.พนมขวัญ รियะมงคล)

.....กรรมการ
(ดร.วรลักษณ์ กงเด่นฟ้า)

.....กรรมการ
(อาจารย์จีราพร พุกสุข)

หัวข้อโครงการ	เกม Hard Choices รุ่นเดสก์ทอป (ผู้เล่นหลายคน)
ผู้ดำเนินโครงการ	นายเกรียงไกร กิจการ รหัส 51361926
อาจารย์ที่ปรึกษา	ดร.สุรเดช จิตประไพกุลสาด
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2555

บทคัดย่อ

โครงการนี้เป็นการสร้าง โปรแกรม Hard choices รุ่นเดสก์ทอป (ผู้เล่นหลายคน) เพื่อใช้ในการเรียนรู้ผลกระทบของการตัดสินใจระหว่างการพัฒนาสถาปัตยกรรมซอฟต์แวร์ ที่รองรับผู้เล่นได้ตั้งแต่ 2 ถึง 4 คน โดยผู้เล่นสามารถเล่นในเครื่องเดียวกัน หรือเล่นผ่านระบบเครือข่าย อินเทอร์เน็ตได้ โปรแกรม Hard choices ทำงานบนเครื่องคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการ วินโดวส์ ที่มีการติดตั้ง Java Developer Kit (JDK) รุ่น 1.6 หรือสูงกว่า

Project Title Desktop version of Hard Choices game (Multiple Human players)
Name Mr. Kreangkrai Kichkan ID 51361926
Project Advisor Suradet Jitprapaikulsarn, Ph.D.
Major Computer Engineering
Department Electrical and Computer Engineering
Academic year 2012

ABSTRACT

In this project we develop a desktop version of Hard choices game to support the learning about decision making during the development of software architectures. This version support 2 to 4 players and playing over the network. The program require the Java Runtime Environment (JRE) version 1.6 or higher.

กิตติกรรมประกาศ

โครงการวิศวกรรมคอมพิวเตอร์เรื่องเกม Hard Choices รุ่นเดสก์ท็อป (ผู้เล่นหลายคน) นี้ ได้สำเร็จลุล่วงได้เนื่องมาจากข้าพเจ้าได้รับความอนุเคราะห์จาก ดร.สุรเดช จิตประไพกุลศาล ซึ่งเป็นอาจารย์ที่ปรึกษาโครงการนี้ ที่กรุณาให้คำปรึกษา แนะนำวิธีการทำงาน พร้อมทั้งแนะนำแนวทางในการแก้ปัญหาต่างๆ ตลอดจนสละเวลาเพื่อตรวจสอบและแก้ไขข้อบกพร่องต่างๆ ข้าพเจ้ารู้สึกเป็นเกียรติอย่างมากที่ได้รับความอนุเคราะห์จากอาจารย์

ในโอกาสนี้ข้าพเจ้าจึงขอขอบพระคุณอาจารย์ทุกท่าน บิศา มรรคา ที่คอยสั่งสอนให้ความรู้ จนข้าพเจ้าสำเร็จการศึกษา ทั้งนี้ขอขอบคุณเพื่อนๆ พี่ๆ ที่ได้ช่วยให้คำปรึกษาบ้างเรื่องเกี่ยวกับโปรแกรมที่ทำ และคอยให้กำลังใจ จนทำให้สำเร็จลุล่วงมาด้วยดี

นายเกรียงไกร กิจการ

สารบัญ

หน้า

ใบรับรองโครงการวิจัย.....	ก
บทคัดย่อภาษาไทย.....	ข
บทคัดย่อภาษาอังกฤษ.....	ค
กิตติกรรมประกาศ.....	ง
สารบัญ.....	จ
สารบัญตาราง.....	ช
สารบัญรูป.....	ฉ
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของโครงการ.....	1
1.2 วัตถุประสงค์ของการดำเนินโครงการ.....	1
1.3 ประโยชน์ที่คาดว่าจะได้รับ.....	1
1.4 ขอบเขตของดำเนินโครงการ.....	2
1.5 ขั้นตอนของการดำเนินโครงการ.....	2
1.6 แผนการดำเนินงานตลอดโครงการ.....	3
1.7 รายละเอียดงบประมาณของโครงการ.....	4
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง.....	5
2.1 การเล่นเกม Hard Choices.....	5
2.1.1 การจัดเตรียมเกม.....	5
2.1.2 กฎ กติกา ของการเล่นเกม Hard Choices.....	6

สารบัญ (ต่อ)

	หน้า
2.1.3 ตาราง Hard Choices	8
2.1.4 การ์ดสะพาน และการ์ดเครื่องมือ	8
2.1.5 รอบการเล่น.....	8
บทที่ 3 วิธีการดำเนินโครงการ.....	10
3.1 องค์ประกอบของโปรแกรม	10
3.2 การออกแบบ และจัดทำโปรแกรม	11
3.2.1 ขั้นตอนการออกแบบไวร์เฟรมของโปรแกรมแบบ Hard choices.....	12
3.3 การทดสอบ โปรแกรม	19
3.3.1 วิธีการทดสอบโปรแกรม	19
บทที่ 4 ผลการทดลอง.....	23
4.1 การทดลอง โปรแกรมภายในเครื่องเดียวกัน.....	23
4.2 การทดลอง โปรแกรมผ่านระบบอินเทอร์เน็ต.....	31
4.2.1 วิธีการส่งข้อมูลสื่อสารกันระหว่างผู้เล่นกับเครื่องบริการ	34
4.3 การเก็บสถิติข้อมูลของผู้เล่น.....	35
4.4 สรุปผลการทดลอง	36
บทที่ 5 สรุปและข้อเสนอแนะ.....	39
5.1 สรุปผลการดำเนินงาน.....	40
5.2 ปัญหาในการดำเนินงาน และแนวทางในการแก้ไขปัญหา	40

สารบัญ (ต่อ)

	หน้า
5.3 ข้อเสนอแนะ.....	42
5.3.1 แนวทางอื่นในการแก้ปัญหา.....	42
5.3.2 แนวคิดที่อาจทำเพิ่มเติม.....	42
5.3.3 ความรู้และทักษะที่ควรมี.....	42
5.3.4 หนังสือแนะนำ.....	43
เอกสารอ้างอิง	44
ภาคผนวก การติดตั้งตัวแปลภาษาจาวา.....	46
ประวัติผู้ดำเนินโครงการ.....	50

สารบัญตาราง

ตารางที่	หน้า
1.1 แผนการดำเนินการ โครงการ.....	3
4.1 สรุปผลการทดลอง.....	36
5.1 สรุปผลการดำเนินงาน	40
5.2 ปัญหาในการดำเนินงาน และแนวทางในการแก้ไขปัญหา.....	40
5.3 แนวทางอื่นในการแก้ไขปัญหา.....	42



สารบัญรูป

รูปที่	หน้า
2.1 ตาราง Hard Choices เกม	6
2.2 การ์ดเครื่องมือ การ์ดสะพาน และตัวผู้เล่นที่จะใช้ในการเล่นเกม Hard Choices.....	7
3.1 Use Case Diagram ของเกม Hard Choices รุ่นเดสก์ทอป	11
3.2 หน้าจอหลัก ของเกม Hard Choices รุ่นเดสก์ทอป.....	12
3.3 หน้าจอสำหรับเลือกว่าจะเล่นเกมแบบออนไลน์ หรือออฟไลน์.....	13
3.4 หน้าจอสำหรับกรอกเลขที่อยู่ไอพีของเครื่องบริการ	13
3.5 หน้าจอสำหรับกรอกชื่อผู้เล่น เมื่อเล่นแบบออนไลน์.....	14
3.6 หน้าจอสำหรับกรอกชื่อผู้เล่น เมื่อเล่นแบบออฟไลน์.....	15
3.7 หน้าจอสำหรับการเล่นเกม.....	16
3.8 หน้าจอเมื่อเข้าเส้นชัย	16
3.9 หน้าจอสรุปผลการแข่งขัน	17
3.10 หน้าจอกฎกติกาการเล่นเกม Hard Choices	18
3.11 หน้าจอเกี่ยวกับผู้จัดทำ โปรแกรมเกม Hard Choices	18
3.12 หน้าจอก่อนออกจาก โปรแกรมเกม Hard Choices	19
3.13 ขั้นตอนการสร้างคลาส JUnit Test	20
3.14 ตั้งชื่อให้กับคลาส JUnit Test	20
3.15 เขียนการทดสอบฟังก์ชัน	21
3.16 ทำการทดสอบ	21
3.17 พบข้อผิดพลาดจากการทดสอบ	22
3.18 ผลการทดสอบผ่าน	22
4.1 เลือกเล่นเกมแบบออฟไลน์	23
4.2 หน้าจอกรอกชื่อตัวผู้เล่น	24
4.3 หน้าจอเกม	24

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.4 การทอยลูกเต๋า.....	25
4.5 เมื่อผู้เล่นคนแรกเล่นจบ.....	25
4.6 การเก็บการ์ดเครื่องมือ	26
4.7 การเดินข้ามสะพาน.....	26
4.8 จำนวนการ์ดสะพานที่อยู่ในมือ ลบแต้มลูกเต๋า.....	27
4.9 เมื่อมีการ์ดสะพานอยู่ในมือสามารถคลิกปุ่ม Skip ได้.....	27
4.10 หลังจากคลิกปุ่ม Skip.....	28
4.11 เมื่อมีผู้เล่นเข้าเส้นชัย.....	28
4.12 การสรุปผลการแข่งขัน.....	29
4.13 ตัวอย่างการเขียน โปรแกรมแบบ Hard code	30
4.14 ตัวอย่างการเขียน โปรแกรมโดยใช้ตัวแปร	30
4.15 เปิดเครื่องบริการ (server).....	31
4.16 เลือกเล่นเกมแบบออนไลน์	31
4.17 การกรอกเลขที่อยู่ไอพีของเครื่องบริการ.....	32
4.18 ข้อความแจ้งเตือนเมื่อกรอกเลขที่อยู่ไอพีของเครื่องบริการผิด.....	32
4.19 การกรอกชื่อตัวผู้เล่น.....	33
4.20 หลังจากกรอกชื่อตัวผู้เล่นเสร็จ.....	33
4.21 เมื่อมีผู้เล่นคนที่สองเข้ามา.....	34
4.22 การเก็บสถิติ	35

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของโครงการ

เนื่องจากเกม Hard choices นี้เป็นเกมที่ใช้ในการเรียนรู้ถึงผลกระทบของการตัดสินใจ ที่อาจเกิดจากความไม่แน่นอน ความเสี่ยง และหนี้ทางเทคนิค (Technical debt) ใช้สำหรับผู้เล่นตั้งแต่ 2 ถึง 4 คน ออกแบบโดยทีมนักวิจัยจากสถาบันวิศวกรรมซอฟต์แวร์ (Software Engineering Institute) มหาวิทยาลัยคาร์เนกี เมลลอน (Carnegie Mellon University) สหรัฐอเมริกา ซึ่งปัจจุบันเป็นเกมกระดาน ทำให้ไม่สะดวกในการเล่น ด้วยเหตุนี้จึงดำเนินการพัฒนาโปรแกรมเกม Hard choices รุ่นเดสก์ท็อป ในลักษณะที่มนุษย์แข่งกับมนุษย์ ให้สามารถเล่นได้บนเครื่องเดียวกัน และเล่นผ่านระบบเครือข่ายอินเทอร์เน็ตได้ ทำให้สะดวกในการเล่นมากขึ้น

1.2 วัตถุประสงค์ของการดำเนินโครงการ

1. เพื่อสร้างเกม Hard choices รุ่นเดสก์ท็อป สำหรับผู้เล่นตั้งแต่ 2 ถึง 4 คน และสามารถเล่นผ่านระบบเครือข่ายอินเทอร์เน็ตได้

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1. ผู้เล่นเกม Hard choices มีความสะดวกในการเรียนรู้มากขึ้น
2. ผู้เล่นเกม Hard choices ได้ศึกษาผลกระทบของการตัดสินใจในสถานการณ์ต่างๆ
3. ผู้เล่นเกม Hard choices ได้ศึกษาผลกระทบจากการตัดสินใจของตนในลักษณะต่างๆ
4. ผู้พัฒนาได้ฝึกฝนทักษะการเขียนโปรแกรม และได้ฝึกฝนเทคนิคในการพัฒนาโปรแกรม
5. ผู้พัฒนาได้นำความรู้ที่ได้ศึกษามาประยุกต์ใช้ในการพัฒนาโปรแกรมอย่างเต็มที่

1.4 ขอบเขตของดำเนินโครงการ

1. สามารถเล่นเกม Hard Choices บนเครื่องคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการวินโดวส์ ที่มีการติดตั้ง Java Developer Kit (JDK) รุ่น 1.6 หรือสูงกว่า
2. ใช้งานได้กับระบบจอภาพของคอมพิวเตอร์ที่มีความละเอียด 1366 x 768
3. ผู้เล่นสามารถเล่นได้ตั้งแต่ 2 ถึง 4 คน
4. มีการเก็บสะสมคะแนน เพื่อหาผู้ชนะในเกม
5. สามารถเล่นผ่านระบบเครือข่ายอินเทอร์เน็ตได้
6. สามารถเก็บข้อมูลการเล่นของผู้เล่นลงในแฟ้มข้อความ (text file) ได้

1.5 ขั้นตอนของการดำเนินโครงการ

1. ศึกษาวิธีการเล่นเกมและกติกาในเกม Hard Choices
2. ศึกษาสถาปัตยกรรมซอฟต์แวร์ในการออกแบบของเกม Hard Choices
3. ศึกษาการสร้างส่วนติดต่อผู้ใช้แบบกราฟิก (Graphical User Interface) ในภาษาจาวา
4. ออกแบบการสร้างส่วนติดต่อผู้ใช้แบบกราฟิกของเกมด้วยภาษาจาวา
5. ศึกษาการสร้างโปรแกรมที่ทำงานผ่านระบบเครือข่ายอินเทอร์เน็ต
6. ตรวจสอบความถูกต้องของโปรแกรม และทำการทดสอบโปรแกรม
7. รวบรวมข้อมูลในการทำโครงการนำมาจัดทำเป็นรูปเล่มโครงการ

1.6 แผนการดำเนินงานตลอดโครงการ

ตารางที่ 1.1 แผนการดำเนินการโครงการ

ลำดับ	การดำเนินการ	ปี 2554										ปี 2555							
		ม.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	
1	ศึกษาวิธีการเล่นและกติกาในการเล่นเกม	↕																	
2	ศึกษาสถาปัตยกรรมซอฟต์แวร์ในการออกแบบของเกม	↕																	
3	ศึกษาการสร้างส่วนติดต่อผู้ใช้แบบกราฟิกของภาษาจาวา				↕														
4	ออกแบบและสร้างส่วนติดต่อผู้ใช้แบบกราฟิกของเกม																		
5	ศึกษาและสร้างโปรแกรมให้ทำงานผ่านระบบเครือข่ายอินเทอร์เน็ต																		
6	ตรวจสอบความถูกต้องและทดสอบโปรแกรม																		
7	รวบรวมข้อมูลนำมาจัดทำรูปเล่มโครงการ																		

1.7 รายละเอียดงบประมาณของโครงการ

1. ค่าเช่าเล่มรายงานฉบับสมบูรณ์

เป็นเงิน 1,000 บาท

รวมเป็นเงินทั้งสิ้น 1,000 บาท
(หนึ่งพันบาทถ้วน)



บทที่ 2

หลักการและทฤษฎีที่เกี่ยวข้อง

เกม Hard Choices ถูกพัฒนาขึ้นเพื่อให้ผู้ที่เล่นเกมนี้มีความเข้าใจได้ดีขึ้นถึงกลยุทธ์ที่พวกเขาใช้ระหว่างการพัฒนาซอฟต์แวร์ และสิ่งที่เกี่ยวข้องกับการลงทุน ความพยายามที่จะได้รับประโยชน์หรือการจ่ายเพื่อแลกกับการใช้วิธีลัด การอภิปรายในช่วงสอบถามรายละเอียดหลังเกมอาจเกี่ยวข้องกับเรื่องต่อไปนี้

- หนี้ทางเทคนิคและการลงทุน
- วิธีการประเมินการเปลี่ยนแปลงเงื่อนไขและอุปสรรค
- วิธีการของตัวบุคคล และการเก็บรวบรวมกลยุทธ์ เปลี่ยนเป็นกระบวนการของโครงการ

เรื่องต่างๆ เหล่านี้จะช่วยให้ผู้เล่นเข้าใจสิ่งที่อาจเกิดขึ้นในระหว่างกระบวนการพัฒนาซอฟต์แวร์ การวางแผน และการตัดสินใจต่าง ๆ เพื่อให้ได้ผลลัพธ์ที่ดีและเหมาะสมที่สุด ไม่ว่าจะเป็นการลงทุนใช้ทางลัดเพื่อทำให้งานเสร็จเร็ว แต่ก็อาจทำให้เกิดหนี้ได้ในภายหลังได้

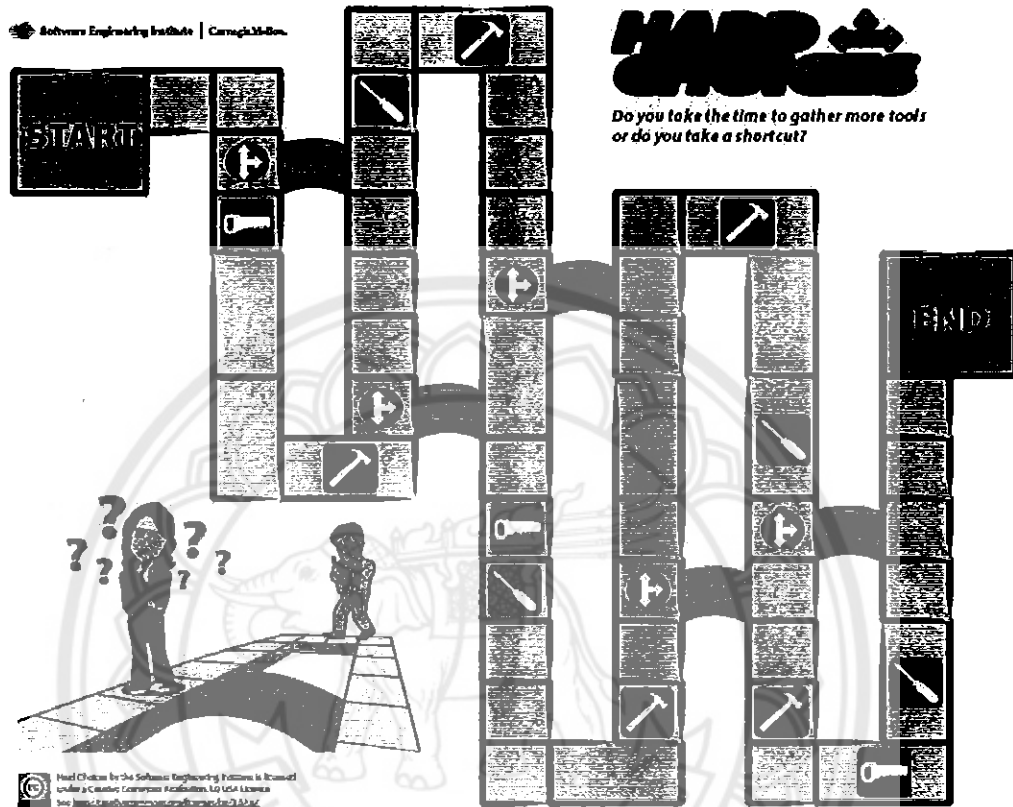
2.1 การเล่นเกม Hard Choices

เกม Hard Choices จะใช้เวลาประมาณ 1 ชั่วโมงในการเล่น และรวมถึงระยะเวลาสอบถามรายละเอียดในภายหลังจากจบเกม เพื่อให้ผู้เล่นสามารถพูดคุยแลกเปลี่ยนประสบการณ์การเล่นของพวกเขา เกมนี้อาจจะเล่นได้ 2 คน 3 คน หรือ 4 คน ผู้เล่นทำการแข่งขันกัน และเมื่อการแข่งขันเสร็จสิ้นลง ผู้เล่นที่มีคะแนนมากที่สุดในตอนท้ายของเกมจะเป็นผู้ชนะ

