Decentralized Tangram

A game of co-operative tangram on distributed systems

Dafang Cao (j2d0b)
Eddie Zhou (u6x9a)
Max Wei (e3v8)
Michael Chen (e6y9a)
Stephanie Chan (g2u9a)

Table of Contents

Introduction **3**

Background **3**

System Topology 4

Trust 4

Problems and Approaches **4**

Clock Synchronization 4

Solution Checking 5

Client Behavior 5

Concurrency Issues 5

Testing **5**

SWOT Analysis **6**

Timeline **6**

References **7**

Introduction

The *tangram* is a tiling puzzle game invented in China in the Song Dynasty (960-1279) and introduced to Europe in the early 19th century. It consists of seven flat shapes, called *tans*, and the objective of the puzzle is to put together the shapes without overlapping to form a new shape. In electronic form, it is a game where a player can select a shape, move it around and rotate it before placing it in its correct position.

Background

There are many implementations of tangram available online and they offer standard single player gameplay. Multiplayer co-op is an aspect that we want to introduce into tangram because this will add complexity to the gameplay in terms of communication and coordination with other players. We have not found any multiplayer co-op versions of tangram, nor any implementation of the game in Golang, which indicates that a version of tangram with a multiplayer co-op feature has yet to be made.

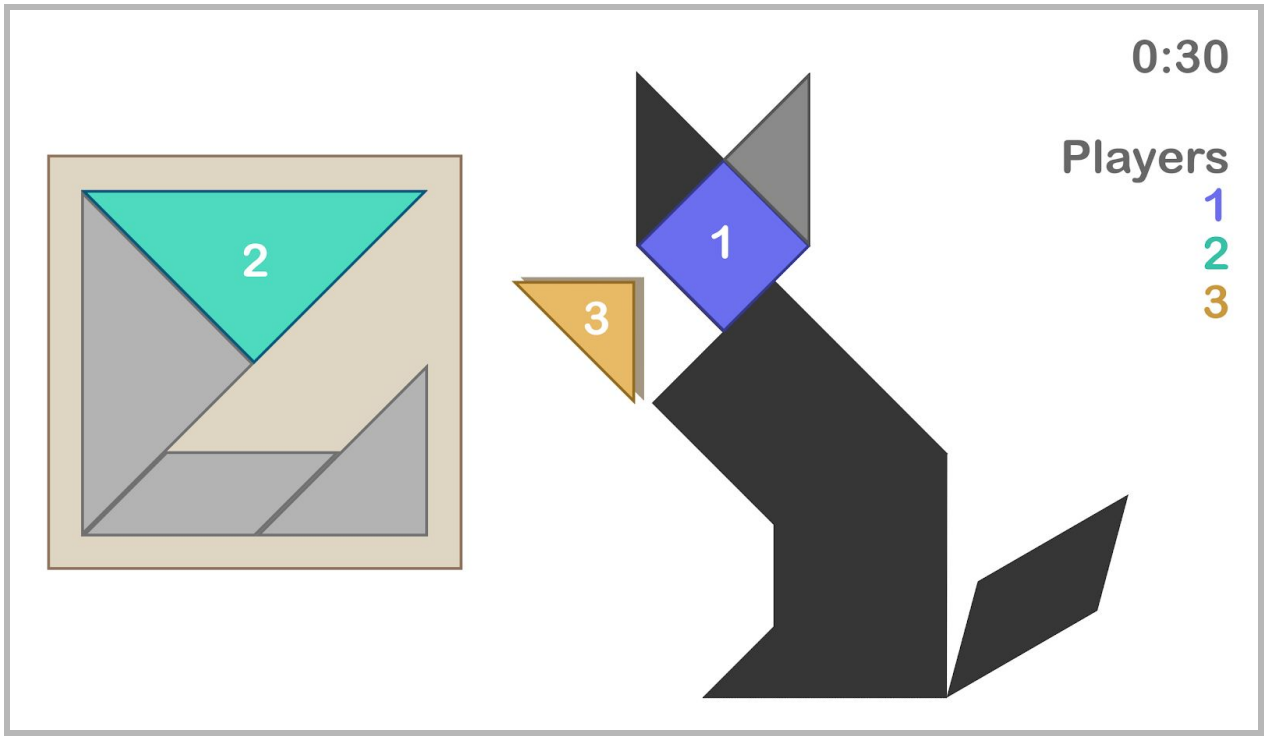
In Decentralized Tangram, our multiplayer co-op tangram game, each player will have to start their own client to start or join a game. The game can have one to seven players and players can join or quit at any point in the game.

