บทที่ 3
การสร้างและพัฒนาโปรแกรม
ตัวต่อประลองปัญญาผ่านระบบเครือข่ายอินเทอร์เน็ต

หลังจากที่ได้ศึกษารวบรวมข้อมูลต่าง ๆ ศึกษาการใช้งานฟังก์ชันของ Visual Basic มาระยะเวลาหนึ่ง ก็ถึงเวลาแล้วที่คณะผู้จัดทำจะได้เริ่มดำเนินการพัฒนาโปรแกรม ในอันดับแรกคณะผู้จัดทำโครงงานก็จะต้องทำการออกแบบโปรแกรมว่าในโปรแกรมนั้นจะประกอบด้วยอะไรบ้าง ซึ่งทางคณะผู้จัดทำได้ข้อสรุปออกมาดังนี้

โปรแกรมตัวต่อประลองปัญญาผ่านระบบเครือข่าย จะประกอบไปด้วย

1. ตัวต่อประลองปัญญาในแบบเล่นคนเดียว (Single Player) และในส่วนนี้ก็จะประกอบไปด้วย
   - สมุดสะสมภาพ (Album)
   - พจนานุกรมของครอสเวิร์ด (Dictionary of Crossword)
2. ตัวต่อประลองปัญญาในแบบเล่นผ่านเครือข่าย ซึ่งสามารถรองรับจำนวนผู้เล่นได้ไม่เกิน 4 คน

สำหรับอัลกอริธึมที่ใช้ในการพัฒนาโปรแกรมนั้นทางคณะผู้จัดทำได้สร้างสรรค์ขึ้นมาเอง อาจจะเป็นอัลกอริธึมที่ไม่เหมาะสมมากนัก โดยอัลกอริทึมในการพัฒนาโปรแกรมสามารถอธิบายได้ดังนี้

3.1 อัลกอริทึมในการสุ่มคำลงในเมตริกซ์ของเกมตัวต่อประลองปัญญา

การสุ่มหรือการแรนดอมเป็นขั้นตอนที่สำคัญขั้นตอนหนึ่งของเกมนี้ การสุ่มคำศัพท์แต่ละคำเพื่อใส่ลงในเมตริกซ์นั้น ก่อนอื่นสิ่งที่เราต้องคำนึงถึงเป็นอันดับแรกคือ ขนาดของเมตริกซ์ที่ผู้เล่นจะใช้เล่นและหมวดของคำศัพท์ เมื่อเราทราบทั้งสองอย่างนี้แล้วเราก็จะสามารถดึงข้อมูลคำศัพท์จากฐานข้อมูลมาใช้ได้อย่างถูกต้องและมีประสิทธิภาพ ตัวอย่างเช่น เมื่อผู้เล่นเล่นในขนาดเมตริกซ์ 5*5 เราก็ดึงคำศัพท์เฉพาะที่มีความยาวไม่เกิน 5 ตัว หรือจำนวนของคำที่สุ่มเพื่อใส่ลงเมตริกซ์ก็จะจำกัดอยู่ที่จำนวนแถวของเมตริกซ์

สำหรับในขั้นตอนต่อไปคือ การสุ่มคำลงในเมตริกซ์ เราจะทำการสุ่มทีละคำโดยแยกอักษรแต่ละตัวจากคำสุ่มลงในตำแหน่งต่าง ๆ ของเมตริกซ์ โดยเราจะต้องสร้างตัวแปรอาเรย์แบบ 2 มิติขึ้นมาเพื่อเก็บค่าในแต่ละตำแหน่งของเมตริกซ์ เมื่อเราสุ่มตัวอักษรใส่ในตำแหน่งใดของเมตริกซ์ แล้วเราจะ