2.1.1 การจัดเตรียมเกม

เกม Hard Choices แสดงให้เห็นถึงกิจกรรมของการพัฒนาซอฟต์แวร์ ในการแสวงหาผู้ที่จะเป็นผู้นำในตลาด ผู้เล่นจะแข่งขันกันปล่อยผลิตภัณฑ์ของตนสู่ตลาด ผู้เล่นได้รับคะแนนเมื่อลงไปอยู่ในช่องที่มีการเคลื่อนไหว (คิดเป็นผลตอบแทนการลงทุนในโครงสร้างพื้นฐานทางเทคนิค) หรือไม่จบเกมเป็นคนสุดท้าย (คิดเป็นผลตอบแทนสำหรับความเร็วในการตลาด)

วัสดุที่ใช้สำหรับการเล่นประกอบด้วย กระดานเกม Hard Choices ลูกเต๋าหกด้าน เครื่องหมายตัวผู้เล่นซึ่งมีไว้สำหรับให้ผู้เล่นใช้เดินไปรอบ ๆ กระดานเกม การ์ดเครื่องมือสำหรับ รางวัล และการตัดสินใจสำหรับการลงโทษ



รูปที่ 2.1 ตาราง Hard Choices เกม

ที่มา : <http://www.sei.cmu.edu/downloads/arch/tools/HardChoices.pdf>

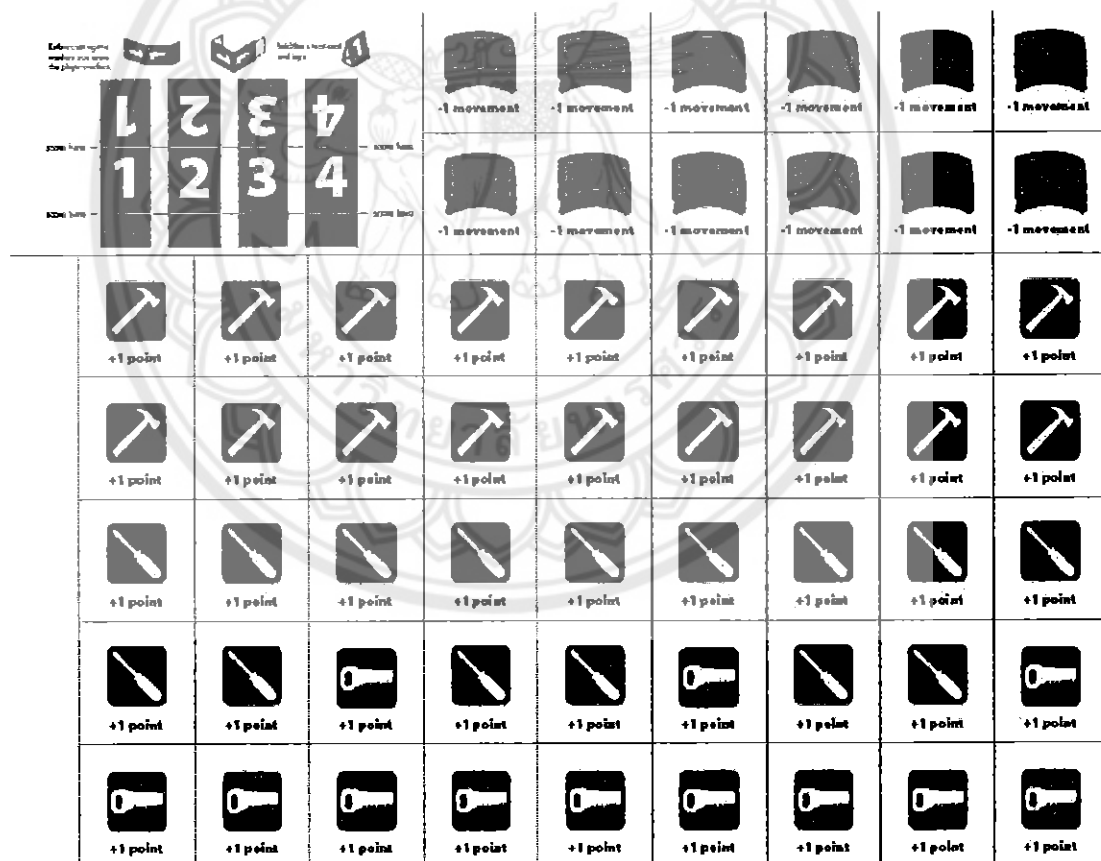
2.1.2 กฎ กติกา ของการเล่น Hard Choices

1. ตำแหน่งหนึ่งบนกระดานจะมีการทำเครื่องหมาย START และอีกตำแหน่งหนึ่งมีการทำเครื่องหมาย END ผู้เล่นทุกคนจะวางตัวเครื่องหมายของพวกเขาอยู่ในตำแหน่ง START ผู้เล่นจะเริ่มต้นด้วยรางวัลและการลงโทษเป็นศูนย์
2. ผู้เล่นแต่ละคนทอยลูกเต๋า ใครทอยได้แต้มมากที่สุดจะได้เริ่มเล่นก่อน จากนั้นผู้ที่อยู่ตามเข็มนาฬิกาเป็นผู้เล่นต่อไป
3. ในระหว่างคาให้ผู้เล่นแต่ละคนทอยลูกเต๋าเพื่อกำหนดจำนวนช่องที่จะย้ายไป

ก. ถ้าผู้เล่นถือการ์ดสะพานอยู่ ผู้เล่นจะทำการย้ายเครื่องหมายของเขา ตามจำนวนเลขที่ระบุบนลูกเต๋าลบด้วยจำนวนการลงโทษที่เกิดขึ้นจากการที่ผู้เล่นถือการ์ดสะพานอยู่

ข. ผู้เล่นอาจเลือกที่จะย้ายไปในทิศทางใดทิศทางหนึ่ง รวมทั้งเดินย้อนกลับไปได้เพื่อเป็นการเพิ่มโอกาสสำหรับผู้เล่นที่จะเดินไปยังช่องที่มีการ์ดเครื่องมือ

4. ผู้เล่นคนแรกที่เดินไปถึงช่อง END จะได้รับ 5 คะแนน คนที่สองได้รับ 3 คะแนน และคนที่สามที่ได้รับ 1 คะแนน ผู้เล่นยังจะได้รับ 1 คะแนนสำหรับการ์ดเครื่องมือแต่ละใบเพื่อที่จะเดินเข้าสู่ช่อง END ผู้เล่นควรทอยลูกเต๋ให้ได้เลขอะไรก็ได้ให้เท่ากับหรือมากกว่าจำนวนของช่องทางเดินที่เหลืออยู่ เกมจะสิ้นสุดเมื่อผู้เล่นคนหนึ่งยังคงอยู่บนกระดาน ผู้เล่นจะต้องหยุดและไม่เดินต่อ ผู้เล่นจะได้รับคะแนนสำหรับการ์ดเครื่องมือแต่ละใบที่เก็บมาระหว่างเกม แต่จะไม่ได้รับคะแนนสำหรับการเดินไปถึงช่อง END ผู้เล่นที่มีคะแนนมากที่สุดในตอนท้ายของเกมจะเป็นผู้ชนะ



รูปที่ 2.2 การ์ดเครื่องมือ การ์ดสะพาน และตัวผู้เล่นที่จะใช้ในการเล่นเกม Hard Choices

ที่มา : <http://www.sei.cmu.edu/downloads/arch/tools/HardChoices.pdf>

2.1.3 ตาราง Hard Choices

เมื่อผู้เล่นเข้ามาอยู่ที่ช่อง Hard Choices จะต้องตัดสินใจว่าจะข้ามสะพานทางลัด หรือจะไปทางที่ไกลกว่า และพยายามที่จะเก็บการ์ดเครื่องมือให้ได้มากที่สุด

2.1.4 การ์ดสะพาน และการ์ดเครื่องมือ

1. การ์ดสะพานนับเป็นหนึ่งช่องในการเดินคล้ายกับช่องตารางทางเดินปกติ
2. ผู้เล่นที่เลือกที่จะข้ามสะพานทางลัดจะต้องรวบรวมการ์ดสะพานแต่ละใบไว้หักหนึ่ง แต่จากการทยอยลูกเต๋าของทุกทุกตาไป
3. ผู้เล่นอาจจะกำจัดการ์ดสะพาน โดยการข้ามตารางหว่างการเล่นเกมนของตัวเอง
4. ถ้าผู้เล่นอยู่บนตารางเครื่องมือ
 - ก. ถ้าผู้เล่น ไม่ได้มีการ์ดเครื่องมือที่นั่นอยู่ ผู้เล่นจะได้รับการ์ดเครื่องมือที่นั่น
 - ข. ถ้าผู้เล่นมีการ์ดเครื่องมืออยู่แล้ว ผู้เล่นอาจจะใช้การ์ดเครื่องมือ (คืนการ์ดกลับไปยังสำรับ) และจะได้รับดาฟรี่ 1 ตา หรือเก็บการ์ดเครื่องมือไว้
 - ค. ผู้เล่นอาจจะเก็บรวบรวมการ์ดได้การ์ดเดียวสำหรับช่องที่กำหนดภายในรอบที่กำหนดเท่านั้น นั่นคือผู้เล่นอาจจะไม่เดินย้อนกลับไปเพื่อไปยังช่องเครื่องมือ และเก็บการ์ดสำหรับช่องนั้นเป็นใบที่สอง

2.1.5 รอบการเล่น

เมื่อผู้เล่นที่ได้ดำเนินการเดินเสร็จสิ้นในรอบแรก ผู้คุมเกมควรจะประกาศการเล่นรอบที่สอง และเพิ่มกระดานอื่น (แนะนำเป็นรุ่นที่สอง)

1. วางแผนประมาณ 20 - 30 นาทีในการเล่นรอบต่อไป
2. ผู้เล่นที่เล่นรอบที่สองนี้จะมีรางวัลและการลงโทษจากรอบก่อน ในการเล่นใช้ประยุกต์กับทุกรอบ
3. เพื่อให้เข้าใจ ผู้เล่นควรจะเล่นให้เสร็จอย่างน้อยสองรอบของเกม ผู้คุมเกมอาจจะตัดรอบที่สองให้สั้น ถ้าเขารู้สึกว่าผู้ร่วมเล่นมีความเข้าใจแนวคิดของเกม
4. ผู้คุมเกมอาจจะทำการเปลี่ยนการใช้งานของเกมก่อนที่จะเริ่มต้นเกมใหม่ในแต่ละรอบ (ยกเว้นในรอบแรก) ผู้คุมเกมให้หนึ่งในผู้เล่นเลือกส้อมการ์ด เปลี่ยนเกม โดยเพิ่มกฎของรางวัล

(เครื่องมือ) และการลงโทษ (สะพาน) และความคลุมเครือของเงื่อนไขเอง บางคนแนะนำให้เปลี่ยนเกมเป็น

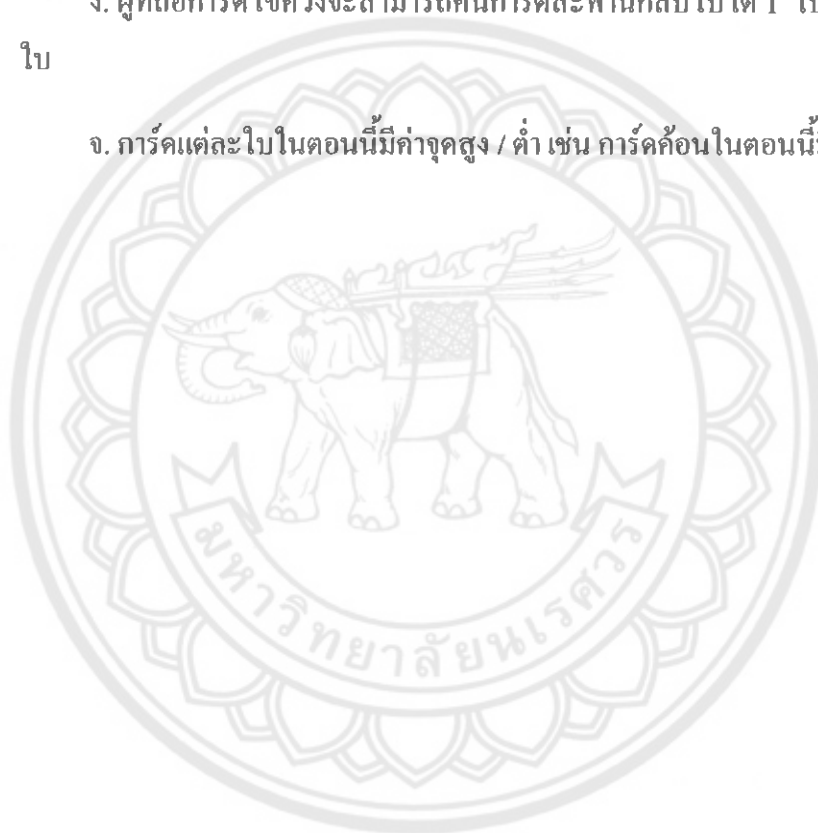
ก. ผู้ถือการ์ดของค้อนสามารถข้ามสะพานโดยไม่มีกรลงโทษถ้าพวกเขาให้การ์ดสะพานกลับคืน

ข. สำหรับผู้เล่นที่ถือการ์ดเลื่อยให้ บวก 1 ของแต่ละการทอยลูกเต๋า

ค. สำหรับผู้เล่นที่ถือการ์ดไขควงให้ ลบ 1 จากแต่ละการทอยลูกเต๋า

ง. ผู้ที่ถือการ์ดไขควงจะสามารถคืนการ์ดสะพานกลับไปได้ 1 ใบต่อการ์ดไขควง 1 ใบ

จ. การ์ดแต่ละใบในตอนนี้มีค่าจุดสูง / ต่ำ เช่น การ์ดค้อนในตอนนี้มีค่า 5 คะแนน



บทที่ 3

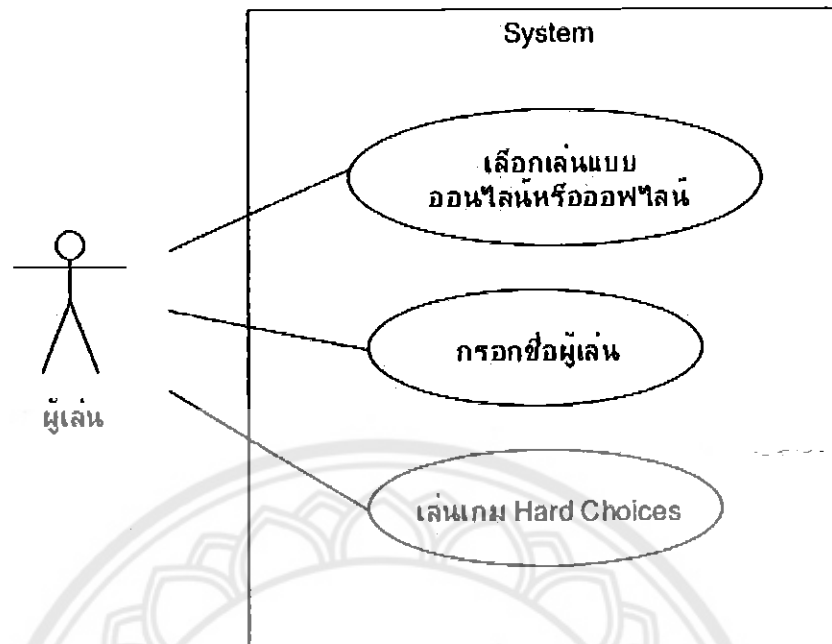
วิธีการดำเนินโครงการ

ในบทนี้จะกล่าวถึงขั้นตอนการออกแบบเกม Hard Choices รุ่นเดสก์ทอป (ผู้เล่นหลายคน) มีขั้นตอนการออกแบบที่ละเอียดเพื่อให้สามารถเข้าใจระบบเกมได้อย่างชัดเจน มีกระบวนการทำงานที่สัมพันธ์กับความต้องการของระบบเกม และเพื่อให้ระบบเกมที่ออกมาสอดคล้องกับแผนงานที่เตรียมไว้ จากการที่ได้ศึกษาข้อมูลต่างๆ ของรูปแบบการเล่นเกม Hard Choices ศึกษาการออกแบบการเขียนโปรแกรมด้วยภาษาจาวา รวมทั้งโครงสร้างการออกแบบระบบจากหลักการของ UML รวมทั้งการค้นคว้าทฤษฎีและงานที่เกี่ยวข้องสามารถออกแบบโปรแกรมเกม Hard Choices ให้มีความถูกต้องของข้อมูล การทำงานของระบบเกมเป็นไปได้อย่างมีประสิทธิภาพ

3.1 องค์ประกอบของโปรแกรม

ในการออกแบบกระบวนการต่างๆ สามารถแบ่งแยกออกเป็นส่วนของการออกแบบระบบได้หลายขั้นตอน ในที่นี้จะขอยกตัวอย่าง Use Case Diagram มาอธิบาย

