

Discuss how you plan to implement the project				
Architecture				
Packages	Subpackages	Interface / Class	Class -> implements interface or extends class	อธิบาย
Game	-	Game -Map<String, Long> getIdentifiers(); -void attack(Direction direction, long money); -void collect(long money); -void invest(long money); -void move(Direction direction); -void relocate(); -long nearby(Direction direction); long opponent();	Game_im -private final Player player1; -private final Player player2; -private final List< Region > territory; -private final Map<String, Long> identifiers;	ตัวสีแดง คือ Interface ตัวสีฟ้า คือ class ตัวสีเขียว คือ enum
			ReadData -private static int row; -private static int col; -private static int init_plan_min; -private static int init_plan_sec; -private static double init_budget; -private static double init_center_dep; -private static int plan_rev_sec; -private static int plan_rev_min; -private static int rev_cose; -private static double max_dep; -private staticdouble interest_pct; -private static List< Region > territory; -public static void getDataFile(); -public static List< Region > createMap(); -public static Player createPlayer(String name);	class นี้ไม่ได้ extends อะไรทั้งสิ้น
Player	-	Player -String getName(); -double getBudget(); -void updateBudget(double money); -void moveCityCrew(String direction); -int getCityCrewLocation(); -int getCityCenterLocation(); -boolean life(); -long attack(String direction); -Map<String, Long> getIdentifiers();	Player_im -private final String name; -private double budget; -private boolean life; -private final int CityCrew; private final int CityCenter; -private final Map<String, Long> identifier;	
Region	-	Region -String getOwner(); -void updateOwner(String owner); -double get Budget(); -void updateBudget(double money); -int getLocation(); -int Oppent(); -int nearby();	Regionr_im -private final int location; -private String owner; -private double budget;	
-	-	-	GSError extends RuntimeException -เอาไว้เก็บ Exception ที่เจอใน Game_state ทั้งหมด เพื่อให้่ายต่อการอ่านและทำความเข้าใจ	
Code design				
data structures ที่ผมคิดว่าจะใช้ในงานนี้ น่าจะเป็น List<Region> กับ Map<String, Long>				
จากเท่าที่คิดและทดลองเขียนมา ข้อดี คือ ทำให้การเก็บข้อมูลของเกม UPBEAT นั้นจะในส่วนของ Game State อย่างเดียว ซึ่งจะทำให้ข้อมูลในส่วนอื่นของตัวเกมมีเพียงข้อมูลที่จำเป็นต้องใช้งานอย่างจริงจังเท่านั้น ข้อเสีย คือ ถ้าเราเก็บข้อมูลในส่วนนี้ไม่เป็นระเบียบอาจจะทำให้การนำไปใช้งานแต่ละที่ยุ่งยาก และต้องเสียเวลาไม่มากนักน้อย ดังนั้นการกำหนดชื่อของสิ่งที่เก็บให้เข้าใจง่ายเป็นสิ่งที่ค่อนข้างสำคัญในระดับหนึ่ง				
Tools				
1. Disscord ใช้ในการติดต่อสื่อสารและแจ้งเตือนเกี่ยวกับ project				
2. Github ใช้ในการอัปเดต code ของ project และเพื่อให้ผู้ตรวจงานสามารถเข้ามาดูองค์ประกอบภายในของ project ได้				
3. Google sheet ใช้ในการเขียนเกี่ยวกับ project โดยที่คนในกลุ่มสามารถเข้ามาแก้ไขหรือปรับแก้บางส่วนของรายงานที่จะส่งได้				