มาร์กค่าที่ตำแหน่งนี้ในตัวแปรเพื่อแสดงว่าตำแหน่งนี้ได้มีการใส่อักษรลงไปแล้ว ทุกครั้งก่อนที่เราจะใส่ตัวอักษรลงในเมตริกซ์เราจะต้องเช็คค่าจากตัวแปรนี้เสมอว่าตำแหน่งที่เครื่องสุ่มได้นั้นเคยมีตัวอักษรจากคำนี้หรือคำอื่นๆใส่อยู่ก่อนแล้วหรือไม่ ถ้าซ้ำโปรแกรมจะทำการสุ่มตำแหน่งใหม่ในเมตริกซ์จนกว่าจะได้ตำแหน่งใหม่ที่ไม่ซ้ำ

แนวคิดนี้รับรองได้ว่าคำที่จะเกิดขึ้นในเมตริกซ์จะมีอยู่อย่างแน่นอนและมีจำนวนคำศัพท์ตามที่ได้สุ่มไว้เป็นอย่างต่ำ ตัวอย่างเช่น ในเมตริกซ์ขนาด 5*5 จะมีคำศัพท์ที่สุ่มไว้จำนวน 5 คำ

3.2 อัลกอริธึมในการเช็คคำของเกมต่อประลองปัญญา

กระบวนการเช็คคำของเกมนั้นจะเกิดขึ้นทุกครั้งที่มีการเปลี่ยนตำแหน่งของตัวอักษรในเมตริกซ์ซึ่งเกิดจากการคลิกเมาส์ของผู้เล่นลงบนเมตริกซ์ ไม่ว่าจะเป็นการคลิกเพื่อสลับตัวอักษรจากซ้ายไปขวา ขวาไปซ้าย หรือบนลงล่าง ล่างขึ้นบน โดยโปรแกรมจะทำการเช็คคำตามแนวนอนในทุก ๆ แถวทั้งเมตริกซ์เพื่อตรวจสอบว่ามีคำศัพท์ใดเกิดขึ้นในเมตริกซ์บ้าง

		COLUMN				
		0	1	2	3	4
ROW	0		A	N	T	
	1		C	S	K	D
	2	E	S	C	A	E
	3	D	Y	F	A	U
	4	O	W	L	I	G

รูปที่ 3.1 เมตริกซ์ขนาด 5*5

จากรูปที่ 3.1 เราจะอธิบายหลักการทำงานได้ดังต่อไปนี้

3.2.1 ทุกครั้งที่มีการเช็คคำเราจะใช้ตัวแปรต่อไปนี้เพื่อช่วยในการเช็คคำ

- Shift เป็นตัวแปรที่จะเป็นตัวอ้างอิงตำแหน่งแรกสุดของคำที่เราจะเช็ค ตัวอย่าง ถ้าเราต้องการจะเช็คคำในตำแหน่งพิกัดที่(0,0) ถึง (0,4) ค่า Shift ขณะนี้ก็คือ 0 ค่า Shift ก็คือ ค่าของคอลัมน์นั่นเอง
- Round เป็นตัวแปรที่จะเป็นตัวบอกว่าขณะกำลังเช็คคำอยู่ใน Row ที่เท่าใด
- State เป็นตัวแปรที่จะเป็นตัวบอกว่าขณะนี้กำลังเช็คคำที่ความยาวกี่ตัวอักษร

3.2.2 จากรูปที่ 3.1 เริ่มแรกเราจะกำหนดค่าของ Shift และ Round ให้เท่ากับ 0, State=5 หมายถึงเราจะเริ่มเช็คคำในตำแหน่งพิกัด (0,0) ถึง (0,4)