Use Case Diagram คือ แผนภาพที่แสดงการทำงานของผู้ใช้งานระบบและความสัมพันธ์กับระบบย่อยภายในระบบใหญ่ ในการเขียน Use Case Diagram ผู้ใช้ระบบจะถูกกำหนดค่าให้เป็น Actor และระบบย่อยคือ Use Case จุดประสงค์หลักของการเขียน Use Case Diagram ก็เพื่อเล่าเรื่องราวทั้งหมดของระบบว่ามีการทำงานอะไรบ้าง เป็นการดึงความต้องการ หรือเรื่องราวต่างๆ ของระบบจากผู้ใช้งาน ซึ่งถือว่าเป็นจุดเริ่มต้นในการวิเคราะห์และออกแบบระบบ สัญลักษณ์ที่ใช้ใน Use Case Diagram จะใช้สัญลักษณ์รูปคนแทน Actor ใช้สัญลักษณ์วงรีแทน Use Case และใช้เส้นตรงในการเชื่อม Actor กับ Use Case เพื่อแสดงการใช้งานของ Use Case ของ Actor นอกจากนี้ Use Case ทุกๆ ตัวจะต้องอยู่ภายในสี่เหลี่ยมเดียวกันซึ่งมีชื่อของระบบระบุอยู่ด้วย ซึ่งในเกม Hard Choices จะมี Use Case Diagram ดังนี้



รูปที่ 3.1 Use Case Diagram ของเกม Hard Choices รุ่นเคสก์ทอป

Use Case Diagram ที่เป็นการจำลองภาพการทำงานของระบบเกม Hard Choices ซึ่งจะเห็นได้ว่า ระบบนี้ประกอบไปด้วย 3 ระบบย่อย คือ

1. ระบบการเลือกเล่นแบบออนไลน์หรือออฟไลน์
2. ระบบการรับชื่อผู้เล่น
3. ระบบการเล่นเกม Hard Choices

3.2 การออกแบบ และจัดทำโปรแกรม

กระบวนการหนึ่งของการจะได้มาซึ่ง โปรแกรมที่มีคุณภาพคือการออกแบบที่ดี และสิ่งที่สำคัญที่สุดของการออกแบบคือการทำไวร์เฟรม (Wireframe) เพราะจะช่วยรวบรวมความคิดต่างๆ มาจัดให้เป็นระเบียบ ในรูปแบบโครงร่างคร่าวๆ ทำให้เข้าใจภาพรวม หน้าตา และการทำงานต่างๆ ภายในโปรแกรม นอกจากนี้ไวร์เฟรมยังจะช่วยให้เห็นรูปร่างหน้าตาของโปรแกรมที่ใกล้เคียงกับของจริงมากที่สุด และเมื่อเกิดปัญหาขึ้นสามารถทำการแก้ไขที่ไวร์เฟรมก่อนได้หากไม่ได้ทำไวร์เฟรม ขึ้นมาก่อนแล้ว เมื่อพบปัญหาภายหลังจากการออกแบบไปแล้ว จะทำให้ต้องเสียเวลาในแก้ไขมากกว่า

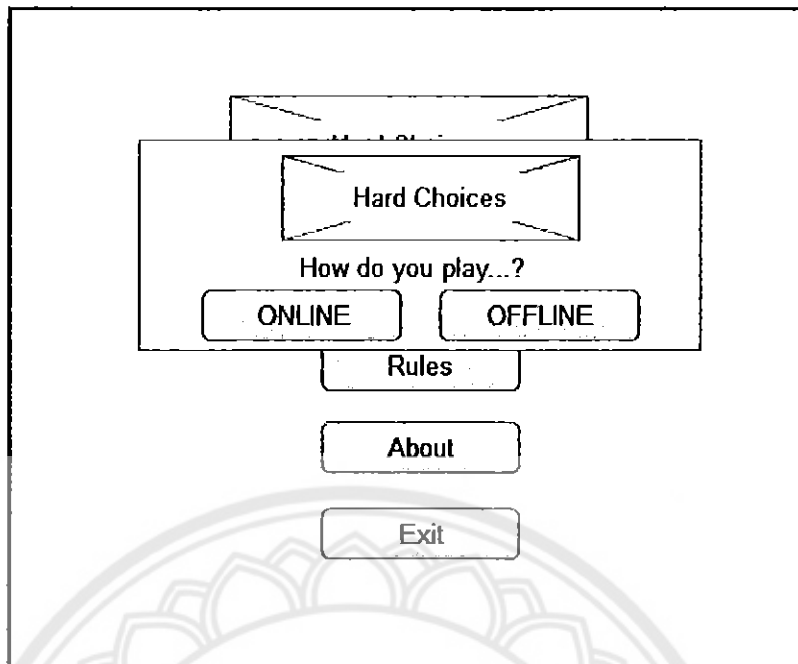
3.2.1 ขั้นตอนการออกแบบเว็บไซต์เฟรมของโปรแกรมเกม Hard choices

1. ออกแบบหน้าจอหลักของโปรแกรม จะประกอบไปด้วย ภาพตราสัญลักษณ์ของโปรแกรม อยู่บริเวณตรงกลางด้านบนของหน้าจอ ปุ่ม Play ปุ่ม Rules ปุ่ม About และปุ่ม Exit อยู่บริเวณตรงกลางของหน้าจอไล่ลงมาตามลำดับ เมื่อคลิกปุ่ม Play จะปรากฏหน้าจอสำหรับเลือกว่าจะเล่นเกมแบบออนไลน์หรือออฟไลน์ ดังรูปที่ 3.3 และเมื่อคลิกปุ่ม Rules จะมาอยู่ที่หน้าจอกฎกติกาการเล่นเกม Hard choices ดังรูปที่ 3.10 และเมื่อคลิกปุ่ม About จะมาอยู่ที่หน้าจอเกี่ยวกับผู้จัดทำเกม Hard choices ดังรูปที่ 3.11 และเมื่อคลิกปุ่ม Exit จะปรากฏหน้าจอก่อนออกจากเกม Hard choices ดังรูปที่ 3.12



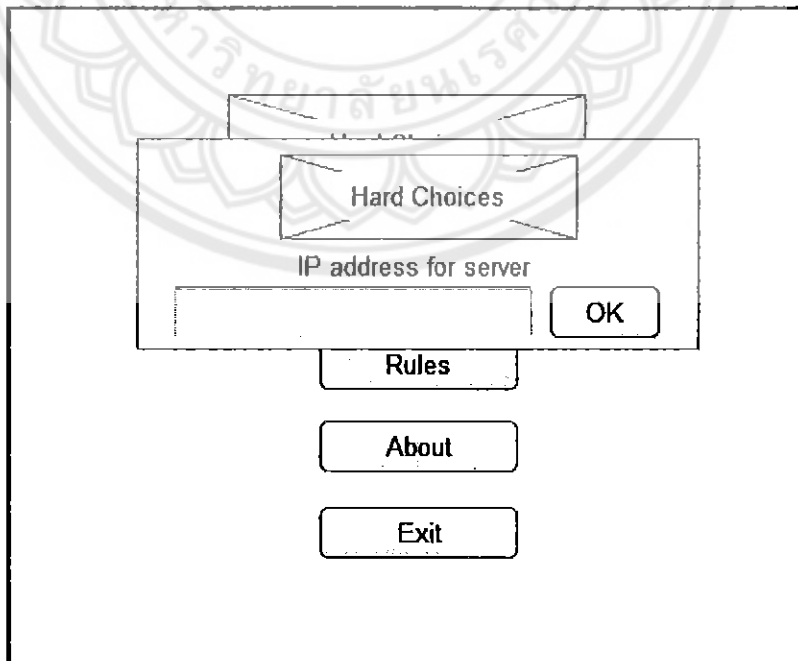
รูปที่ 3.2 หน้าจอหลัก ของเกม Hard Choices รุ่นเดสก์ทอป

2. ออกแบบหน้าจอสำหรับเลือกว่าจะเล่นเกมแบบออนไลน์หรือออฟไลน์ เมื่อคลิกปุ่ม Play ที่หน้าจอหลักของโปรแกรม จะประกอบไปด้วย ภาพตราสัญลักษณ์ของโปรแกรม ข้อความคำถามว่าจะเล่นแบบไหน ปุ่ม Online เมื่อคลิกปุ่ม Online จะปรากฏหน้าจอสำหรับกรอกเลขที่อยู่ไอพีของเครื่องบริการ ดังรูปที่ 3.4 และปุ่ม Offline อยู่ถัดจากปุ่ม Online เมื่อคลิกปุ่ม Offline จะมาอยู่หน้าจอสำหรับกรอกชื่อผู้เล่น ดังรูปที่ 3.6



รูปที่ 3.3 หน้าจอสำหรับเลือกว่าจะเล่นเกมแบบออนไลน์ หรือออฟไลน์

3. ออกแบบหน้าจอสำหรับกรอกเลขที่อยู่ไอพีของเครื่องบริการ เมื่อคลิกปุ่ม Online จะประกอบไปด้วย ภาพตราสัญลักษณ์ของโปรแกรม ข้อความบอกให้กรอกเลขที่อยู่ไอพีของเครื่องบริการ ช่องสำหรับกรอกเลขที่อยู่ไอพีของเครื่องบริการ และปุ่ม Ok จะอยู่ถัดจากช่องสำหรับกรอกเลขที่อยู่ไอพี เมื่อคลิกปุ่ม Ok จะปรากฏหน้าจอสำหรับกรอกชื่อผู้เล่น ดังรูปที่ 3.4



รูปที่ 3.4 หน้าจอสำหรับกรอกเลขที่อยู่ไอพีของเครื่องบริการ

4. ออกแบบหน้าจอสำหรับกรอกชื่อผู้เล่น เมื่อคลิกปุ่ม Ok ที่หน้าจอสำหรับกรอกเลขที่อยู่ไอพีของเครื่องบริการ จะประกอบไปด้วย ภาพตราสัญลักษณ์ของโปรแกรม ข้อความบอกให้กรอกชื่อผู้เล่น ช่องสำหรับกรอกชื่อผู้เล่น และปุ่ม Join จะอยู่ถัดจากช่องสำหรับกรอกชื่อ เมื่อคลิกปุ่ม Join จะมาอยู่หน้าจอสำหรับกรอกชื่อผู้เล่น ดังรูปที่ 3.6

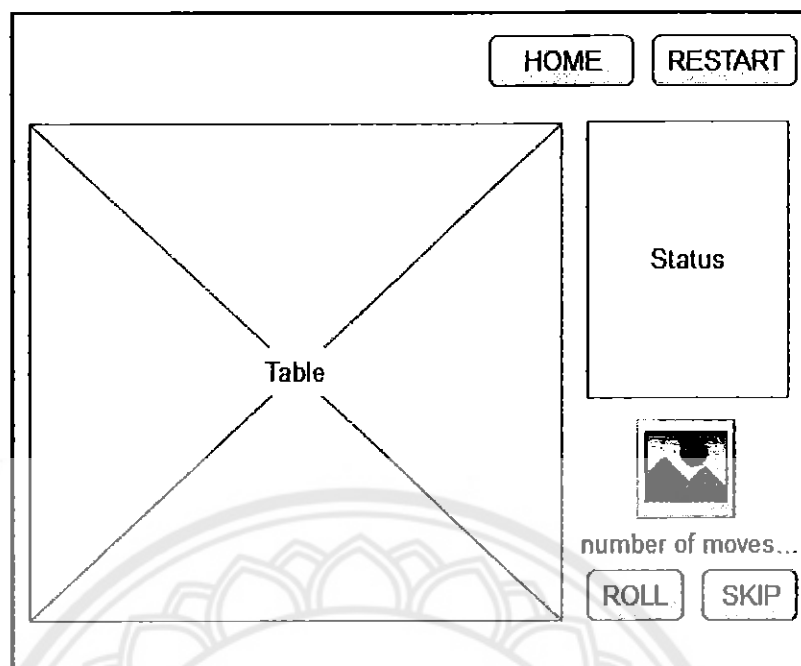
รูปที่ 3.5 หน้าจอสำหรับกรอกชื่อผู้เล่น เมื่อเล่นแบบออนไลน์

5. ออกแบบหน้าจอสำหรับกรอกชื่อผู้เล่น เมื่อคลิกปุ่ม Online หรือเมื่อคลิกปุ่ม Join จากการเลือกเล่นแบบออนไลน์ จะประกอบไปด้วย ภาพตราสัญลักษณ์ของโปรแกรม อยู่บริเวณตรงกลางด้านบนของหน้าจอ ข้อความบอกให้กรอกชื่อผู้เล่น ช่องสำหรับกรอกชื่อผู้เล่น มีทั้งหมดจำนวน 4 ช่อง อยู่บริเวณตรงกลางของหน้าจอ ปุ่ม Start อยู่บริเวณมุมล่างซ้ายของหน้าจอ เมื่อคลิกปุ่ม Start จะไปหน้าจอสำหรับการเล่นเกม Hard choices ดังรูปที่ 3.7 และปุ่ม Back อยู่บริเวณมุมบนขวาของหน้าจอ เมื่อคลิกปุ่ม Back จะกลับไปอยู่หน้าจอหลักของโปรแกรม

The screenshot shows a game interface with a 'BACK' button at the top left and a 'START' button at the bottom right. In the center, there is a banner that says 'Hard Choices'. Below the banner, the text 'Enter player name' is displayed. There are four input fields labeled 'Player 1', 'Player 2', 'Player 3', and 'Player 4' stacked vertically. A large, faint watermark of a Thai university seal is visible in the background of the screenshot.

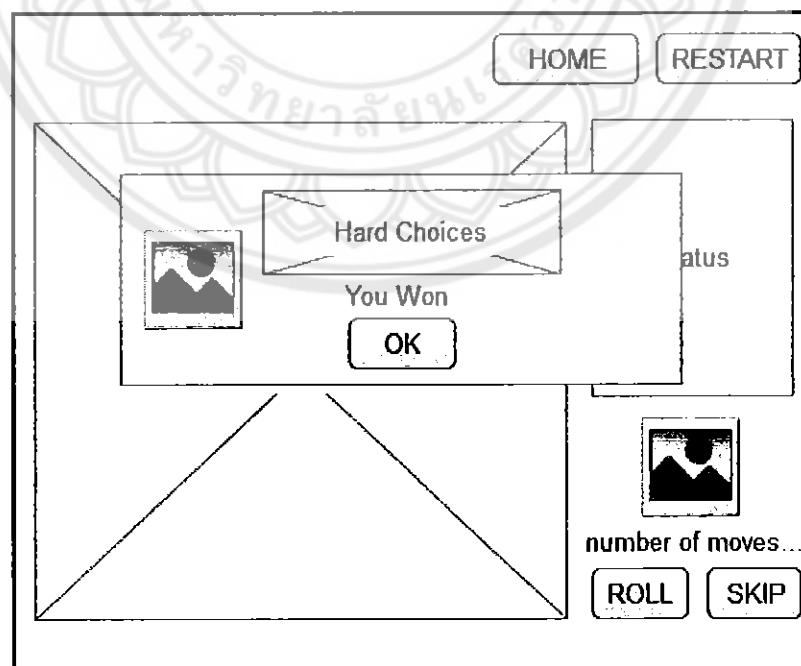
รูปที่ 3.6 หน้าจอสำหรับกรอกชื่อผู้เล่น เมื่อเล่นแบบออฟไลน์

6. ออกแบบหน้าจอสำหรับการเล่นเกม Hard choices เมื่อคลิกปุ่ม Start จะประกอบไปด้วย ตารางสำหรับทางเดินของตัวผู้เล่น อยู่บริเวณตรงกลางของหน้าจอ สถานะของตัวผู้เล่น อยู่บริเวณตรงกลางของหน้าจอถัดจากตาราง ทางเดิน รูปหน้าของลูกเต๋า อยู่บริเวณถัดลงมาด้านล่างของสถานะตัวผู้เล่น ปุ่ม Roll อยู่บริเวณมุมล่างซ้ายของหน้าจอ ปุ่ม Skip อยู่ถัดมาจากปุ่ม Roll ปุ่ม Home อยู่บริเวณมุมบนซ้ายของหน้าจอ เมื่อคลิกปุ่ม Home จะกลับไปอยู่หน้าจอหลักของโปรแกรม และปุ่ม Restart อยู่ถัดมาจากปุ่ม Home



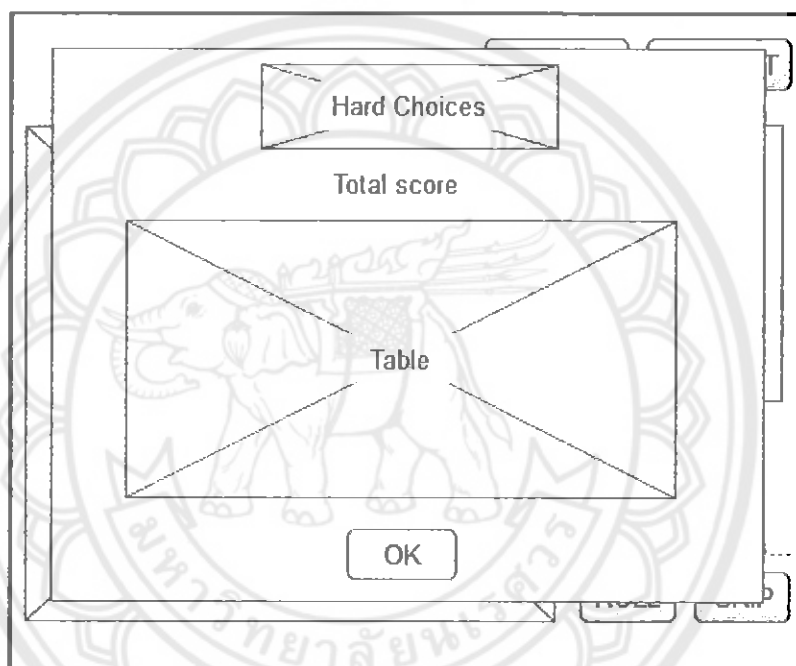
รูปที่ 3.7 หน้าจอสำหรับการเล่นเกม

7. ออกแบบหน้าจอเมื่อเข้าเส้นชัย จะประกอบไปด้วย ภาพตราสัญลักษณ์ของโปรแกรม รูปภาพถ้วยรางวัล และปุ่ม OK เมื่อคลิกปุ่ม Ok จะกลับมาอยู่ที่หน้าจอสำหรับเล่นเกม แต่ถ้าขณะนั้นมีผู้เล่นเหลืออยู่บนตารางทางเดิน 1 คน จะปรากฏหน้าจอสรุปผลการแข่งขัน ดังรูปที่ 3.9