The game state is held in every single client and every time the client does something to change the game state, it has to communicate the changes with all of the other clients.

A player can vote to kick another player out of the game. If the majority of players vote to kick out a certain player, that player will be kicked out of the game.

The game state consists of the following:

- The seven tangram pieces, ordered by last interaction, and their locations (x, y coordinate), rotation alignment (in increments of 5 degrees), and the player controlling the piece
- A countdown timer indicating the time remaining
- A list of players
- The shape to make
- Voting status (contains identifier of player to kick from game)
- List of kicked players to prevent rejoining



(Figure 1) Mockup of Game with 3 Players.

System Topology

The system will be a fully connected graph, where each client is aware of every other client within the system. Clients, upon connection to any node, will receive information about the other participants of the game. Each collection of nodes represents one tangram game.

Trust

The clients will be assumed to be trusted. That is, as the gameplay genre is co-operative puzzle and it is between individuals who have shared their IP and port to one another, the incentive for cheating is minimized.

Problems and Approaches

Clock Synchronization

Clocks across different clients need to be synchronized because of the countdown timer and shape piece movement. We will use vector clocks to determine the order of concurrent actions made by different players.

Solution Checking

To check a placed set of tans against the solution we will be using a set of equivalent solution areas that will be checked against the tans that have been placed. Since the square piece can be equal to two triangle pieces, and many pieces can be equivalent at any rotation, all of those conditions must be checked.

Client Behavior

- The player may specify, on client launch, the address of a another player to join their game. If the game already has the maximum number of players connected, the client will default to a new game. Omitting the address on client launch will launch a new game.
- Clients maintain heartbeats to every client in the game.
- On every game tick, each player sends out the state for the the piece it is manipulating to the other players, and updates its UI with any updates from other players.

Concurrency Issues

- If 2 players attempt to manipulate the same piece at the exact same time, the game should resolve the “winner” in a reasonable amount of time with minimal visual anomaly.

Testing

We will manually test various functionalities of our game. Examples of behaviour to test include:

- Single player tangram
    - General behaviour
        - Movement
        - Selecting and moving pieces
    - Clients are able to connect to this tangram client via websocket protocol
- Multiplayer tangram
    - Actions from other clients should register for every client
    - Timers are synchronized with a small margin of error
    - Client disconnections during a tangram game should not stall the game
    - Client reconnections during a tangram game should be possible

SWOT Analysis

	Helpful	**Harmful**
Internal	Strengths 1. One person in our group has experience in making games 2. Another person has experience in websockets 3. A third person in our group has an awareness of potential problems with distributed multiplayer games	Weaknesses 1. One person is new to the group so there is some uncertainty in group cohesiveness 2. All group members only started learning go at the beginning of this course 3. Nobody has worked with vector clocks before.
External	Opportunities 1. Currently there is no existing multiplayer co-op game for Tangram in Golang	Threats 1. Checking shape intersection and validation will be difficult unless we can find a specific library for it

Timeline

March 9th:
- Final Proposal due

March 13th:
- Start creating data structures for game

March 15th:
- Data structures due
- Split work into networking and UI (includes websockets)
- Create base for vector clock and time sync

March 21th:
- Send email to TAs to schedule a meeting on our progress
- Networking due
- Split work into game lag and time sync

March 26th:
- Game lag and time sync due
- Do testing

March 28th:

- UI due
- Start squashing all bugs

April 1st:

- All bugs should be fixed by today
- Start writing final report

April 6th:

- Final report and project code due

References

https://en.wikipedia.org/wiki/Tangram
https://www.mathsisfun.com/games/tangrams.html