3.2.3 ตรวจสอบตำแหน่งพิกัดที่ (0,0) ถึง (0,4) ว่ามีตำแหน่งใดหรือไม่ที่เป็นช่องว่างที่ไม่มีตัวอักษรการที่เราจะทราบว่าตำแหน่งใดในเมตริกซ์เป็นช่องว่างหรือไม่นั้น เราจะมีตัวแปรอาเรย์แบบ 2 มิติตัวหนึ่งซึ่งจะเก็บค่าในทุกตำแหน่งของเมตริกซ์เพื่อคอยกำหนดว่าตำแหน่งใดว่างหรือไม่ว่าง จากรูปจะเห็นว่าที่ (0,0) และ (0,4) เป็นตำแหน่งว่างไม่มีตัวอักษร เพราะฉะนั้นเราจะลดค่าของ State ลงเหลือเท่ากับ 4 แล้วเริ่มเช็คต่อไปโดยเริ่มที่ (0,0) ถึง (0,3) ก็จะเห็นว่าเรายังไม่สามารถจะเช็คคำที่มีขนาดความยาว 4 ตัวได้ ถึงแม้ว่าเราจะลดค่าของ State ลงมาเหลือเท่ากับ 2 ซึ่งเป็นขนาดของคำที่สั้นที่สุดที่เกมนี้จะเช็คได้

3.2.4 เลื่อนตำแหน่งจุดอ้างอิง จากข้อ 3 จะเห็นว่าเราไม่สามารถจะเช็คคำใดได้เลย เราจึง ต้องย้ายจุดอ้างอิงมาอยู่ที่ตำแหน่ง (0,1) โดยการปรับค่า Shift=1 ค่า Round ยังเท่าเดิม แล้วเช็คคำในตำแหน่งที่ (0,1) ถึง (0,4) จะเห็นว่าก็ยังไม่สามารถเช็คได้เนื่องจากมีช่องว่างอยู่ที่ตำแหน่ง (0,4) จึงต้องลดค่า State ลงมาเหลือ 3 เพื่อเช็คในตำแหน่งที่ (0,1) ถึง (0,3) จะเห็นได้ว่าไม่ปรากฏมีช่องว่างในช่วงนี้ เพราะฉะนั้นเราจะนำค่าของตัวอักษรในช่วงนี้มาประสมเป็นคำแล้วนำไปตรวจสอบว่าตรงกับคำศัพท์ในฐานข้อมูลหรือไม่ ถ้าไม่ตรงเราจะเลื่อนค่า Shift ไปอีก 1 แล้วทำการเช็คในแบบเดิม จากรูป เกมนี้เช็คคำได้ต่ำสุดที่ความยาว 2 ตัวอักษรเพราะฉะนั้น ค่า Shift จะเลื่อนไปได้สูงสุดไม่เกินตำแหน่งที่ (0,3) ถ้าเช็คคำพบก็จะสร้างภาพอนิเมชั่นที่คำแล้วทำให้คำศัพท์คำนั้นหายไปพร้อมกับเลื่อนตัวอักษรด้านบนลงมา