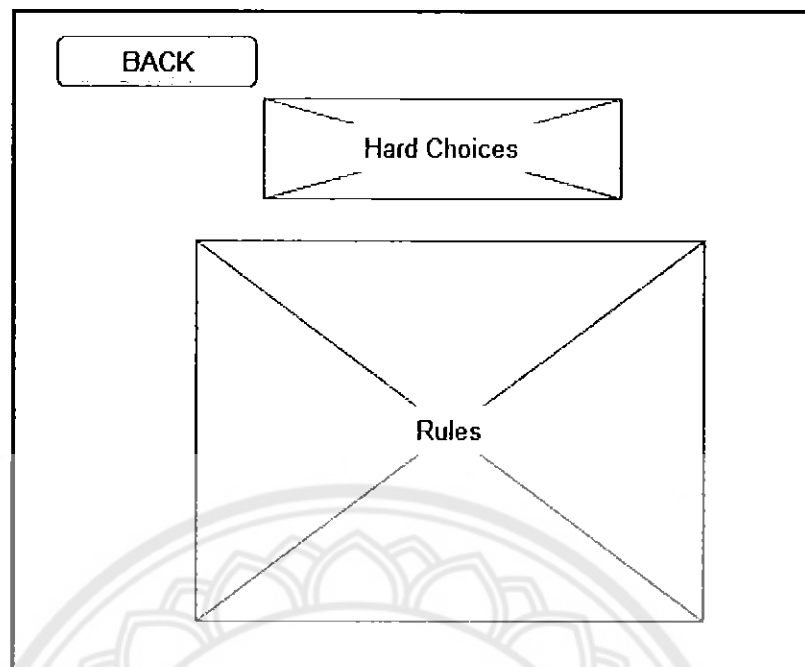
รูปที่ 3.8 หน้าจอเมื่อเข้าเส้นชัย

8. ออกแบบหน้าจอสรุปผลการแข่งขัน เมื่อคลิกปุ่ม Ok ของหน้าจอ Win ดังรูปที่ 3.8 และเมื่อผู้เล่นเหลืออยู่บนตารางทางเดิน 1 คน จะประกอบไปด้วย ภาพตราสัญลักษณ์ของโปรแกรม อยู่บริเวณตรงกลางด้านบนของหน้าจอ ตารางสรุปแสดงผลการแข่งขัน จะมีรายละเอียดดังนี้ คือ ตัวผู้เล่นที่ทำการแข่งขันทั้งหมดในรอบนั้น จำนวนการ์ดค้อนที่เก็บมาได้ของผู้เล่นแต่ละคน จำนวนการ์ดเคียวที่เก็บมาได้ของผู้เล่นแต่ละคน จำนวนการ์ดไขตวงที่เก็บมาได้ของผู้เล่นแต่ละคน จำนวนสะพานที่เดินข้ามมาของผู้เล่นแต่ละคน และคะแนนรวมของผู้เล่นแต่ละคน อยู่บริเวณตรงกลางหน้าจอ และปุ่ม OK อยู่บริเวณด้านล่างของหน้าจอ เมื่อคลิกปุ่ม Ok จะกลับมาอยู่ที่หน้าจอสำหรับเล่นเกม



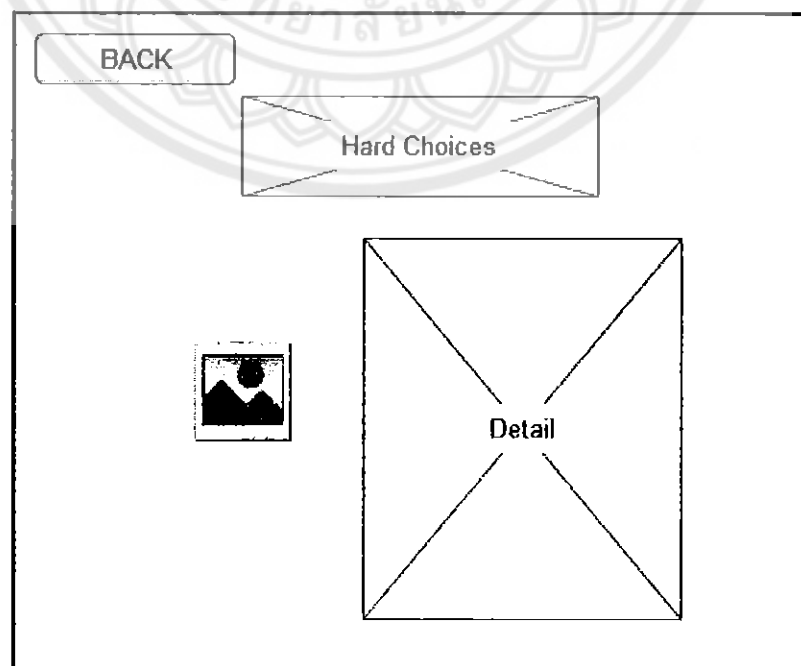
รูปที่ 3.9 หน้าจอสรุปผลการแข่งขัน

9. ออกแบบหน้าจอกฎกติกาการเล่น Hard choices เมื่อคลิกปุ่ม Rules ที่หน้าจอหลักของโปรแกรม จะประกอบไปด้วย ภาพตราสัญลักษณ์ของโปรแกรม อยู่บริเวณตรงกลางด้านบนของหน้าจอ ข้อความแสดงกฎกติกาการเล่น Hard choices อยู่บริเวณตรงกลางของหน้าจอ และปุ่ม Back อยู่บริเวณมุมบนขวาของหน้าจอ เมื่อคลิกปุ่ม Back จะกลับไปหน้าจอหลักของโปรแกรม



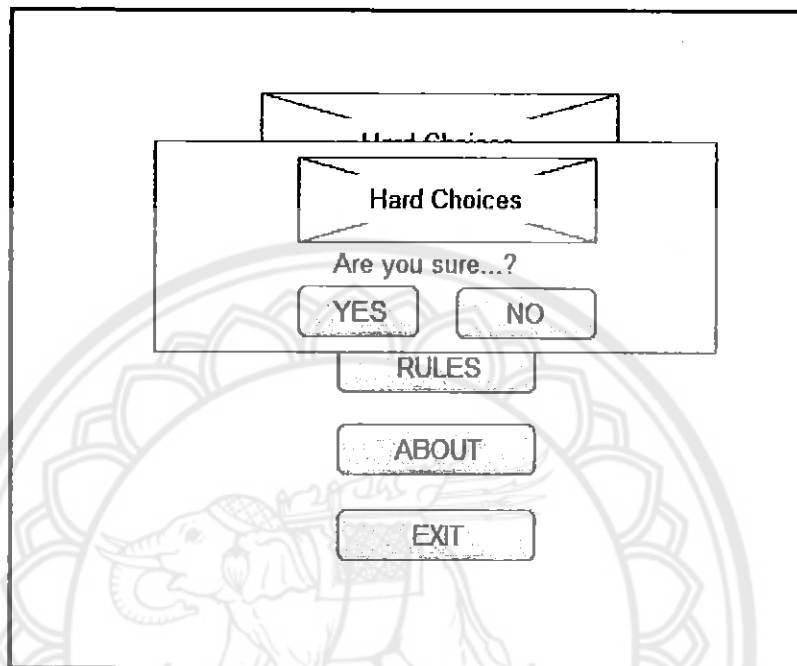
รูปที่ 3.10 หน้าจอถูกติกาการเล่น Hard choices

10. ออกแบบหน้าจอเกี่ยวกับผู้จัดทำเกม Hard choices เมื่อคลิกปุ่ม About ที่หน้าจอหลักของโปรแกรม จะประกอบไปด้วย ภาพตราสัญลักษณ์ของโปรแกรม อยู่บริเวณตรงกลางด้านบนของหน้าจอ รูปของผู้จัดทำ อยู่บริเวณตรงกลางของหน้าจอ ข้อความแสดงประวัติของผู้จัดทำเกม Hard choices อยู่บริเวณตรงกลางของหน้าจอถัดจากรูปของผู้จัดทำ และปุ่ม Back อยู่บริเวณมุมบนขวาของหน้าจอ เมื่อคลิกปุ่ม Back จะกลับไปหน้าจอหลักของโปรแกรม



รูปที่ 3.11 หน้าจอเกี่ยวกับผู้จัดทำโปรแกรมเกม Hard choices

11. ออกแบบหน้าจอก่อนออกจากเกม Hard choices เมื่อกดปุ่ม Exit ที่หน้าจอหลักของโปรแกรม จะประกอบไปด้วย ภาพตราสัญลักษณ์ของโปรแกรม ข้อความถามว่าแน่ใจจะออกจากโปรแกรมหรือไม่ ปุ่ม Yes และปุ่ม No อยู่ถัดจากปุ่ม Yes เมื่อกดปุ่ม No จะกลับไปหน้าจอหลังของโปรแกรมอีกครั้ง และเมื่อกดปุ่ม Yes จะทำการออกจากโปรแกรม



รูปที่ 3.12 หน้าจอก่อนออกจากโปรแกรมเกม Hard choices

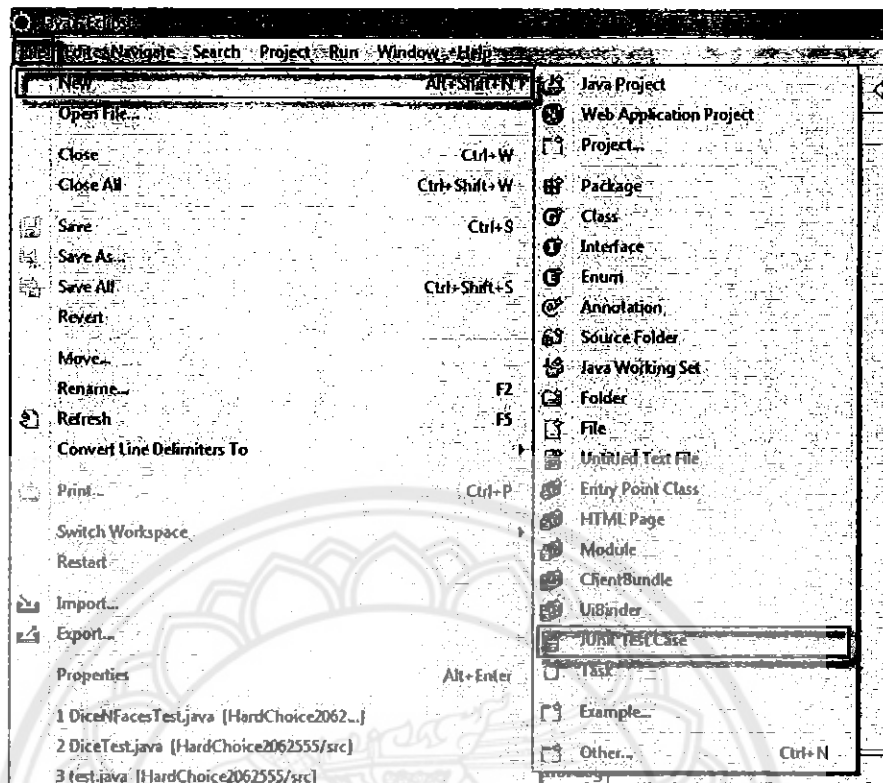
3.3 การทดสอบโปรแกรม

การทดสอบโปรแกรมเป็นกระบวนการที่สำคัญอย่างมาก เพื่อช่วยให้ซอฟต์แวร์ที่กำลังพัฒนาอยู่นั้นมีความถูกต้อง และมีคุณภาพมากที่สุด ค่าเคินการพัฒนาซอฟต์แวร์อย่างมีประสิทธิภาพ เนื่องจากโปรแกรมนี้ใช้เครื่องมือในการพัฒนาโปรแกรมคือ Eclipse IDE for Java EE Developers จึงเลือกใช้ JUnit เพื่อทำการทดสอบโปรแกรมนี้

3.3.1 วิธีการทดสอบโปรแกรม

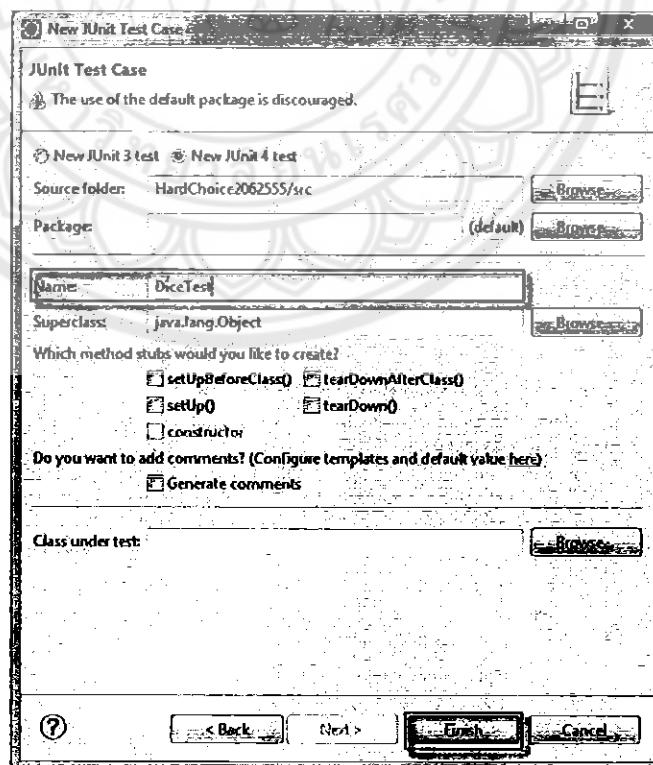
1. สร้างคลาสที่เอาไว้สำหรับเขียนการทดสอบฟังก์ชัน คลิก File -> New -> JUnit Test

Case



รูปที่ 3.13 ขั้นตอนการสร้างคลาส JUnit Test

2. ใส่ชื่อคลาส หลังจากนั้นคลิกปุ่ม Finish



รูปที่ 3.14 ตั้งชื่อให้กับคลาส JUnit Test

3. เขียนการทดสอบฟังก์ชัน สิ่งสำคัญในการเขียนการทดสอบคือ ต้องขึ้นต้นด้วย @Test ก่อนทุกครั้ง

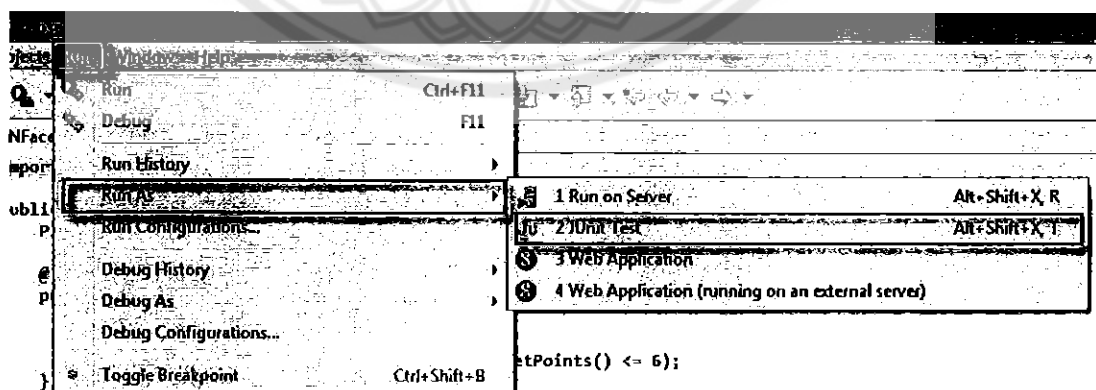
```

10 import static org.junit.Assert.*;
4
5 public class DiceNFacesTest {
6     Player aPlayer = new Player();
7
8     @Test
9     public void dicePointBetween1And6() {
10        Dice aDice = new DiceNFaces(6);
11        aDice.roll(aPlayer);
12        assertTrue(1 <= aDice.getPoints() && aDice.getPoints() <= 6);
13
14
15    @Test
16    public void test60Rolls(){
17        Dice aDice = new DiceNFaces(6);
18        int points[] = new int[DiceNFaces.nFaces + 1];
19        // Initialize values in the array
20        for (int i = 0; i < points.length; i++) {
21            points[i] = 0;
22        }
23        // roll and count
24        for (int i = 0; i < 6000; i++) {
25            aDice.roll(aPlayer);
26            points[aDice.getPoints()]++;
27        }
28        // Check count should be +/- 25%
29        for (int i = 1; i <= DiceNFaces.nFaces; i++) {
30            System.out.println(i + " : " + points[i]);
31            assertTrue(700 <= points[i] && points[i] <= 1300);
32        }
33        System.out.println(aDice.getPoints());
34    }

```

รูปที่ 3.15 เขียนการทดสอบฟังก์ชัน

4. ทำการทดสอบโดยการคลิก Run -> Run As -> JUnit Test



รูปที่ 3.16 ทำการทดสอบ

5. ผลการทดสอบ จะแสดงรายละเอียด จำนวนที่ทดสอบ และจำนวนข้อผิดพลาดที่พบ

```

10 import static org.junit.Assert.*;
11
12 public class DiceNFacesTest {
13     Player aPlayer = new Player();
14
15     @Test
16     public void dicePointBetween1And6() {
17         Dice aDice = new DiceNFaces(6);
18         aDice.roll(aPlayer);
19         assertEquals("1 <= aDice.getPoints() <= 6", aDice.getPoints(), 1);
20     }
21
22     @Test
23     public void test60Rolls(){
24         Dice aDice = new DiceNFaces(6);
25         int points[] = new int[DiceNFaces.nFaces + 1];
26         // Initialize values in the array
27         for (int i = 0; i < points.length; i++) {
28             points[i] = 0;
29         }
30         // roll and count
31         for (int i = 0; i < 6000; i++) {
32             aDice.roll(aPlayer);
33             points[aDice.getPoints()]++;
34         }
35         // Check count should be +/- 25%
36         for (int i = 1; i <= DiceNFaces.nFaces; i++) {
37             System.out.println(i + " : " + points[i]);
38         }
39     }
40 }

```

รูปที่ 3.17 พบข้อผิดพลาดจากการทดสอบ

จากรูปที่ 3.17 จะเห็นว่าผลการทดสอบพบข้อผิดพลาดเกิดขึ้น จะแสดงรายละเอียด จำนวนข้อผิดพลาดที่พบ และสามารถบอกได้ด้วยว่าพบข้อผิดพลาดตรงไหน ทำให้สามารถกลับไปแก้ไขให้ทำการทดสอบผ่านได้ ในที่นี้พบข้อผิดพลาดเกิดขึ้นในบรรทัดที่ 12

6. ผลการทดสอบที่สมบูรณ์ ไม่พบข้อผิดพลาดจะเป็นไปดังรูปที่ 3.18

```

10 import static org.junit.Assert.*;
11
12 public class DiceNFacesTest {
13     Player aPlayer = new Player();
14
15     @Test
16     public void dicePointBetween1And6() {
17         Dice aDice = new DiceNFaces(6);
18         aDice.roll(aPlayer);
19         assertTrue("1 <= aDice.getPoints() && aDice.getPoints() <= 6");
20     }
21
22     @Test
23     public void test60Rolls(){
24         Dice aDice = new DiceNFaces(6);
25         int points[] = new int[DiceNFaces.nFaces + 1];
26         // Initialize values in the array
27         for (int i = 0; i < points.length; i++) {
28             points[i] = 0;
29         }
30     }
31 }

```

ที่ 3.18 ผลการทดสอบผ่าน

บทที่ 4

ผลการทดลอง

จากการดำเนินโครงการตามบทที่ 3 ทำให้ได้เกม Hard choices รุ่นเดสก์ทอปแบบผู้เล่นหลายคน เพื่อใช้ในการเรียนรู้ผลกระทบจากการตัดสินใจระหว่างการพัฒนาซอฟต์แวร์ ในการทดลองจะแบ่งเป็น 3 การทดลอง คือ

