

Discuss how you plan to implement the project

Architecture

Packages	Brief	Interface	Class -> implements interface or extends class	อธิบาย
AST (collect all type of Nodes that use in AST)	AssignmentStatement - to represent an assignment e.g. x = "opponent"	Node 1. void evaluate();	AssignmentNode -private final String identifier; -private final Node expression;	ตัวสีแดง คือ Interface ตัวสีฟ้า คือ class
	DoneCommand -to represent done commands e.g. "done"		DoneNode	
	RelocateCommand -to represent relocate commands e.g. "relocate"		Relocate	
	MoveCommand -to represent move commands e.g. "move downright"		MoveNode -private final String Direction	
	RegionCommand -to represent region commands e.g. "invest 1" or "collect 2"		RegionNode -private final long amount -private final String actionRe	
	AttackCommand -to represent attack commands e.g. "shoot downleft 5"		AttackNode -private final long moneyattack -private final String direction	
	BlockStatement -to represent blockstatement e.g. "{statement 1, statement 2,...}"		BlockNode -private List< Node > statements; -private List< Node > findAllNodes();	
	IfStatement -to represent if-then-else statement e.g. "if(x >= 1) then statement 1 else statement 2"	Node 1. void evaluate(); has class ExprNode implements it ExprNode 1. long eval(Map<String, Long> mem); this 5 class is subclass ExprNode	IfNode -private final Node condition; -private final Node trueBranch; -private final Node falseBranch;	
	WhileStatement -to represent whileloop statement e.g. "while(x > 0) statement"		WhileNode -private final Node condition; -private final Node body;	
	BinaryOperation -to represent binary expressions e.g. "x+1" or "y*2"		BinaryOperatorNode -private final String operator; -private final ExprNodes left; -private final ExprNodes right;	
	Nuber -to represent number e.g. "9"		NuberNode -private long value;	
	Identifier -to represent identifiers e.g. "x"		Identifier -private final String idf;	
	Opponent -to represent information expression e.g. "opponent"		OpponentNode	
	Near by -to represent information expression e.g. "nearby down"		NearbyNode -private final String direction;	

Parser	<p>Parser</p> <p>-เป็นตัวกรองภาษาเกมตัวหลักว่าเป็นไปตามที่ควรจะเป็นหรือไม่ แล้วถึงจะปล่อยให้ AST ได้เก็บเข้าไปในที่สุด</p>	<p>Parser</p> <p>-Node parse();</p>	<p>Parser_im</p> <p>-Tokenizer tkz;</p> <p>-Map<String, Long> mem;</p> <p>-Node parsePlan()</p> <p>-Node parseStatement()</p> <p>-Node parseBlockStatement()</p> <p>-Node parseIfStatement()</p> <p>-Node parseWhileStatement()</p> <p>-Node parseCommand()</p> <p>-Node parseAssignmentStatement()</p> <p>-Node parseActionCommand()</p> <p>-Node parseMoveCommand()</p> <p>-String parseDirection()</p> <p>-Node parseInvestCommand()</p> <p>-Node parseCollectCommand()</p> <p>-Node parseShootCommand()</p> <p>-ExprNode parseExpression()</p> <p>-ExprNode parseTerm()</p> <p>-ExprNode parseFactor()</p> <p>-ExprNode parsePower()</p>	<p>อิงแนวคิดมาจาก lab 12 และทำตาม grammar ที่อาจารย์ให้เอาไว้</p>
Tokenizer	<p>Tokennizer</p> <p>-เป็นตัวเอาไวก์รอนในขั้นต้นว่า การเขียนใน construction plan นั้นถูกต้องอย่างที่ควรเป็นหรือไม่</p>	<p>Tokenizer</p> <p>1. boolean hasNextToken();</p> <p>2. String peek();</p> <p>3. boolean peek(String s);</p> <p>4. String consume();</p> <p>5. boolean consume(String s);</p>	<p>Tokennizer_im</p> <p>-private final String src;</p> <p>-private int pos;</p> <p>-private String next;</p>	<p>อิงมาจาก lab 12 เามาประยุกต์ให้มันเข้ากับการใช้งานของเกม</p>

Code design


data structures ที่ผมคิดว่าจะใช้ในงานนี้ น่าจะเป็น List<Node> กับ Map<String, Long>