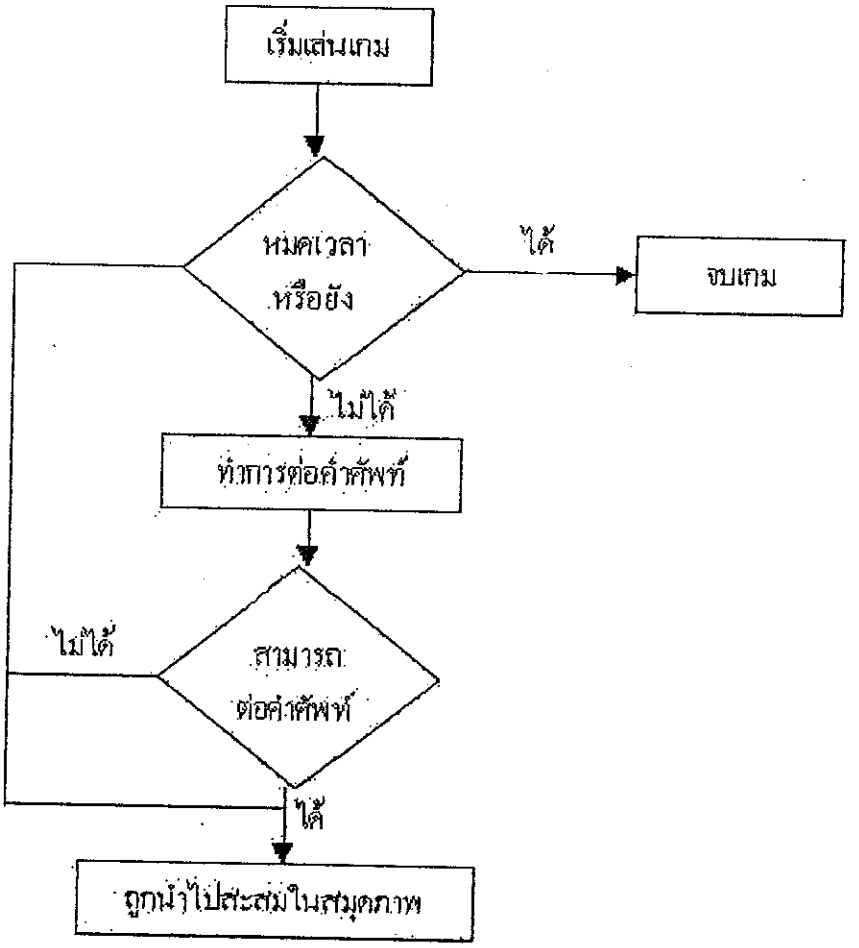
3.2.5 เพิ่มค่า Round ในกรณีที่เราเช็คคำจนหมดทั้งแถวใน Round ที่ 0 หรือ Row=0 แล้วยังไม่พบเราจะเพิ่มค่า Round ไปอีก 1 เพื่อเช็คคำในแถวถัดไปแล้วทำการเช็คคำในแบบเดิม ถ้าไม่พบก็จะเพิ่มค่า Round ไปเรื่อย ๆ จนเช็คหมดทุกแถว

3.2.6 ทุกครั้งที่เช็คคำพบ โปรแกรมจะหยุดการเช็คคำทันทีเพื่อแสดงอนิเมชั่นที่คำ แล้วแสดงความหมายของคำ ภาพประกอบ และบันทึกหมายเลขภาพลงฐานข้อมูลสมุดสะสมของผู้เล่น หลังจากนั้นก็รอการเปลี่ยนแปลงของตัวอักษรในเมตริกซ์เพื่อเริ่มกระบวนการเช็คคำใหม่จากข้อ 3.2.1

หลังจากที่ได้ทราบถึงอัลกอริธึมที่จะนำมาใช้ในการพัฒนาเกมตัวต่อประลองปัญญาผ่านระบบเครือข่ายอินเตอร์เน็ตแล้ว ต่อไปเราจะนำเอาอัลกอริธึมนี้ไปใช้เพื่อพัฒนาโปรแกรม ดังจะกล่าวในหัวข้อถัดไป

3.3 โครงสร้างของเกมแบบเล่นคนเดียว (Stand Alone)

ในส่วนนี้จะประกอบด้วย เกมตัวต่อประลองปัญญา สมุดสะสมภาพและพจนานุกรม ซึ่งในส่วนของเกมตัวต่อประลองปัญญาและสมุดสะสมภาพจะมีความสัมพันธ์กันคือ ถ้าเล่นเกมตัวต่อประลองปัญญาและสามารถต่อคำศัพท์ได้ ภาพและความหมายของคำศัพท์คำนั้นก็จะนำมาแสดงและภาพของคำศัพท์คำนั้นก็จะถูกนำไปเก็บไว้ในสมุดสะสมภาพของผู้เล่นอีกด้วย สามารถแสดงได้ดังรูป