1. การทดลองแบบแข่งขันภายในเครื่องเดียวกัน
2. การทดลองแบบแข่งขันผ่านระบบเครือข่ายอินเทอร์เน็ต
3. การทดลองการเก็บสถิติข้อมูลของผู้เล่น

4.1 การทดลองแบบแข่งขันภายในเครื่องเดียวกัน

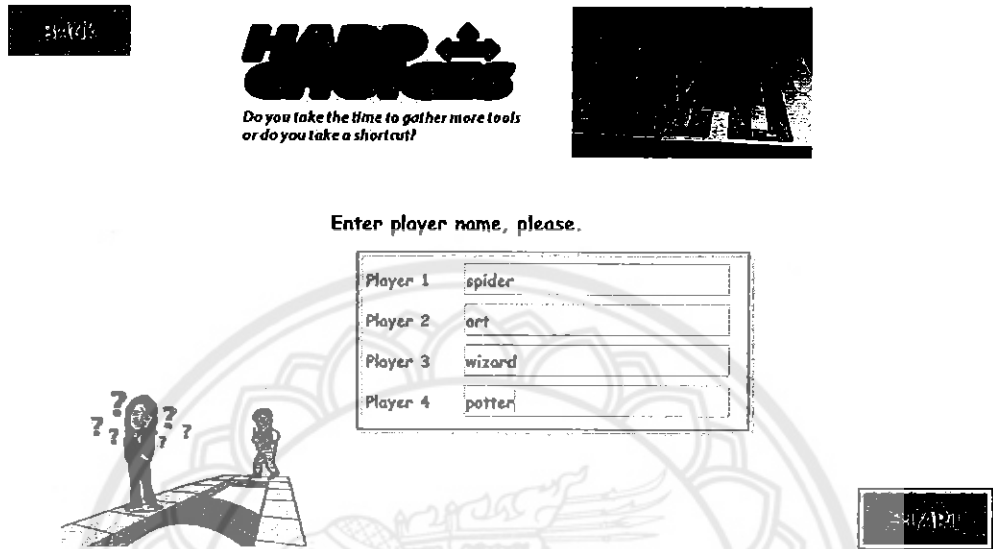
ขั้นตอนการทดลอง โปรแกรมภายในเครื่องเดียวกันมีดังนี้

1. เมื่อเข้าเกมมาแล้ว คลิกปุ่ม Play เลือกเล่นเกมแบบออฟไลน์ โดยการคลิกปุ่ม Offline



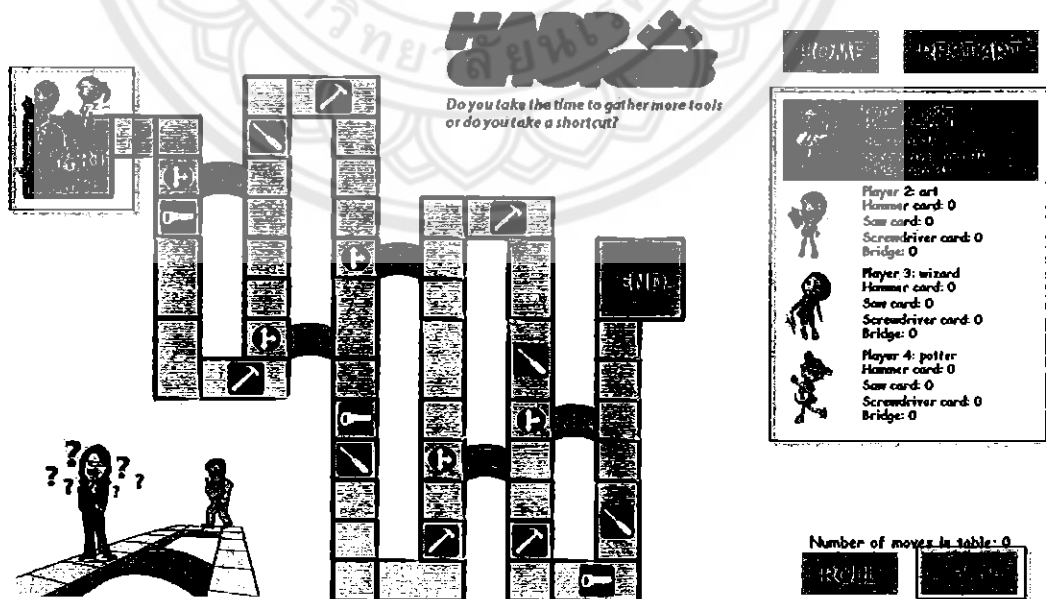
รูปที่ 4.1 เลือกเล่นเกมแบบออฟไลน์

2. เมื่อเข้าสู่หน้าจอกรอกชื่อตัวผู้เล่น ทำการกรอกชื่อผู้เล่นลงในช่องกรอกชื่อ สามารถเล่นได้ตั้งแต่ 2 ถึง 4 คน ต้องกรอกชื่อได้ลงมาตั้งแต่ Player 1 ไม่สามารถข้ามมากรอกชื่อ Player 2 ได้ หลังกรอกชื่อผู้เล่นเสร็จ คลิกปุ่ม Start เพื่อเข้าเล่นเกม



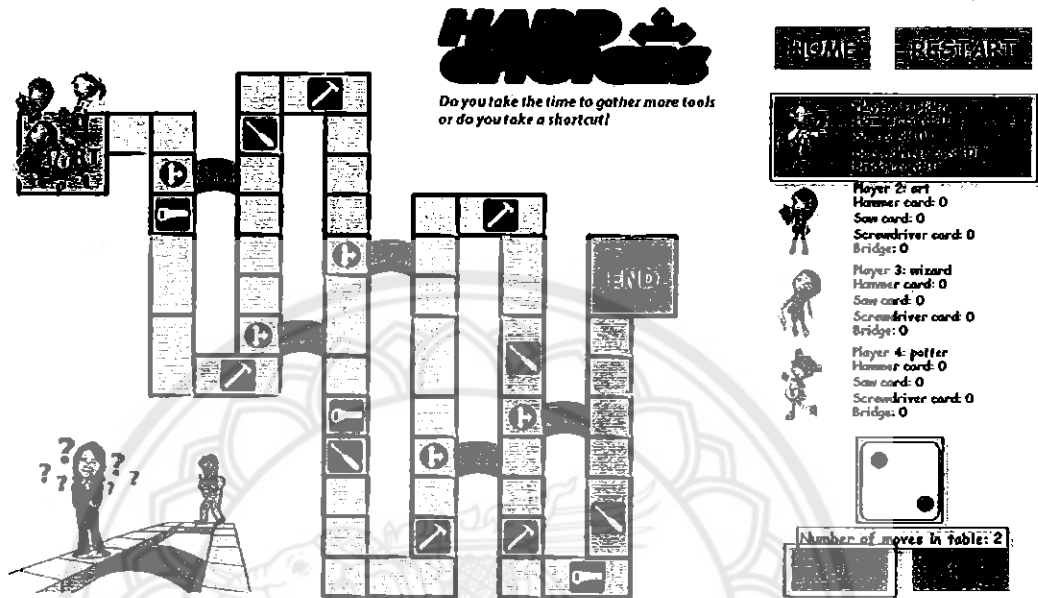
รูปที่ 4.2 หน้าจอกรอกชื่อตัวผู้เล่น

3. เมื่อเข้าสู่หน้าจอเกม ตัวผู้เล่นจะอยู่ที่จุด Start การ์ดเครื่องมือต่างๆ การ์ดสะพาน ของผู้เล่นแต่ละคนจะเริ่มต้นเป็น 0 ปุ่ม Skip จะไม่สามารถคลิกได้ถ้าผู้เล่นแต่ละคนยังไม่มีการข้ามสะพาน



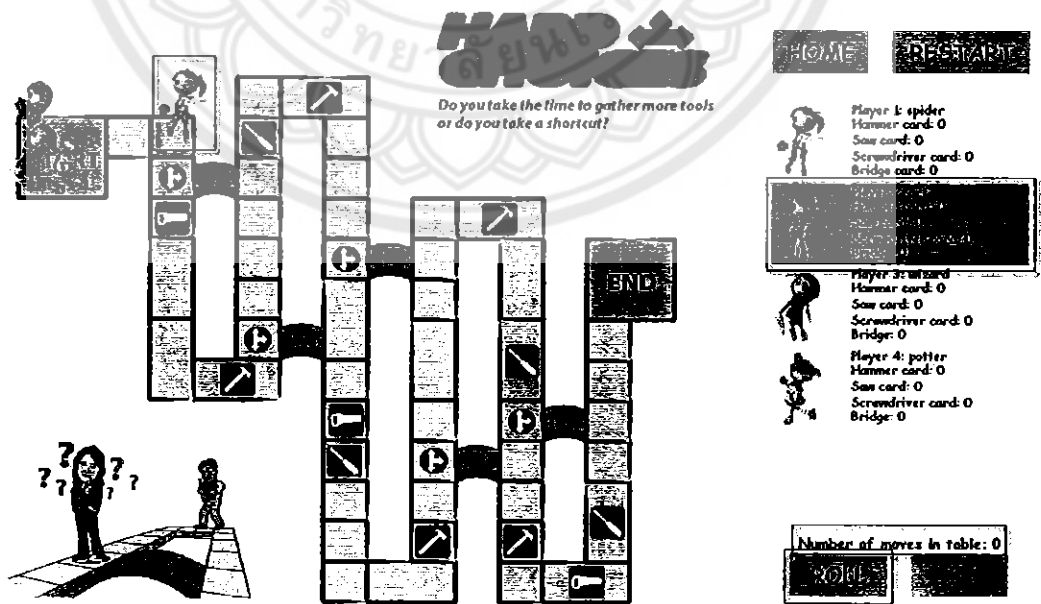
รูปที่ 4.3 หน้าจอเกม

4. คนที่จะสามารถทอยลูกเต๋านั้นได้จะมีแถบสีคลุมที่สถานะของผู้เล่นนั้น เมื่อทอยลูกเต๋ได้แล้ว จะมีหน้าลูกเต๋าส่งขึ้นมาบอกจำนวนแต้มที่ทอยได้ และมีข้อความบอกจำนวนช่องที่สามารถเดินได้ ปุ่มลูกเต๋าก็จะไม่สามารถคลิกซ้ำได้ จนกว่าจะเดินครบจำนวนช่อง



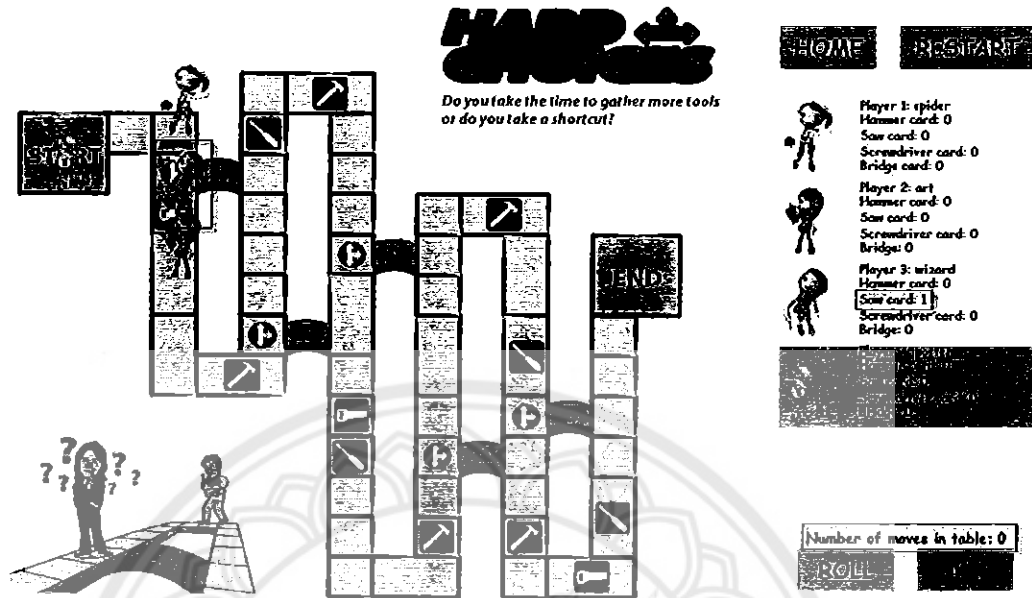
รูปที่ 4.4 การทอยลูกเต๋่า

5. หลังจากผู้เล่นคนแรกทำการเดินเสร็จ ผู้เล่นคนต่อไปจะเป็นคนเล่นต่อ ปุ่ม Roll จะสามารถคลิกได้อีกครั้ง และแถบสีจะย้ายมาคลุมที่สถานะของผู้ที่จะเล่นในตาถัดไป



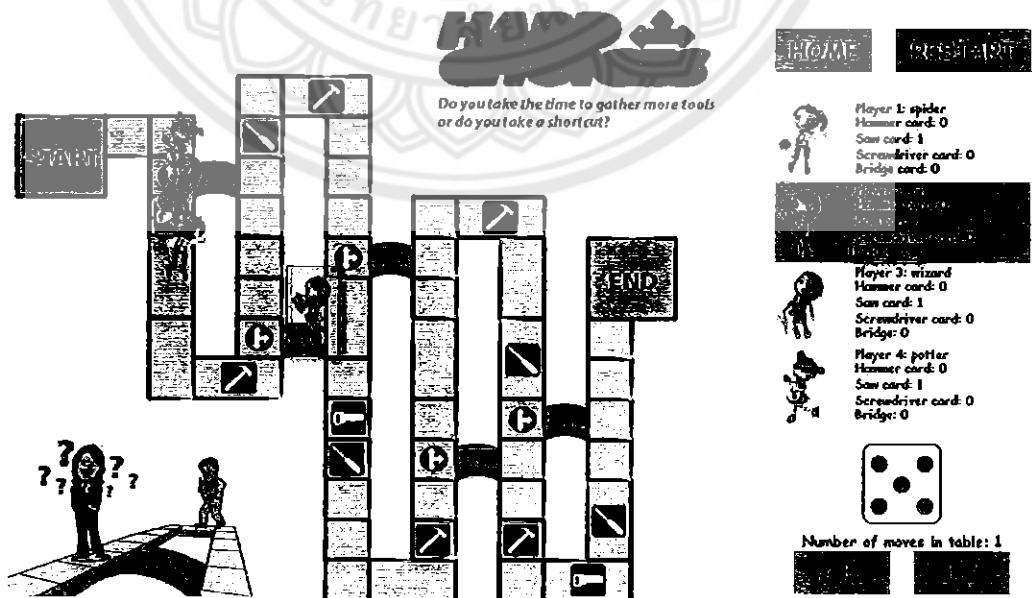
รูปที่ 4.5 เมื่อผู้เล่นคนแรกเล่นจบ

6. เมื่อผู้เล่นเดินอยู่ที่ช่องที่มีการ์ดเครื่องมือ และเดินเป็นช่องสุดท้ายพอดี จะได้รับการ์ดเครื่องมือนั้น

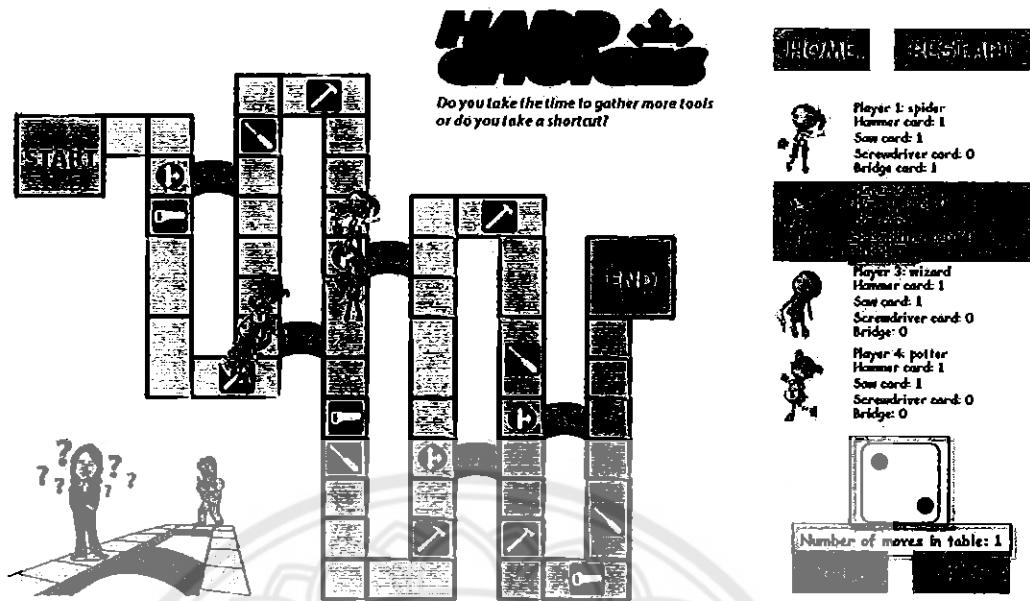


รูปที่ 4.6 การเก็บการ์ดเครื่องมือ

7. เมื่อผู้เล่นเดินข้ามสะพาน จะต้องเก็บการ์ดสะพานไว้ด้วย การที่ผู้เล่นเก็บการ์ดสะพานไว้ จะทำให้การทอยลูกเต๋าค้างครั้งต่อไปจะนำค่าจากจำนวนที่ถือการ์ดสะพานอยู่ ไปลบออกจากจำนวนหน้าลูกเต๋า เช่น ทอยลูกเต๋าค้างได้ 2 แต้ม แต่มีการ์ดสะพานอยู่ 1 ใบ ทำให้จำนวนครั้งที่สามารถเดินได้ในตาครั้งนั้นคือ 1 ช่อง ดังรูปที่ 4.8