จากเท่าที่คิดและทดลองเขียนมา ข้อดี คือ การออกแบบข้างต้นนั้นจะทำให้เห็นองค์ประกอบส่วนที่เล็กที่สุดของตัว Grammar ได้อย่างชัดเจนซึ่งจะทำให้เวลาต้องกลับมาใช้งานหรือมีความจำเป็นต้องอธิบายให้คนในกลุ่มแบบเจาะลึกจะทำให้เห็นภาพได้ง่าย


ข้อเสีย คือ จะแยกไฟล์ค่อนข้างเยอะถ้าไม่เก็บให้เป็นระเบียบอาจจะทำให้เวลาต้องหยิบมาแก้อาจจะใช้เวลานาน

Tools

- 1. Disscord ใช้ในการติดต่อสื่อสารและแจ้งเตือนเกี่ยวกับ project
- 2. Github ใช้ในการอัปเดต code ของ project และเพื่อให้ผู้ตรวจงานสามารถเข้ามาดูองค์ประกอบภายในของ project ได้
- 3. Google sheet ใช้ในการเขียนเกี่ยวกับ project โดยที่คนในกลุ่มสามารถเข้ามาแก้ไขหรือปรับแก้บางส่วนของรายงานที่จะส่งได้


 main ▾


 1 branch

 0 tags

Go to file


Add file ▾


 Code ▾

 ball46

AST is finish


466adcc 7 hours ago

 15 commits

 .idea


is test

5 days ago

 project

AST is finish

7 hours ago

 README.md

Update README.md

4 days ago

README.md



Project_UPBEAT

link for project spec and time line - <https://cmu.to/projectspec>

link for look overview project - <https://cmu.to/overviewproject>

link for design overview for evaluator - <https://cmu.to/xh2hR>

<p>จากการออกแบบข้างบน ทางสมาชิกในกลุ่มทุกคนได้คิดว่าการออกแบบข้างต้นออกแบบได้ตรงตามความต้องการของ Grammar ตัวเกมครบทุกส่วน ทำให้การเขียนคำสั่งในการเล่นเกรมีรูปแบบที่ง่ายและความชัดเจนในการเล่นซึ่งจะช่วยให้ผู้เล่นไม่สับสนในการเขียนคำสั่งต่างๆและประหยัดเวลาในการวางแผนและการแก้ไขแผนการเล่น</p>	
<p>Testing: describe your testing plan</p>	
<p>Blackblock: คิดว่าน่าจะทำการ test ในรูปแบบคำตอบที่แตกต่างกันออกไป อย่างในส่วนของ tokenizer ที่มี method อยู่ทั้งหมด 5 method นั้นก็จะเขียน test ออกมาของทุก method เพื่อให้เห็นว่ามียอะไรผิดปกติเกิดขึ้นหรือไม่ โดยที่ทุก testmethod จะใช้ ตัวอย่างในการ test อยู่ 4 แบบ คือ null, "", "abc" และ "a b" ที่ว่าทำไมถึงเป็น 4 แบบนี้ เพราะว่าการให้มีความแตกต่างกันในทุกแบบ เพื่อให้การทดสอบที่ออกมา นั้นจะได้เห็นผลลัพธ์ว่าตรงกับที่คิดและออกแบบไว้หรือไม่ ส่วนที่เหลือจะทำการ test ในส่วนของ parser อย่างเดียว ในส่วนของ AST จะไม่ทำการทดสอบเพราะว่ามันเป็นส่วนที่เล็กที่สุดของ Grammar เราจึงสามารถเห็นการกระทำของมันได้อย่างไม่ยาก ซึ่งในส่วน of parser ก็คิดจะออกแบบในลักษณะเดียวกับ test คือหารูปแบบที่แตกต่างกับออกไปเพื่อให้ดูว่าผลลัพธ์เป็นไปตามที่ต้องการหรือไม่</p> <p>Glassblock: เเท่าที่ออกแบบเอาไว้คือจะหา testcase ที่จะทำการทดสอบว่า code ที่เราออกแบบไว้นั้นได้ใช้ในส่วนเราได้เขียนเอาไว้ตามกรณีที่เราคิดเอาไว้หรือไม่ อย่างถ้าเป็น tokenizer จะเห็นได้ง่ายที่สุดก็คงเป็นอย่าง method computeNext() ดูว่าถ้าเราใส่ testcase ที่แตกต่างกันมันจะเข้าไปในกรณีในแบบที่คิดเอาไว้หรือไม่</p>	