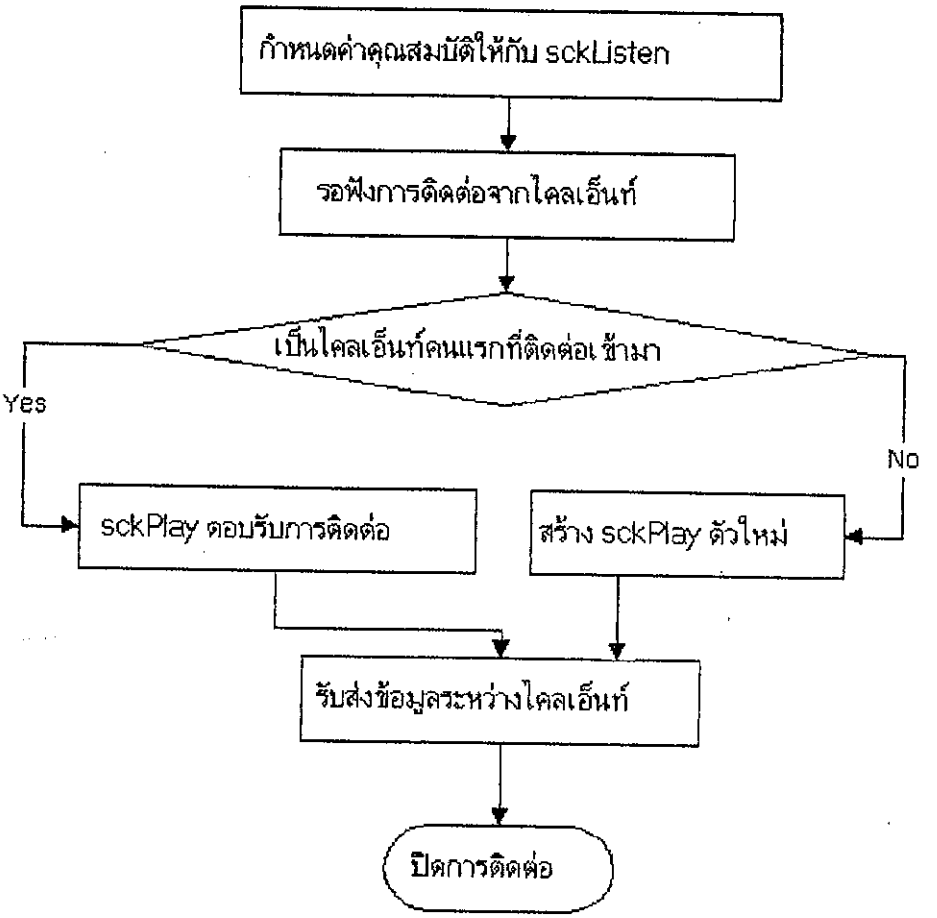


รูปที่ 3.2 การทำงานของเกมแบบเล่นคนเดียว

สำหรับในส่วนของพจนานุกรมนั้น ก็มีลักษณะโครงสร้างการทำงานเหมือนกับซอต์ฟแวร์พจนานุกรมทั่ว ๆ ไปคือ สามารถค้นหาความหมายของคำศัพท์ได้ทั้งภาษาไทยและภาษาอังกฤษ โดยพจนานุกรมนี้มีข้อจำกัดคือ พจนานุกรมนี้จะรวบรวมเฉพาะคำศัพท์ที่ใช้ในการเล่นเกมครอสเวิร์ดเท่านั้น

3.4 โครงสร้างของเกมแบบเล่นผ่านเครือข่าย

3.4.1 โฟลชาร์ตแสดงลำดับการทำงานของโปรแกรมในโหมดเซิร์ฟเวอร์



รูปที่ 3.3 ลำดับการทำงานของเซิร์ฟเวอร์

โปรแกรมที่ทำงานในโหมดของเซิร์ฟเวอร์มีจำนวนวินซ็อกคอนโทรลที่ใช้จำนวน 2 ตัวซึ่งมีชื่อว่า sckListen และ sckPlay

-sckListen มีหน้าที่รอฟังการติดต่อจากไคลเอ็นท์ที่จะทำการติดต่อเข้ามา

-sckPlay มีหน้าที่รับส่งข้อมูลกับเครื่องไคลเอ็นท์ โดยจำนวนของ sckPlay สามารถมีได้หลายตัวซึ่งจะแปรผันตามจำนวนเครื่องไคลเอ็นท์ที่ติดต่อเข้ามา