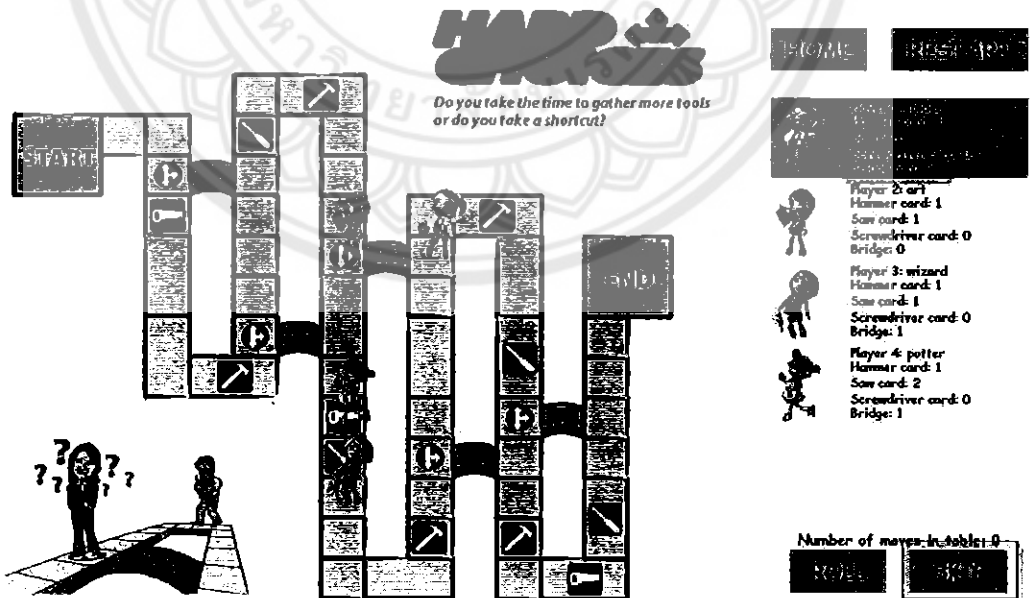


รูปที่ 4.7 การเดินข้ามสะพาน

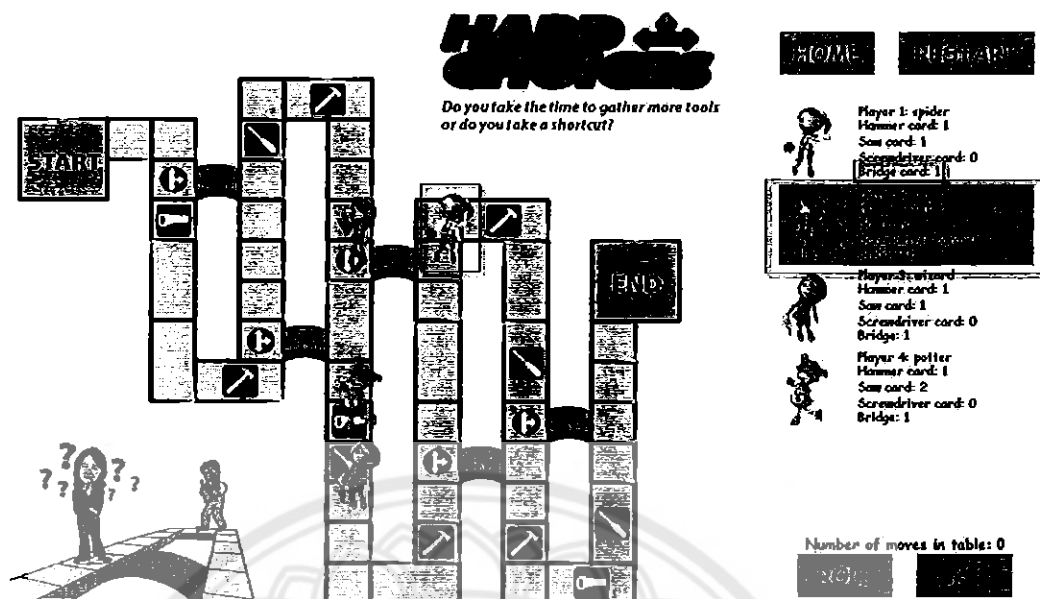


รูปที่ 4.8 จำนวนการ์ดสะพานที่อยู่ในมือ ลบแต้มลูกเต๋า

8. ผู้เล่นที่มีการ์ดสะพาน เมื่อถึงตาตัวเอง ปุ่ม Skip จะสามารถคลิกได้ และเมื่อคลิกปุ่ม Skip การ์ดสะพานจะหายไป ผู้เล่นจะไม่สามารถเดินได้ และผู้เล่นคนถัดไปจะเป็นคนเล่นคนต่อไป เช่น ผู้เล่นคนที่หนึ่งมีการ์ดสะพานอยู่ในมือ 2 ใบ ทำการคลิกปุ่ม Skip ทำให้ต้องทิ้งการ์ดสะพานไป 1 ใบ ดังนั้นจึงเหลือการ์ดสะพานอยู่ในมือ 1 ใบ ดังรูปที่ 4.10

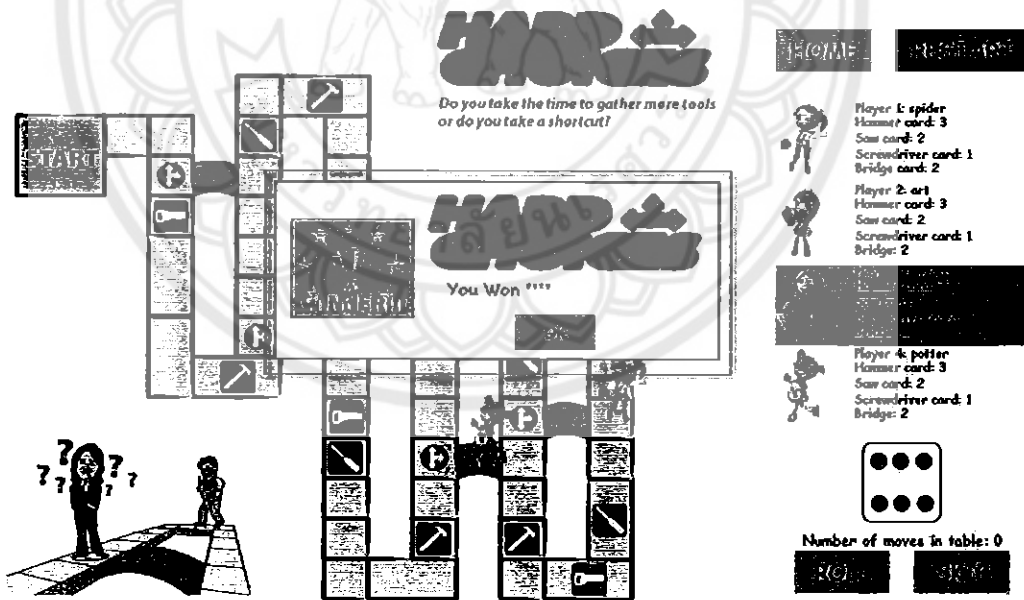


รูปที่ 4.9 เมื่อมีการ์ดสะพานอยู่ในมือสามารถคลิกปุ่ม Skip ได้



รูปที่ 4.10 หลังจากคลิกปุ่ม Skip

9. เมื่อมีผู้เล่นเดินเข้าเส้นชัย จะมีข้อความขึ้นมาบอกว่าคุณชนะ ผู้ที่เข้าเส้นชัยคนแรก จะได้แต้มจากการเข้าเส้นชัย 5 คะแนน เข้าเส้นชัยคนที่สองได้ 3 คะแนน และเข้าคนที่สามได้ 1 คะแนน



รูปที่ 4.11 เมื่อมีผู้เล่นเข้าเส้นชัย

10. เมื่อมีผู้เล่นเหลืออยู่บนตารางทางเดินคนเดียว เกมจะจบลงทันที และมีการสรุปผลการแข่งขันหาผู้ชนะในรอบนั้น โดยจะมีรายละเอียดแสดงไว้ดังนี้ จำนวนการ์ดค้อน จำนวนการ์ดเลื่อย จำนวนการ์ดไขควง จำนวนการ์ดสะพาน และสรุปคะแนนรวม ผู้ที่มีคะแนนรวมมากที่สุดจะเป็นผู้ชนะ

The screenshot shows a game interface with a 'Game Result' table and player statistics. The table has columns for 'Players', 'Hammer card', 'Saw card', 'Screwdriver card', 'Bridge card', and 'Total scores'. The 'Total scores' column has a 'WIN' label next to the value 16. To the right of the table, there are player statistics for four players: Player 1 (spider), Player 2 (art), Player 3 (potter), and Player 4 (potter). Each player's stats include Hammer card, Saw card, Screwdriver card, and Bridge card counts. There is also a 'Number of moves in table: 0' indicator and a 'STOP' button.

Players	Hammer card	Saw card	Screwdriver card	Bridge card	Total scores
	5	2	4	0	16 WIN
	5	3	4	3	13
	5	3	4	1	12
	5	3	4	3	15

Player 1: spider
 Hammer card: 5
 Saw card: 2
 Screwdriver card: 4
 Bridge card: 0

Player 2: art
 Hammer card: 5
 Saw card: 3
 Screwdriver card: 4
 Bridge: 3

Player 3: potter
 Hammer card: 5
 Saw card: 3
 Screwdriver card: 4
 Bridge: 3

Player 4: potter
 Hammer card: 5
 Saw card: 3
 Screwdriver card: 4
 Bridge: 3

Number of moves in table: 0

รูปที่ 4.12 การสรุปผลการแข่งขัน

สำหรับการข้ามสะพานในเกมนี้ เหมือนกับการเขียนโปรแกรมแบบ Hard code เมื่อต้องกลับมาปรับปรุงหรือเพิ่มฟีเจอร์เข้ามาใหม่จะทำได้ยาก ใช้เวลานาน และเกิดปัญหาได้ง่ายกว่าปกติ เหมือนกับในเกมคือ เมื่อทอยลูกเต๋าในครั้งถัดไปจะนำการ์ดสะพานมาลบออกจากค่าลูกเต๋าทิ้งที่ทอยได้ ทำให้เดินได้น้อยลง เป็นอย่างนี้ไปทุกๆตา มีวิธีการแก้ไขโดยการกลับมาเขียนโปรแกรมใหม่แทนที่จะแทนค่าลงไปตรงๆ ไม่ผ่านตัวแปรก็นำตัวแปรเข้ามาช่วยในการแทนค่า เมื่อเวลากลับมาแก้ไขจะทำให้ง่ายและเร็วขึ้น เหมือนกับการที่เราคลิกปุ่ม skip จะทำให้การ์ดสะพานหายไป 1 ใบ แต่เราก็จะไม่ได้เดินตานั้น ถึงอย่างไรก็ดีการข้ามสะพานก็ยังพอมีข้อดีอยู่เหมือนกัน ซึ่งก่อนที่จะข้ามสะพานผู้เล่นจะต้องไตร่ตรองดูแล้วว่าการข้ามสะพานครั้งนี้คุ้มที่จะแลก


```

public void homePanel() {
    playerPanel();
    scorePanel();
    aboutPanel();
    exitPanel();
    onOfflinePanel();

    frameHome = new JFrame();
    createFrame(frameHome);

    //play button
    play = new JButton("PLAY");
    play.setBounds(599, 299, 153, 63);
    createButton(play);
    ButtonAction playAction = new ButtonAction();
    play.addActionListener(playAction);
    frameHome.getContentPane().add(BorderLayout.CENTER, play);

    //rules button
    rules = new JButton("RULES");
    rules.setBounds(599, 399, 153, 63);
    createButton(rules);
    ButtonAction rulesAction = new ButtonAction();
    rules.addActionListener(rulesAction);
    frameHome.getContentPane().add(BorderLayout.CENTER, rules);

    //about button
    about = new JButton("ABOUT");
    about.setBounds(599, 499, 153, 63);
    createButton(about);
    ButtonAction aboutAction = new ButtonAction();
    about.addActionListener(aboutAction);
    frameHome.getContentPane().add(BorderLayout.CENTER, about);
}

```

รูปที่ 4.13 ตัวอย่างการเขียนโปรแกรมแบบ Hard code

```

public void homePanel() {
    playerPanel();
    scorePanel();
    aboutPanel();
    exitPanel();
    onOfflinePanel();

    frameHome = new JFrame();
    createFrame(frameHome);

    //play button
    play = new JButton("PLAY");
    play.setBounds(horzButtonPosition, vertButtonPosition, horzButtonSize, vertButtonSize);
    createButton(play);
    ButtonAction playAction = new ButtonAction();
    play.addActionListener(playAction);
    frameHome.getContentPane().add(BorderLayout.CENTER, play);

    //rules button
    rules = new JButton("RULES");
    rules.setBounds(horzButtonPosition, vertButtonPosition + vertButtonDistance, horzButtonSize, vertButtonSize);
    createButton(rules);
    ButtonAction rulesAction = new ButtonAction();
    rules.addActionListener(rulesAction);
    frameHome.getContentPane().add(BorderLayout.CENTER, rules);

    //about button
    about = new JButton("ABOUT");
    about.setBounds(horzButtonPosition, vertButtonPosition + (vertButtonDistance * 2), horzButtonSize, vertButtonSize);
    createButton(about);
    ButtonAction aboutAction = new ButtonAction();
    about.addActionListener(aboutAction);
    frameHome.getContentPane().add(BorderLayout.CENTER, about);
}

```

รูปที่ 4.14 ตัวอย่างการเขียนโปรแกรมโดยใช้ตัวแปร

4.2 การทดลองโปรแกรมผ่านระบบอินเทอร์เน็ต

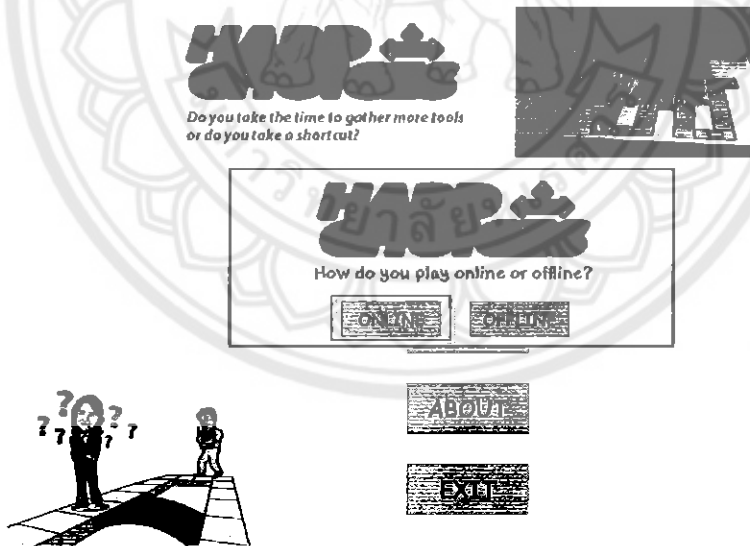
ในที่นี้จะกล่าวถึงการเปิดเครื่องบริการ (Server) และการเข้าเล่น แต่จะไม่กล่าวถึงการเล่น เพราะสามารถทำการเล่นได้เหมือนกับการเล่นภายในเครื่องเดียวกัน มีขั้นตอนการทดลองดังนี้

1. ก่อนที่จะเล่นผ่านระบบอินเทอร์เน็ตนั้น ต้องทำการเปิดเครื่องบริการก่อนทุกครั้ง



รูปที่ 4.15 เปิดเครื่องบริการ (server)

2. เมื่อเข้าเกมเลือกเล่นเกมแบบออนไลน์ โดยการคลิกปุ่ม Online

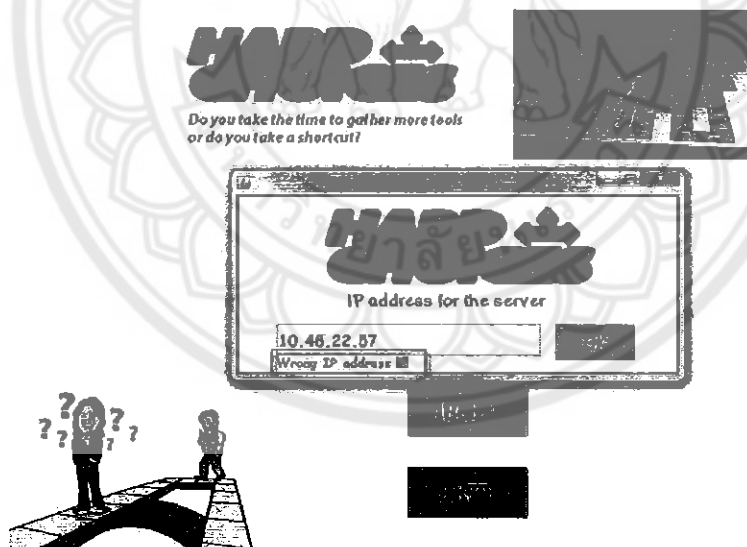


รูปที่ 4.16 เลือกเล่นเกมแบบออนไลน์

3. หลังจากคลิกปุ่ม Online เข้ามาแล้ว กรอกเลขที่อยู่ไอพี (IP address) ของเครื่องบริการ และคลิกปุ่ม Ok หลังจากคลิกปุ่ม Ok ถ้าเลขที่อยู่ไอพีถูกจะปรากฏหน้าต่างให้กรอกชื่อเข้าเล่น ดังรูปที่ 4.19 แต่ถ้ากรอกกรอกเลขที่อยู่ไอพีผิด จะมีข้อความขึ้นมาบอกว่า ที่อยู่ไอพีผิด ดังรูปที่ 4.18



รูปที่ 4.17 การกรอกเลขที่อยู่ไอพีของเครื่องบริการ

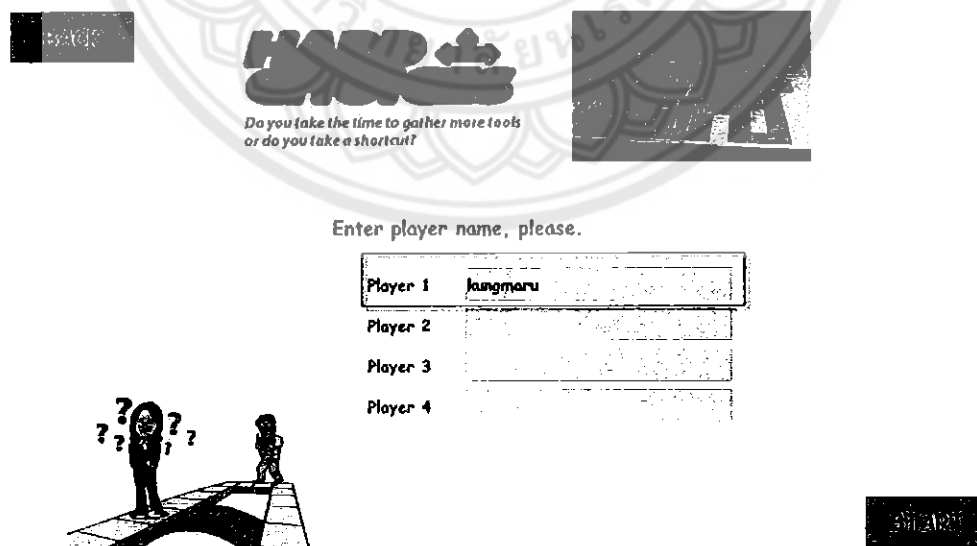


รูปที่ 4.18 ข้อความแจ้งเตือนเมื่อกรอกเลขที่อยู่ไอพีของเครื่องบริการผิด

4. หลังจากคลิกปุ่ม Ok ที่หน้าต่างกรอกเลขที่อยู่ไอพีของเครื่องบริการเข้ามาแล้ว ทำการกรอกชื่อผู้เล่น และคลิกปุ่ม Join หลังจากคลิกปุ่ม Join คนที่เข้ามาคนแรกจะเป็น Player 1 แล้วรอให้มีคนเข้ามาเล่นด้วย ดังรูปที่ 4.20