จากการออกแบบข้างต้น ทางสมาชิกในกลุ่มได้หารือกันว่าโครงสร้างของตัว Game State ในตอนนี้ได้ตรงตามความต้องการที่ตัวเกมต้องการทุกอย่าง ทำให้ตัวเกมสามารถอ่านข้อมูลและส่งข้อมูลได้ตามที่ต้องการ เรียบง่ายและมีประสิทธิภาพมากที่สุด เพื่อลดภาระการประมวลผลที่จะมีระหว่างการเล่นเกมในมีเท่าที่จำเป็น	
Testing: describe your testing plan	
Blackblock: คิดว่าน่าจะทำการ test กับตัว Game เพียงอย่างเดียวเพราะ Game นั้นจะเป็นคนดึงการใช้งานในส่วนของ Player กับ Region ซึ่งการ test นั้นน่าจะแบ่งออกเป็น การทดสอบใส่ข้อมูลว่าการรับข้อมูลจากทาง frontend ด้วย file สกุล json นั้นเป็นไปตามที่ควรจะเป็นหรือไม่ ถ้าการที่เราใส่ข้อมูลไปไม่ครบนั้นจะทำให้เกิดปัญหาอย่างที่คิดเอาไว้หรือไม่, การทดสอบการสร้างแผนที่ของเกมขึ้นมาว่าการที่เราใส่ row กับ col เข้าไปในั้น territory จะต้องมีความ region ตรงตามที่ควรจะเป็นหรือไม่, การทดสอบสุดท้ายจะเป็นในส่วนของการทดสอบสร้างตัวละครของผู้เล่นขึ้นมา ซึ่งจะทดสอบว่าถ้าเราใส่ชื่อเข้าไปแบบไหนจะทำให้เกิดผลลัพธ์เป็นอย่างไรที่ออกแบบเอาไว้หรือไม่ อย่างถ้าเราได้สร้าง map แล้วไปสร้างตัวละครเลย ก็ควรที่จะไม่ผ่าน หรือไม่ก็ถ้า row หรือ col เป็น 0 ก็ไม่ควรที่จะสร้างตัวละครขึ้นมาได้	
Glassblock: เท่าที่ออกแบบเอาไว้คือจะหา testcase ที่จะทำการทดสอบว่า code ที่เราออกแบบไว้นั้นได้ใช้ในส่วนเราได้เขียนเอาไว้ตามกรณีที่เราคิดเอาไว้หรือไม่ ซึ่งในส่วนนี้จะมีความเกี่ยวข้องกับ Grammar มากพอสมควรทำให้คิดว่าการใช้เขียน test แบบ glassblock นั้นจะเอียงไปทาง Grammar ที่ออกไปในครั้งที่แล้วมากกว่า	
Work plan	
บอล 640610626	เป็นคนทำงานในด้านของ backend ทำให้ในส่วนของ Game State ก็ยังคงเป็นคนออกแบบโครงสร้างเป็นหลัก และจะถามความคิดเห็นกับสมาชิกในกลุ่มที่เหลือว่า โอเคหรือไม่กับการออกแบบที่ได้คิดมาหรือไม่ อยากทำอะไรตรงไหนรึเปล่า
ดิด 640610668	เป็นคนทำงานในด้านของ frontend ทำให้ในส่วนของ Game State จะออกเป็นคนแสดงความคิดเห็นและลองอ่านดูว่ามีจุดไหนดูแปลกไปหรือไม่ โดยงานที่ทำในตอนนี้จึงเอียงไปทางด้านของ User Interface ที่เดินหน้าไปได้มากพอสมควร
พิมพ์ 640610654	เป็นคนออกแบบหน้าเว็บของตัวเกม UPBEAT ซึ่งจะเป็นคนทำงานควบคู่ไปกับ ดิด ซึ่งก็จะถามความคิดเห็นกับทุกคนในกลุ่มว่าการออกแบบนี้ตรงตามที่ต้องการหรือไม่ หรือไม่ก็จะถามถึงส่วนที่จำเป็นต้องมีในหน้าเว็บ ทำให้ไม่ต่างจากคนที่ประสานการทำงานระหว่าง back กับ front
การเปลี่ยนแปลงหน้าที่ของคนในกลุ่มนั้นเกิดจาก การอธิบายการทำงาน code แล้วให้อีกฝ่ายไปเขียน test นั้น มีประสิทธิภาพการทำงานต่ำกว่าที่คิดเอาไว้ และในส่วนของทาง Frontend ก็มีปัญหานในเรื่องของไม่รู้ว่าจะออกแบบหน้าเว็บยังไงดี จึงเกิดการเปลี่ยนแปลงหน้าที่การทำงาน	
และการออกแบบหน้าเว็บนั้น ทำให้ได้รู้ว่าจะต้องมีข้อมูลจุดไหนที่ต้องการส่งไปยัง backend จึงทำให้เกิดการเชื่อมต่อระหว่าง frontend และ backend มากยิ่งขึ้น	
Known problems	
1 css ทำงานไม่ตรงตามที่ต้องการ	
2 map ไม่สามารถสร้างแบบที่คาดหวังได้ คือสร้างทีละ region เพราะตอนนี้สร้างได้ทีละ row แล้วเอามาต่อกัน	
3 ยังนึกไม่ออกว่าจะ implement api จาก backend ยังไง	
4 ยังไม่สามารถคิดได้ว่าการคำนวณเรื่องของคำสั่งต่างๆ นั้นต้องทำประมาณไหน	
5 จากการทำงานที่ผ่านมาคิดว่า project นี้จะสามารถทำงานให้เสร็จได้ตามเวลาที่จารย์ขึ้นได้กำหนดเอาไว้	