จากรูป 3.3 โฟลชาร์ตแสดงการทำงานของโปรแกรมในโหมดเซิร์ฟเวอร์ซึ่งสามารถอธิบายการทำงานเป็นขั้นตอนได้ดังนี้

1. เมื่อโปรแกรมเริ่มต้นทำงานจะต้องทำการกำหนดค่าคุณสมบัติให้กับ sckListen คือ การกำหนดค่าพอร์ต (LocalPort) ให้กับ sckListen ค่าพอร์ตนี้ใช้ในการอ้างอิงเพื่อให้เครื่องไคลเอ็นท์

สามารถติดต่อเข้ามายังเซิร์ฟเวอร์ได้ ในที่นี้จะกำหนดไว้ที่ค่า 1001 โดยปกติแล้วโปรแกรมในโหมดของไคลเอ็นท์จะรู้โดยอัตโนมัติว่าต้องติดต่อมายังพอร์ตนี้

2. เซิร์ฟเวอร์จะรอฟังการติดต่อจากไคลเอ็นท์ว่ามีเครื่องของไคลเอ็นท์เครื่องใดติดต่อเข้ามาหรือไม่

3. เมื่อปรากฏว่ามีไคลเอ็นท์ติดต่อเข้ามาโปรแกรมจะทำการตรวจสอบว่าไคลเอ็นท์ที่ติดต่อเข้ามานั้นเป็นคนแรกที่ติดต่อเข้ามาหรือไม่

* ถ้าเป็นไคลเอ็นท์คนแรกที่ติดต่อเข้ามา sckPlay(0) จะทำหน้าที่รับการติดต่อจากไคลเอ็นท์จากนั้นก็จะส่งข้อความแสดงการตอบรับไปให้เครื่องไคลเอ็นท์ที่ติดต่อเข้ามา ซึ่งจะมีข้อมูลที่เป็นหมายเลข IP Address ของเครื่องเซิร์ฟเวอร์ และหมายเลขพอร์ตที่รับการติดต่อรวมอยู่ด้วย

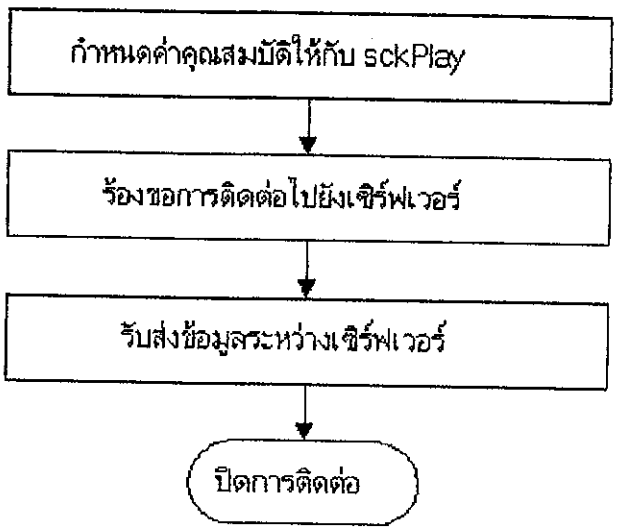
* ถ้าไม่ใช่ไคลเอ็นท์คนแรกโปรแกรมจะทำการสร้าง sckPlay(index) | index=1,2,...,n ขึ้นมาอีกตัวโดยอาศัยหลักการของ Control Array ซึ่งจะทำการเพิ่มค่าดัชนี (index) ขึ้นไปอีกหนึ่ง ในการใช้งาน sckPlay แต่ละตัวเพื่อติดต่อกับไคลเอ็นท์แต่ละเครื่องนั้นสามารถกระทำได้โดยการระบุค่าดัชนี (index)