รูปที่ 4.19 การกรอกชื่อตัวผู้เล่น



รูปที่ 4.20 หลังจากกรอกชื่อตัวผู้เล่นเสร็จ

5. เมื่อมีผู้เล่นคนที่สองเข้ามาจะเป็น Player 2 และสามารถคลิกปุ่ม Start เพื่อเข้าเล่นเกมได้



รูปที่ 4.21 เมื่อมีผู้เล่นคนที่สองเข้ามา

4.2.1 วิธีการส่งข้อมูลสื่อสารกันระหว่างผู้เล่นกับเครื่องบริการ

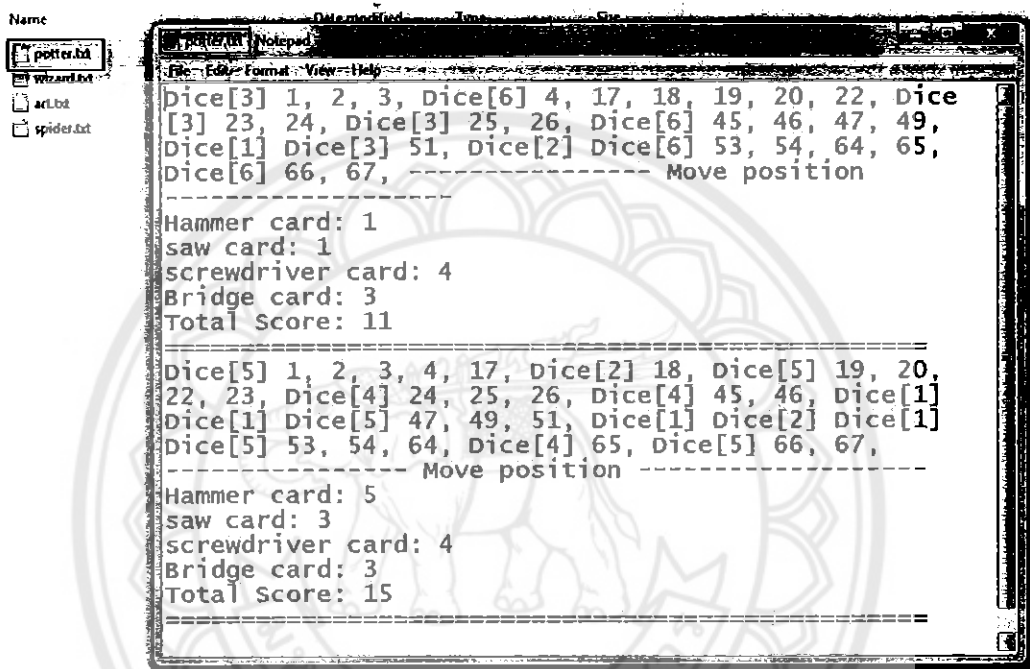
การส่งข้อมูล ผู้เล่นจะส่งข้อมูลให้กับเครื่องบริการ เครื่องบริการจะอ่านค่าที่รับมาจากผู้เล่น แล้วทำการส่งกลับให้ผู้เล่นทุกคนที่เข้ามาเล่นในรอบนั้น ยกตัวอย่างการส่งข้อมูลที่ผู้เล่นส่งให้กับเครื่องบริการ

"\$" + Integer.toString(moveX) + "-" + Integer.toString(moveY) + "%" + turnP + "N" + Integer.toString(gameNoPlay)

สำหรับตัวอย่างนี้คือการส่งค่าให้ผู้เล่นย้ายตำแหน่ง โดยการส่งค่า moveX คือตำแหน่ง x ที่จะย้ายไป ค่า moveY คือตำแหน่ง y ที่จะย้ายไป ส่งค่า turnP เพื่อบอกว่าผู้เล่นคนใดเป็นผู้เล่นอยู่ และค่า gameNoPlay เพื่อบอกว่าอยู่กลุ่มการเล่นที่เท่าไร แต่ละกลุ่มสามารถมีผู้เล่นได้สูงสุด 4 คน รูปแบบการส่งข้อมูลจะเป็นการส่งค่าในรูปแบบ String

4.3 การเก็บสถิติข้อมูลของผู้เล่น

หลังจากที่การแข่งขันจบลงจะมีการบันทึกข้อมูลต่างๆลงเพิ่มความ จะเก็บสถิติของผู้เล่นแต่ละคน ประกอบไปด้วย แด้มลูกเต๋าที่ทอย ตำแหน่งการเดินของผู้เล่น จำนวนการ์ดค้อนที่เก็บได้ จำนวนการ์ดเลื่อยที่เก็บได้ จำนวนการ์ดไขควงที่เก็บได้ จำนวนสะพานที่เดินข้าม และคะแนนรวมทั้งหมดที่ได้ เพื่อเก็บไว้นำไปใช้ในการวิเคราะห์ต่อไป



รูปที่ 4.22 การเก็บสถิติ

การบันทึกจะบันทึกตามชื่อผู้เล่นที่เข้ามาเล่น ถ้าเป็นคนเดียวกันเข้ามาเล่นซ้ำจะบันทึกสถิติไว้ในไฟล์เดียวกัน แต่ถ้าคนที่เข้ามาเล่นครั้งแรกจะสร้างไฟล์เก็บสถิติใหม่เป็นของผู้เล่นคนนั้นที่เข้ามาเล่น

4.4 สรุปผลการทดลอง

ตารางที่ 4.1 สรุปผลการทดลองแบบออฟไลน์

การทดลอง	ผลการทดลอง
1. เข้าเล่นออฟไลน์	✓
2. กรอกชื่อผู้เล่น แล้วเข้าเล่น	✓
3. ทอยลูกเต๋า แล้วแสดงหน้าลูกเต๋า	✓
4. ทอยลูกเต๋า แล้วแสดงจำนวนช่องที่สามารถเดินได้	✓
5. คลิกช่องบนตาราง ผู้เล่นเดิน ไปตำแหน่งนั้น	✓
6. แลบลีเปลี่ยนเมื่อถึงรอบผู้เล่นนั้น	✓
7. ผู้เล่นเดินลงช่องสุดท้ายที่ช่องการ์ดค้อน แล้วเก็บการ์ดค้อน	✓
8. ผู้เล่นเดินลงช่องสุดท้ายที่ช่องการ์ดเลื่อย แล้วเก็บการ์ดเลื่อย	✓
9. ผู้เล่นเดินลงช่องสุดท้ายที่ช่องการ์ดไขควง แล้วเก็บการ์ดไขควง	✓
10. ผู้เล่นเดินข้ามสะพาน แล้วเก็บการ์ดสะพาน	✓
11. ผู้เล่นมีการ์ดสะพาน การทอยลูกเต๋าค้างต่อไป นำจำนวนการ์ดสะพานไปลบออกจากจำนวนหน้าของลูกเต๋า	✓
12. ผู้เล่นมีการ์ดสะพาน รอบต่อไปจะสามารถคลิกปุ่ม skip ได้	✓
13. คลิกปุ่ม skip แล้วการ์ดสะพานหายไป 1 ใบ	✓
14. ผู้เล่นเดินเข้าเส้นชัย มีข้อความขึ้นมาบอกว่าคุณชนะ	✓
15. ผู้เล่นเดินเข้าเส้นชัย คนแรกได้ 5 คะแนน	✓
16. ผู้เล่นเดินเข้าเส้นชัย คนที่สองได้ 3 คะแนน	✓
17. ผู้เล่นเดินเข้าเส้นชัย คนที่สามได้ 1 คะแนน	✓
18. เมื่อผู้เล่นเดินเข้าเส้นชัย แล้วมีผู้เล่นเหลืออยู่บนตาราง 1 คน เมื่อคลิกปุ่ม OK ที่หน้าต่างที่แสดงข้อความว่าคุณชนะ จะมีหน้าต่างสรุปผลการแข่งขันแสดงขึ้นมา	✓
19. หน้าจอแสดงสรุปผลการแข่งขันแสดงตัวผู้เล่น	✓

ตารางที่ 4.1 (ต่อ) สรุปผลการทดลอง

การทดลอง	ผลการทดลอง
20. หน้าจอแสดงสรุปผลการแข่งขันแสดงจำนวนการ์ดค้อน	✓
21. หน้าจอแสดงสรุปผลการแข่งขันแสดงจำนวนการ์ดเคียว	✓
22. หน้าจอแสดงสรุปผลการแข่งขันแสดงจำนวนการ์ดไขควง	✓
23. หน้าจอแสดงสรุปผลการแข่งขันแสดงจำนวนการ์ดสะพาน	✓
24. หน้าจอแสดงสรุปผลการแข่งขันแสดงคะแนนรวม	✓
25. หน้าจอแสดงสรุปผลการแข่งขันแสดงบอกผู้ชนะ	✓
26. ทำการเปิดเครื่องบริการ (server)	✓
27. เข้าเล่นแบบออนไลน์	✓
28. กรอกชื่อผู้เล่น รอผู้เล่นอื่นเข้ามา แล้วเข้าเล่น	✓
29. เข้าเล่นพร้อมกันหลายกลุ่ม	✓
30. เก็บชื่อผู้เล่นลงเพิ่มข้อความ	✓
31. เก็บค่าลูกเต๋าของผู้เล่นลงเพิ่มข้อความ	✓
32. เก็บการเดินของตัวผู้เล่นลงเพิ่มข้อความ	✓
33. เก็บจำนวนการ์ดค้อนของผู้เล่นลงเพิ่มข้อความ	✓
34. เก็บจำนวนการ์ดเคียวของผู้เล่นลงเพิ่มข้อความ	✓
35. เก็บจำนวนการ์ดไขควงของผู้เล่นลงเพิ่มข้อความ	✓
36. เก็บจำนวนการ์ดสะพานของผู้เล่นลงเพิ่มข้อความ	✓
37. เก็บคะแนนรวมของผู้เล่นลงเพิ่มข้อความ	✓

จากการทดลองที่กล่าวมาทั้งหมดพบว่า สามารถเล่นเกม Hard choices แบบออฟไลน์และแบบออนไลน์ได้ สามารถเล่นได้ตั้งแต่ 2 ถึง 4 คน ในระหว่างเกมสามารถเก็บการ์ดเครื่องมือต่างๆ ได้ รวมถึงการ์ดสะพานด้วย และเมื่อผู้เล่นมีการ์ดสะพานอยู่ในมือ ในการทยอยลูกเต๋ารอบต่อไปจะ

นำค่าการวัดสะพานไปลบออกจากค่าลูกเต๋าที่ทอยได้ และยังสามารถทิ้งการวัดสะพานได้ด้วย โดยการข้ามรอบเดินนั้นของตัวเองไป หลังจากจบเกมมีการสรุปผลการแข่งขัน บอกถึงรายละเอียดต่างๆ และบอกผู้ชนะในรอบนั้น ในการเก็บสถิติสามารถเก็บจำนวนคะแนนลูกเต๋ามาจากการทอยของผู้เล่นได้ เก็บตำแหน่งการเดินของผู้เล่นได้ เก็บจำนวนการวัดเครื่องมือต่างๆของผู้เล่นได้ รวมทั้งการวัดสะพานด้วย และยังสามารถเก็บคะแนนรวมทั้งหมดในรอบนั้นของผู้เล่นได้อีกด้วย นอกจากนี้การเล่นแบบออนไลน์ยังสามารถเข้าเล่นพร้อมกันได้หลายกลุ่ม แต่ละกลุ่มจะมีผู้เล่นได้สูงสุดได้ 4 คน



บทที่ 5

สรุปและข้อเสนอแนะ

โดยปกติแล้วผู้ที่เล่นเกม Hard choices จะต้องไปอยู่รวมกันในสถานที่ใดที่หนึ่ง เพื่อที่จะทำการเล่นเกม Hard choices ทำให้เกิดความไม่สะดวกและวุ่นวาย เกม Hard choices รุ่นเดสก์ทอป (ผู้เล่นหลายคน) นี้พัฒนาขึ้นมาเพื่อความสะดวกในการเล่น Hard choices ที่สามารถรองรับผู้เล่นได้ตั้งแต่ 2 ถึง 4 คน และสามารถเล่นผ่านระบบเครือข่ายอินเทอร์เน็ตได้ สามารถนำไปประยุกต์ใช้ได้หลายสถานที่ เช่น ที่พักออาศัย มหาวิทยาลัย บริษัท เป็นต้น

หลักการและทฤษฎีที่นำมาใช้ทำโครงการนี้ ได้แก่ หลักการเขียนโปรแกรมด้วยภาษาจาวา เพื่อใช้ในการสร้างส่วนติดต่อผู้ใช้แบบกราฟิก (Graphical User Interface) หลักการแม่ข่ายลูกข่าย เพื่อให้โปรแกรมสามารถใช้งานผ่านระบบอินเทอร์เน็ตได้

ขั้นตอนการทำงานของโปรแกรมโดยภาพรวมขั้นตอนแรกเลือกว่าจะเล่นเกมแบบ ไหน (ออนไลน์ หรือ ออฟไลน์) จากนั้นทำการกรอกชื่อผู้เล่นแล้วเข้าเล่นเกม ในระหว่างการเล่น จะมีการเก็บบันทึกค่าลูกเต๋าและตำแหน่งการเดินของผู้เล่น เมื่อการแข่งขันสิ้นสุดลงจะมีการสรุปผลการแข่งขันเพื่อหาผู้ชนะ และจะมีบันทึกสรุปผลการแข่งขันของแต่ละคนลงในแฟ้มข้อความ (text file)

ผลการทดลองของ โปรแกรมที่พัฒนานั้น สามารถมีผู้เล่นเข้าเล่นได้ตั้งแต่ 2 ถึง 4 คน สามารถเล่นผ่านระบบเครือข่ายอินเทอร์เน็ตได้ และสามารถบันทึกข้อมูลของผู้เล่นแต่ละคนลงในแฟ้มข้อความได้อย่างถูกต้อง

5.1 สรุปผลการดำเนินงาน

ตารางที่ 5.1 สรุปผลการดำเนินงาน

การดำเนินการ	ผลการดำเนินงาน
1. ทดลองเล่นเกมแบบออฟไลน์	✓
2. ทดลองเล่นเกมแบบออนไลน์	✓
2.1 ทดลองเข้าเล่นพร้อมกันหลายกลุ่ม	✓
3. ทดลองการเก็บสถิติข้อมูลของผู้เล่น	✓

โปรแกรมเกม Hard choices รุ่นเดสก์ทอป (ผู้เล่นหลายคน) เป็นโปรแกรมที่สามารถเล่นได้ทั้งแบบออนไลน์ และแบบออฟไลน์ พร้อมทั้งยังสามารถเก็บสถิติต่างๆของผู้เล่น ได้แก่ ค่าลูกเต๋าที่ทอยได้ ตำแหน่งการการเดิน การ์ดเครื่องมือต่างๆที่เก็บมา จำนวนครั้งที่ข้ามสะพาน และคะแนนรวมในรอบการเล่นนั้น และสามารถเล่นเข้าเล่นได้พร้อมกันหลายกลุ่มเมื่อเล่นแบบออนไลน์

5.2 ปัญหาในการดำเนินงาน และแนวทางในการแก้ไขปัญหา

ตารางที่ 5.2 ปัญหาในการดำเนินงาน และแนวทางในการแก้ไขปัญหา

ปัญหาในการดำเนินงาน	แนวทางในการแก้ไขปัญหา
1. ขาดความรู้เกี่ยวกับวิธีการเล่น กฎกติกา และไม่เข้าใจเกี่ยวกับเกม Hard choices	1. ศึกษา และค้นคว้าหาข้อมูลเกี่ยวกับเกม Hard choices โดยละเอียด ทั้งวิธีการเล่น และกฎกติกาในการเล่น จากเว็บไซต์ของสถาบันวิศวกรรมซอฟต์แวร์ มหาวิทยาลัยคาร์เนกี เมลลอน สหรัฐอเมริกา 2. ปรึกษาอาจารย์ที่ปรึกษาถึงจุดประสงค์ของเกม Hard choices
2. ขาดความรู้เรื่องการออกแบบให้โปรแกรมออกมาดูดี และน่าสนใจ	1. เข้าร่วม โครงการอบรมพิเศษเรื่องแนวคิดพื้นฐานของการออกแบบประสบการณ์ของผู้ใช้ที่ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ มหาวิทยาลัยนเรศวร

ตารางที่ 5.1 (ต่อ) ปัญหาในการดำเนินงาน และแนวทางในการแก้ไขปัญหา

ปัญหาในการดำเนินงาน	แนวทางในการแก้ไขปัญหา
	2. ปรึกษาอาจารย์ที่ปรึกษาถึงความเหมาะสมต่างๆ เกี่ยวกับการออกแบบโปรแกรม
3. ยังขาดความรู้ในบางเรื่องของการเขียนโปรแกรมด้วยภาษาจาวา และเครื่องมือที่ใช้ในการพัฒนา	1. ศึกษา และค้นคว้าข้อมูลจากหนังสือ และอินเทอร์เน็ต และลองทำตามตัวอย่างที่ได้ศึกษามา เพื่อนำมาประยุกต์ใช้ในการพัฒนาโปรแกรม 2. เข้าร่วมอบรมการเขียนโปรแกรมจาวา ที่ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ มหาวิทยาลัยนเรศวร 3. ปรึกษาอาจารย์ที่ปรึกษาเกี่ยวกับแนวคิดและเทคนิคเกี่ยวกับการเขียนโปรแกรม
4. พบปัญหาขึ้นขณะกำลังพัฒนาโปรแกรมอยู่บ่อยๆ	1. ปรึกษาอาจารย์ที่ปรึกษาถึงสาเหตุของการเกิดปัญหา ได้คำแนะนำให้ทำการทดสอบโปรแกรมก่อนที่จะเริ่มเขียนโปรแกรม 2. ศึกษา และค้นคว้าหาข้อมูลเกี่ยวกับวิธีการทดสอบโปรแกรม จากอินเทอร์เน็ต 3. การทดสอบโปรแกรม โดยใช้เครื่องมือที่มีชื่อว่า JUnit ในการทำการทดสอบโปรแกรม
5. ขาดความรู้เกี่ยวกับระบบเครือข่ายอินเทอร์เน็ต และการเขียนโปรแกรม	1. ศึกษา และค้นคว้าข้อมูลจากหนังสือ และอินเทอร์เน็ต และลองทำตามตัวอย่างที่ได้ศึกษามา เพื่อนำมาประยุกต์ใช้ในการพัฒนาโปรแกรม 2. ปรึกษาอาจารย์ที่ปรึกษาเกี่ยวกับทฤษฎีแนวคิด เกี่ยวกับระบบเครือข่ายอินเทอร์เน็ต และการเขียนโปรแกรม