<p>Work plan</p>	
<p>บอล 640610626</p>	<p>เป็นคนทำงานในด้านของ backend ทำให้ในส่วนของ Grammar จะออกเป็นคนออกแบบโครงสร้างเป็นหลัก และจะถามความคิดเห็นกับสมาชิกในกลุ่มที่เหลือว่า โอเคหรือไม่กับการออกแบบที่ได้คิดมาหรือไม่ อยากแก้ไขตรงไหนหรือไม่</p>
<p>ดิด 640610668</p>	<p>เป็นคนทำงานในด้านของ frontend ทำให้ในส่วนของ Grammar จะออกเป็นคนแสดงความคิดเห็นและลองอ่านดูว่ามีจุดไหนดูแปลกไปหรือไม่ โดยงานที่ทำในตอนนี้คือทดสอบออกแบบ map 6 เหลี่ยมของตัวเกม UPBEAT</p>
<p>พิมพ์ 640610654</p>	<p>เป็น tester และสนับสนุนทั้งสองฝั่งของงาน ทำให้เป็นคนเสนอการออกแบบตัว test ต่างๆทางฝั่ง backend และทางฝั่ง frontend ต้องรอให้ทำตัว demo ออกได้ก่อน ส่วนที่ทำในตอนนี้ก็ออกความคิดเห็นต่างๆ เกี่ยวกับการออกแบบทางฝั่ง backend</p>
<p>1. ส่วนของการแบ่งงานที่สามารถเห็นได้อย่างชัดเจนในตอนนี้ก็คือในฝั่งของ พิมพ์ ที่ไม่สามารถออกแบบการ test ได้ถ้าทางฝั่งของ บอล ยังไม่สามารถเขียน demo ของ tokenizer, AST หรือ parser ได้ตามที่คิดเอาไว้ และจะต้องรอหรือหันไปทำงานอื่นในช่วงที่ต้องมีการแก้ไข code ถ้าเจอปัญหาที่เกิดขึ้นมา</p>	
<p>Known problems</p>	
<p>1. การทดลองเขียน grammar ในตอนนี้ยังไม่มีความแน่ชัดในส่วนของการวิธีการคำนวณ เช่นถ้า done เรารู้ว่ามันจะทำให้ฝั่งของผู้เรียกใช้งานจะจบ turn ของเขา แต่ยังไม่แน่ใจว่าควรเขียนอะไรออกมาให้ตรงตามที่ต้องการ และยังไม่มีความแน่ชัดในส่วน of parser คือ parseBlockStatement() และ parseWhileStatement() ที่ตอนนี้ยังไม่สามารถคิดได้ว่าควรเขียนออกมาแบบไหนให้ตรงกับความต้องการและที่ออกแบบเอาไว้</p>	
<p>2. การทดลองทำในส่วน of frontend ในตอนนี้ทดลองทำได้เพียงในส่วนของการออกแบบ map ของเกมเท่านั้นเนื่องจากในฝั่งของทาง backend ที่ยังไม่ถึงไหนจึงไม่สามารถต่อยอดไปทำอย่างอื่นได้เท่าที่ควร</p>	
<p>3. grid coordination ใน spec วัดด้วย array 2 มิติ แต่ใน library ของ React วัดด้วยพิกัด q r s ด้วยความที่รูปหกเหลี่ยมไม่สามารถนำมาวางเรียงกันแบบรูปสี่เหลี่ยมได้ (ต้องวางเหลื่อมกัน) ทำให้ต้องมีการใช้สูตรคำนวณพิเศษเพื่อเทียบระยะและตำแหน่งของรูปที่อยู่ถัดไป แต่แบบนี้ทำให้เกิดปัญหาใหม่เพราะเลข coord ของ grid หกเหลี่ยมมันเรียงกันค่อนข้างซับซ้อน ทำให้อาจจะทำงานร่วมกับ backend ได้ยากขึ้น</p>	