<u>หมายเหตุ</u> สำหรับ sckPlay แต่ละตัวนั้นก็จะมีการกำหนดหมายเลขพอร์ตที่แตกต่างกันเช่นเดียวกันกับ sckListen แต่ว่าจะไม่มีการกำหนดอย่างถาวร แต่จะให้โปรแกรมทำการสุ่มหาหมายเลขพอร์ตในเครื่องที่ยังว่างอยู่ให้กับ sckPlay แต่ละตัว

4. เมื่อมีการรับการติดต่อกับไคลเอ็นท์แล้ว เซิร์ฟเวอร์กับไคลเอ็นท์เครื่องนั้นก็จะสามารถติดต่อรับส่งข้อมูลถึงกันได้

5. เมื่อเซิร์ฟเวอร์ต้องการยกเลิกการติดต่อกับไคลเอ็นท์เครื่องใด สามารถกระทำได้โดยให้ sckPlay ตัวที่ติดต่อกับไคลเอ็นท์เครื่องนั้น ใช้วิธี Close เพื่อบอกยกเลิกการติดต่อ

3.4.2 โฟลชาร์ตแสดงลำดับการทำงานของโปรแกรมในโหมดไคลเอ็นท์



รูปที่ 3.4 ลำดับการทำงานของไคลเอนท์

โปรแกรมที่ทำงานในโหมดไคลเอ็นท์นั้นจะมีจำนวนวินซ็อคคอนโทรลจำนวน 2 ตัวคือ sckListen และ sckPlay เช่นเดียวกันกับในโหมดเซิร์ฟเวอร์ เนื่องจากโปรแกรมนี้สามารถทำงานได้ทั้งเป็นแบบเซิร์ฟเวอร์หรือไคลเอ็นท์แต่ว่าในโหมดไคลเอ็นท์นั้น sckListen จะไม่ถูกใช้งานแต่จะใช้เพียง sckPlay เพื่อทำการติดต่อกับเซิร์ฟเวอร์เท่านั้น ซึ่งมีลำดับการทำงานดังนี้

1. เมื่อโปรแกรมเริ่มต้นทำงานจะต้องทำการกำหนดค่าคุณสมบัติให้กับ sckPlay คือ การกำหนดค่าพอร์ตที่ต้องการติดต่อ (RemotePort) ในที่นี้จะกำหนดค่าเป็น 1001 เพราะเป็นค่าเดียวกันกับค่าพอร์ต (LocalPort) ของ sckListen ในโหมดเซิร์ฟเวอร์ซึ่งจะทำให้สามารถทำการติดต่อกับโปรแกรมที่ทำงานในโหมดเซิร์ฟเวอร์ได้และการกำหนดชื่อเครื่องคอมพิวเตอร์หรือหมายเลข IP Addressที่ต้องการติดต่อ (RemoteHost) ให้กับ sckPlay ค่านี้ผู้เล่นจะเป็นผู้กำหนดเองในขณะที่โปรแกรมกำลังทำงานโดยผู้เล่นจำเป็นจะต้องทราบชื่อเครื่องคอมพิวเตอร์หรือค่า IP Address ของเครื่องที่มีโปรแกรมทำงานในโหมดเซิร์ฟเวอร์เพื่อสามารถทำการร้องขอการติดต่อได้
2. ร้องขอการติดต่อไปยังเครื่องที่มีโปรแกรมทำงานในโหมดเซิร์ฟเวอร์โดยการระบุค่า IP Address หรือชื่อเครื่องคอมพิวเตอร์เครื่องนั้น
3. เมื่อทำการติดต่อไปยังเซิร์ฟเวอร์ได้สำเร็จก็จะสามารถรับส่งข้อมูลกับเซิร์ฟเวอร์ได้ เมื่อไคลเอ็นท์ต้องการยกเลิกการติดต่อกับเซิร์ฟเวอร์สามารถกระทำได้โดยคลิกปุ่มออกจากโปรแกรม sckPlay จะใช้วิธีการ Close