5.3 ข้อเสนอแนะ

5.3.1 แนวทางอื่นในการแก้ปัญหา

ยกตัวอย่างเช่น ปัญหาเรื่องการออกแบบ

ตารางที่ 5.3 แนวทางอื่นในการแก้ไข้ปัญหา

ปัญหาในการดำเนินงาน	แนวทางอื่นในการแก้ไข้ปัญหา
1. ขาดความรู้เรื่องการออกแบบให้โปรแกรมออกมาดูดี และน่าสนใจ	1. ศึกษาค้นหาหาความรู้จากหนังสือ และ อินเทอร์เน็ต 2. หาโปรแกรมต่างๆ มาลองใช้งาน เพื่อศึกษารูปแบบของ โปรแกรม แล้วนำประสบการณ์มาใช้ในการออกแบบ โปรแกรมของเรา

5.3.2 แนวคิดที่อาจทำเพิ่มเติม

- พัฒนาในส่วนของหน้าจอ ให้สามารถใช้งานกับระบบจอภาพของคอมพิวเตอร์ได้ทุกความละเอียด เพื่อความสะดวกในการใช้งาน
- พัฒนาในส่วนของการเล่นผ่านระบบเครือข่ายอินเทอร์เน็ตในสามารถใช้งานได้ดีและสมบูรณ์มากขึ้นกว่านี้

5.3.3 ความรู้และทักษะที่ควรมี

- ควรมีความรู้พื้นฐานทางด้านการสร้างส่วนติดต่อผู้ใช้แบบกราฟิก (Graphical User Interface) ของภาษาจาวา เช่น รูปแบบการจัดวาง การทำเหตุการณ์ต่างๆ เป็นต้น ตัวอย่างคลาสที่ต้องไปศึกษา คลาส Component คลาส BorderLayout คลาส Event Interface คลาส MouseEven คลาส Point เป็นต้น
- ควรมีความรู้พื้นฐานทางด้านกราฟิก 2 มิติ ภาษาจาวา เช่น การวาดรูปสี่เหลี่ยม การวาดรูปภาพจากวิถี (path) ของอ็อบเจกต์ (object) อิมเมจ (image) การเขียนตัวอักษรในรูปแบบของกราฟิก เป็นต้น ตัวอย่างคลาสที่ต้องไปศึกษา คลาส Graphic เป็นต้น
- ควรมีความรู้พื้นฐานในการสร้างสวิงคอมโพเนนซ์ (swing components) ที่ใช้ในการสร้าง GUI ในภาษาจาวา เช่น การสร้างเฟรม สร้างปุ่ม สร้างรูปภาพ สร้างตัวอักษร เป็นต้น ตัวอย่าง

คลาสที่ต้องไปศึกษา คลาส ImageIcon คลาส JFrame คลาส JPanel คลาส JLabel คลาส JTextField
คลาส JButton เป็นต้น

4. ควรมีความรู้พื้นฐานเกี่ยวกับการสร้างไฟล์ การเขียนข้อมูลลงไฟล์ การอ่านค่าข้อมูล
ในภาษาจาวา เป็นต้น ตัวอย่างคลาสที่ต้องไปศึกษา คลาส File คลาส FileOutputStream เป็นต้น

5. ควรมีความรู้พื้นฐานด้านระบบเครือข่ายอินเทอร์เน็ต โดยเฉพาะในเรื่องของ TCP/IP
ซึ่งเป็นระบบโปรโตคอล ของการสื่อสารพื้นฐานของระบบเครือข่ายอินเทอร์เน็ต ตัวอย่างคลาสที่
ต้องไปศึกษา คลาส Socket คลาส BufferedReader คลาส PrintWriter คลาส Thread เป็นต้น

5.3.4 หนังสือแนะนำ

1. Head First Java (2ndED) by Kathy Sierra and Bert Bates บทที่ 12 เรื่อง A Very
Graphic Story หน้า 355 – 385 บทที่ 13 เรื่อง Work on your Swing หน้า 400 – 413 บทที่ 14 เรื่อง
Saving Objects หน้า 431 – 458 บทที่ 15 เรื่อง Make a Connection หน้า 473 – 521



เอกสารอ้างอิง

- [1] Carnegie Mellon University. (2012). **Software Engineering Institute**. Retrieved September 4, 2011, from <http://www.sei.cmu.edu/architecture/tools/hardchoices>
- [2] Kathy Sierra and Bert Bates. (2005). **Head First Java**. (2ndED). United States of America: O'Reilly Media, Inc.
- [3] Hong Zhang and Y. Daniel Ling. (2007). **Computer graphics using Java 2 and 3D**. United States of America: Pearson Education, Inc.
- [4] Kenneth L. Calvert and Michael J. Donahoo. (2008). **TCP/IP sockets in Java: practical guide for programmers**. (2ndED). United States: Elsevier, Inc.
- [5] อรพิน ประวัตติบริสุทธ์. (2553). **คู่มือเขียนโปรแกรมด้วยภาษา Java ฉบับปรับปรุงใหม่**. กรุงเทพฯ ฯ : บริษัท โปรวิชั่น จำกัด.
- [6] สมชาย ประสิทธิ์ชูตระกูล. (2553). **เริ่มเรียนเขียนโปรแกรม ฉบับวาจาจาวา**. กรุงเทพฯ ฯ : บริษัท ส. เอเชียเพรส จำกัด.
- [7] Function.in.th. (2010). **Windows Apps: JAVA**. Retrieved October 22, 2012, from <http://code.function.in.th/java>
- [8] MediaGroup1 LLC. (2012). **Java Tutorials**. Retrieved April 2, 2012, from <http://www.dreamincode.net/forums/topic/143089-linked-list-tutorial/>
- [9] ณัฐภัทรศญา ทับทิมเทศ. (2552). **การเขียนโปรแกรมเชิงวัตถุ**. Retrieved September 23, 2011, from <http://www.no-poor.com/JavaandAi/chapter8-java.htm>

- [10] Manuel Moons. (November 4, 2005). **How to set JFrame to full screen**. Retrieved July 2, 2012, from <http://www.coderanch.com/t/341780/GUI/java/set-Jframe-full-screen>
- [11] พลวัฒน์ แซ่ลิ้ม. (2009). **ระบบสื่อสารข้อมูลและเครือข่ายคอมพิวเตอร์**. Retrieved September 12, 2012, from <http://portal.in.th/elec64/pages/6291/>
- [12] Simple Machines LLC. (2010). **การพัฒนาโปรแกรม client-server**. Retrieved September 12, 2012, from <http://www.codeshade.com/talk/index.php?topic=119.0>
- [13] Thaiit. (2009). **โปรโตคอล TCP/IP**. Retrieved October 9, 2012, from http://www.it.co.th/networkdetail.php?n_id=21
- [14] ชวลิต ทินกรสูติบุตร. (2011). **ความรู้พื้นฐานเกี่ยวกับ โปรโตคอล TCP/IP**. Retrieved October 23, 2012, from http://www.tnetsecurity.com/content_basic/tcp_ip_knowledge.php

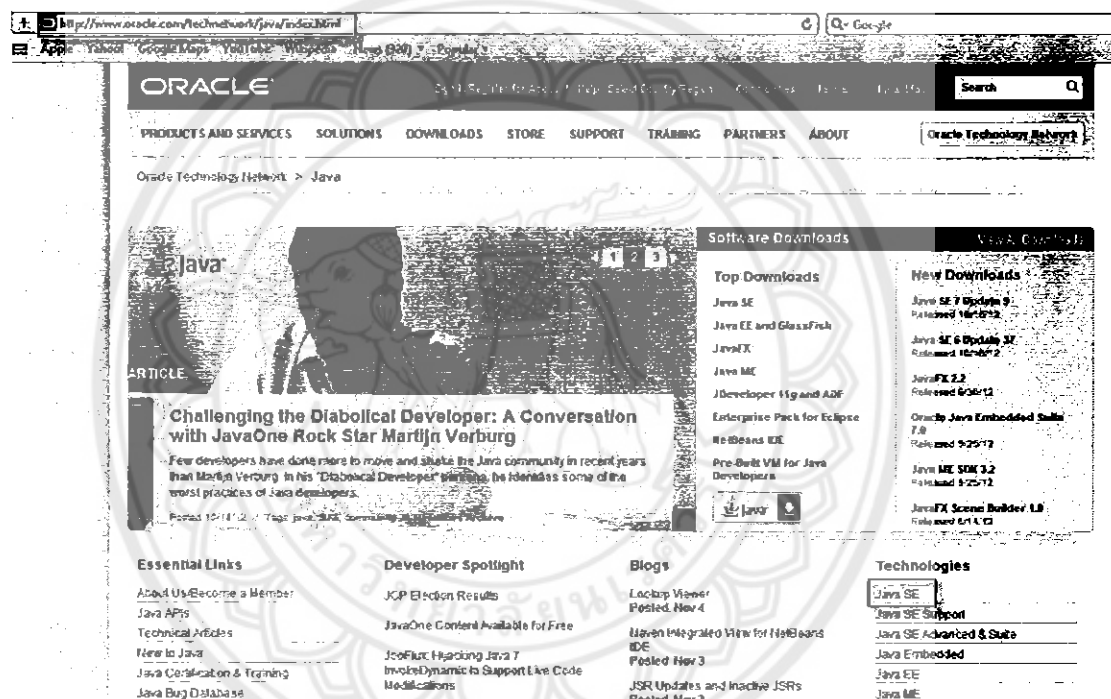


ภาคผนวก

การติดตั้งตัวแปลภาษาจาวา

วิธีการดาวน์โหลดและติดตั้งตัวแปลภาษาจาวา มีขั้นตอนดังต่อไปนี้

1. ใช้เว็บเบราว์เซอร์เข้าไปที่ <http://www.oracle.com/technetwork/java/index.html> แล้วคลิกที่ Java SE ภายใต้หัวข้อ Technologies ดังรูป



2. เมื่อเข้าสู่หน้าจอ JAVA SE Downloads ให้เลื่อนไปที่หัวข้อ Updates และคลิกลิงก์ Java SE 7 Updates 9 Released เพื่อดาวน์โหลด

What's New

Oracle Outlines Plans to Make the Future Java During JavaOne 2012 Strategy Keynote - Details Upcoming Java Platform Advancements and Highlights Java Momentum (Press Release)

Oracle Outlines Roadmap for Java SE and JavaFX at JavaOne 2012 - Java SE 8 to

Updates

Java SE 7 Update 9 Released

This releases address security concerns. Oracle strongly recommends that all Java SE 7 users upgrade to this release. JavaFX 2.2.3 is now bundled with the JDK on Windows, Mac and Linux x86/x64. [Learn More](#) [Download](#)

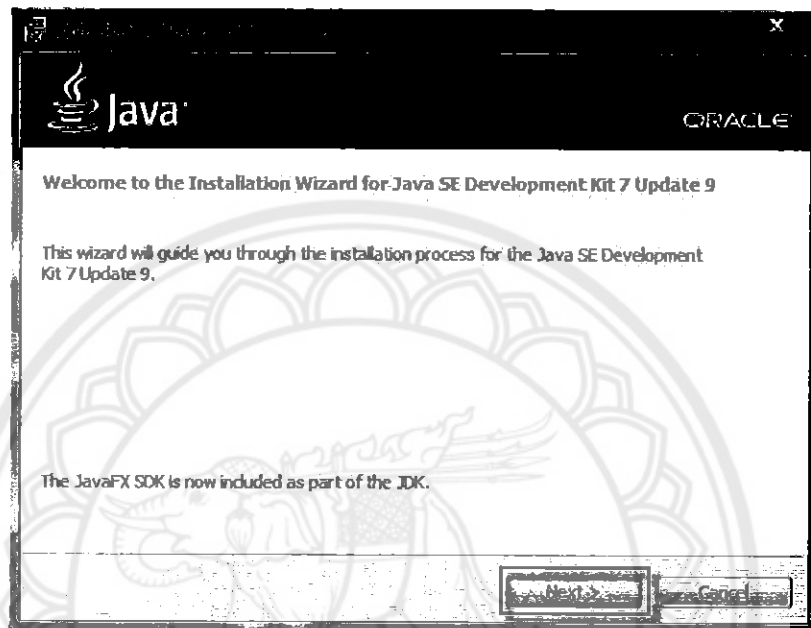
3. จะเข้าสู่หน้าจอ Downloads ภายใต้หัวข้อ Java SE Downloads คลิกปุ่มดาวน์โหลด ดังรูป เพื่อเริ่มต้นดาวน์โหลด JDK

4. จะเข้าสู่หน้าจอ Downloads ภายใต้หัวข้อ Java SE Development Kit 7u9 คลิกเลือก Accept License Agreement ดังรูป และคลิกถึงที่ `jdk-7u9-windows-i586.exe` เพื่อเริ่มต้นดาวน์โหลด

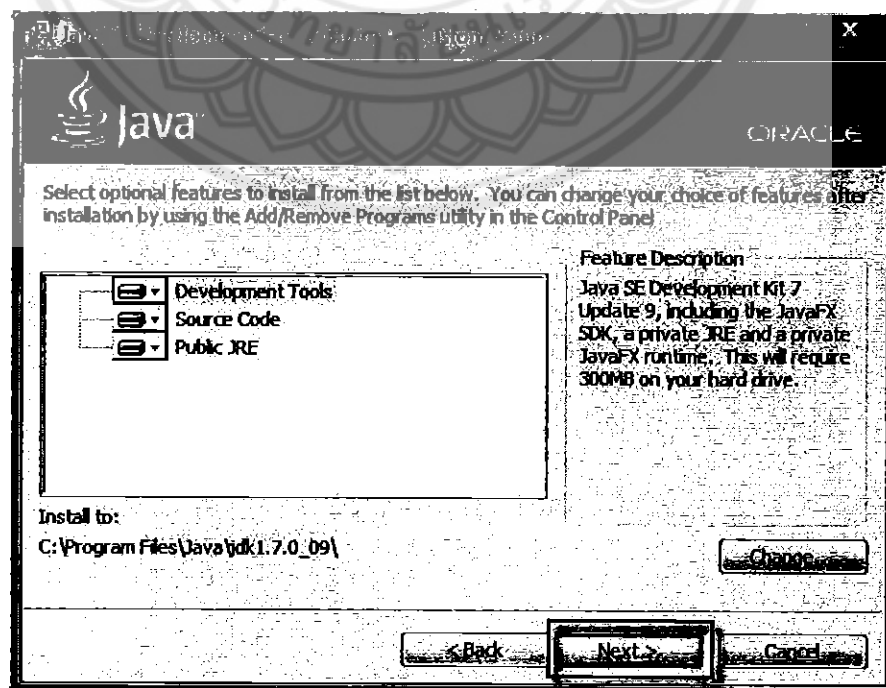
Product	File Size	Download
Linux x86	120.63 MB	jdk-7u9-linux-i586.rpm
Linux x86	92.85 MB	jdk-7u9-linux-i586.tar.gz
Linux x64	118.82 MB	jdk-7u9-linux-x64.rpm
Linux x64	91.59 MB	jdk-7u9-linux-x64.tar.gz
Mac OS X	143.47 MB	jdk-7u9-macosx-x64.dmg
Solaris x86	135.14 MB	jdk-7u9-solaris-i586.tar.Z
Solaris x86	91.51 MB	jdk-7u9-solaris-i586.tar.gz
Solaris SPARC	135.7 MB	jdk-7u9-solaris-sparc.tar.Z
Solaris SPARC	95.15 MB	jdk-7u9-solaris-sparc.tar.gz
Solaris SPARC 64-bit	22.8 MB	jdk-7u9-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	17.51 MB	jdk-7u9-solaris-sparcv9.tar.gz
Solaris x64	22.48 MB	jdk-7u9-solaris-x64.tar.Z
Solaris x64	14.94 MB	jdk-7u9-solaris-x64.tar.gz
Windows x86	88.35 MB	jdk-7u9-windows-i586.exe
Windows x64	90.03 MB	jdk-7u9-windows-x64.exe

ถึงตอนนี้หาก JDK 7 ถูกดาวน์โหลดจนครบสมบูรณ์แล้ว ก็เริ่มเข้าสู่ขั้นตอนการติดตั้งเครื่องมือได้เลย

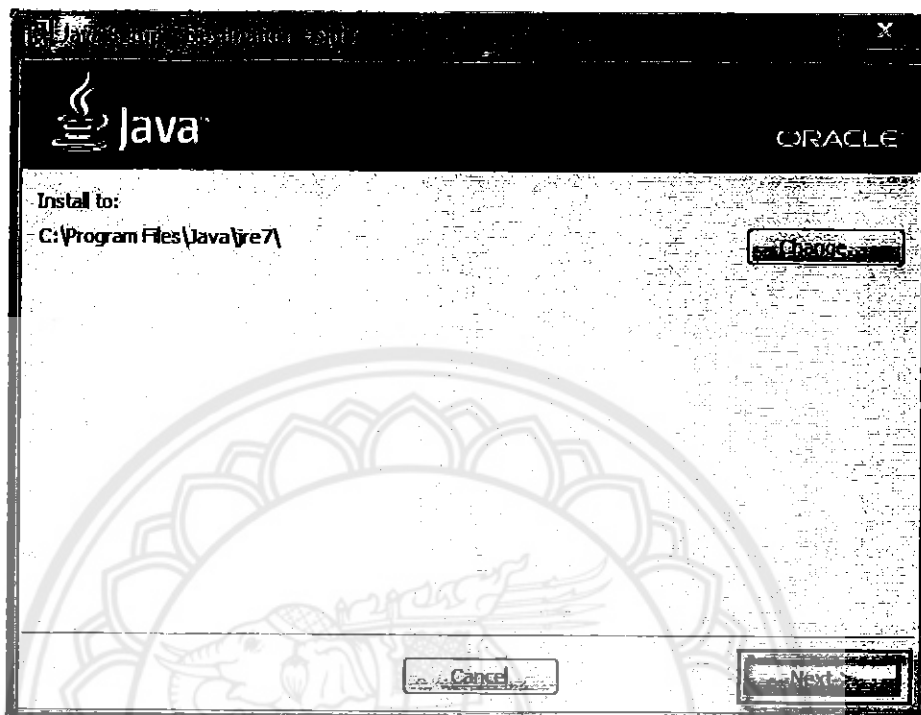
5. ให้ดับเบิลคลิกที่ไฟล์ `jdk-7u9-windows-i586.exe` ซึ่งเป็น JDK ที่ดาวน์โหลดมา จะเข้าสู่หน้าจอการติดตั้งของโปรแกรม ให้คลิกปุ่ม **Next**



6. จะมีกล่องประกอบให้เลือกติดตั้งทั้งหมด 3 ตัวด้วยกัน คือ Development Tools, Source Code, Public JRE ให้คลิกปุ่ม **Next**



7. ในระหว่างนี้ให้รอสักครู่เนื่องจากโปรแกรมกำลังติดตั้งลงในเครื่อง จากนั้นจะมีหน้าจอปรากฏดังรูป ให้คลิกปุ่ม Next เพื่อติดตั้ง JRE ลงที่เครื่อง



8. เมื่อติดตั้งเสร็จ ให้คลิกปุ่ม Next เป็นอันเสร็จเรียบร้อย

