Supplementary Information for Limitation of time promotes cooperation in temporal games

Jiasheng Wang, Yichao Zhang, Guanghui Wen, Jihong Guan, Shuigeng Zhou, Guanrong Chen, Krishnendu Chatterjee, Matjaž Perc

*Corresponding author(s). E-mail(s): yichaozhang@tongji.edu.cn; wenguanghui@gmail.com; jhguan@tongji.edu.cn;

Volunteers Recruitment and Experimental Setup

The experiment was carried out with a total of 183 volunteers. The participants are students mainly from Tongji University and Southeast University in China. All the participants of the experiment are required to register an account on the experimental platform in advance (see below). After logging in, they are asked to check the experiment schedule on the landing page, where the type of scenario is shown to the participants who are going to enroll in the match. Meanwhile, they can choose whether to enroll in the next match or not. When the match begins, only the participants who have enrolled can play. After kicking off a game, their accounts will automatically be redirected to the gaming page. For beginners, there is a casual mode for their training.

In the online experiments, participants played a traditional Prisoner's dilemma (PD) game, where C (cooperative) and D (defective) were the only available actions. Each participant interacted with the individuals who had agreements with him in one round, after which the agreements needed to be redrafted.

Each match on the platform comprises two stages. In the first stage, the system generates a network with a social network model. In the experiments, two types of networks are generated. One is Barabási and Albert (BA) scale-free network with degree $m_0 = m = 3$; the other is Watts and Strogatz (WS) small-world network with $P_{rewire} = 0.1$ and $K = 6$. There were 150 players participating in the experiment with the BA networks (56 for the 'divide-and-conquer' (D&C) games and 94 for the temporal games) and 99 players playing with the WS networks (55 for the D&C games and 44 for the temporal games). The subjects are then allocated to the nodes of the network. Thus, the connections among the subjects are randomly predetermined. The second stage is an n-round iterated PD game, where $10 \leq n \leq 30$ is unknown to individuals so as to avoid the ending-game effects.

In the match, participants are shown their identities, which are in-game-generated participant IDs. They are allowed to see their own gaming histories, where each record includes the actions (cooperation or defection) of both sides and the gaming time duration. Some necessary information about the game progress is visible to them, including food, time resources, neighbors, number of rounds, and time left for consideration.

After each match, the food resources of the players, namely payoffs, is the base for the reward. Each player gets 1 RMB for 1 unit of food resources as a basic reward. The top 3 players with the most food resources per round are the winners of the match. All the interaction logs of winners are opened to the participants, so they can vote for their favorite strategies. The winners who receive more votes can get more extra bonuses. The bonus pool is 1,000 RMB.

Experimental Platform and Interface

Today, there are many platforms designed for empirical experiments [1–9]. The most widely used is the z-Tree toolkits [1], proposed in 2007. It is used to perform social or economic experiments. But the questionnaire-like user interface cannot support complicated interactions, such as reconnection, chatting, etc. On the other hand, the system cannot support real-time interactions. Modern empirical platforms are quite different. One good example is nodeGame [2], which provides online service based on the browser/server (B/S) architecture. To recruit more participants, they normally connect with Amazon Mechanical Turk [10] (AMT). These platforms support real-time interactions to make the environment closer to real scenarios. Unfortunately, none of them can support the divide-and-conquer (D&C) gaming environment, let alone the temporal social dilemma process.

Overview

To implement the experimental scenarios of the temporal divide-and-conquer games, a novel online gaming experimental platform was developed in this work, called *War of Strategies*[1] (*WoS*). The features of the platform are listed below:

(1) Supporting D&C games. The platform provides an easy way to configure and conduct a D&C game experiment.

(2) Having built-in bots for training. Beginners can be familiarized with the platform by playing with the training bots. The current strategy of the bots is uniformly set to a random selection; that is, they will accept the gaming request with a probability 50% and cooperate with a probability 50%.

(3) Performing real-time interactions. The user interface is similar to a browser-based online game. The interaction between participants is real-time and stressful, stimulating the participants to make fast and cautious decisions.

(4) Having scalability. All modules of the experimental platform can be deployed on standalone servers or distributed machines. Docker containers are also supported.

(5) Having customizability. The gaming settings are easy to adjust to fit the models.

[1] http://strategywar.net

Architecture

The platform is developed based on several open-source software, composed of three components: Portal, Distributor, and Worker. The architecture is shown in Fig. 1.

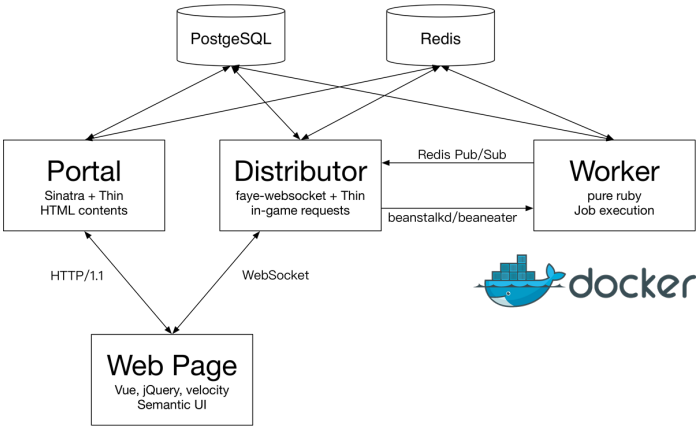

Fig. 1 Platform architecture of the WoS.

In the WoS, PostgreSQL [11] and Redis [12] are used to store platform data. PostgreSQL is responsible for data persistence, which manages the rarely updated data, such as configurations, user profiles, archived logs, and archived gaming results. Redis is used as an in-memory cache, storing the data which are read and written frequently, such as match data, participant data, in-game requests, and runtime logs. The Redis Pub/Sub message system supports the communications between the Worker and the Distributor module.

The Portal module deals with HTTP requests and web pages, such as the landing page. The module is developed on Sinatra [13] with Ruby [14], which is a lightweight web platform. Thin [15] is adopted as a web server. The module provides an authentication service and management interface. The module can be customized to provide a specific user interface for the participants and researchers. The default theme of the WoS is shown in Fig. 2.

During the experiment, the communication between the webpage and the server follows the WebSocket [16] protocol. In the Distributor module, the server communicates with a webpage by Faye-WebSocket [17]. Unlike traditional web-based applications, WoS adopts WebSocket to process real-time requests of the participants, including friend requests, gaming requests, chat messages, match processing data, etc. The Distributor module listens on the Redis Pub/Sub channel to process the request from the Worker module, for instance, broadcasting the match progress information.

The Worker module processes the delayed jobs, including starting the match according to the schedule, match process management, match result processing, etc. The module sends messages to the Redis Pub/Sub channel to notify the Distributor module. Since the match is conducted in the Worker module, one can modify this module to customize its functionality.

Fig. 2 Landing page of the WoS.

Fig. 3 Login page of the WoS.

User-Interface

Some screenshots of the default theme "the Lost Island" are provided, which are used in the experiment and shown to the participants. The volunteers first register or log in to participate in the experiment. The login interface is shown in Fig. 3. To register an account, a user requires an email address, a nickname, and a password. The section "Privacy Policies" covers the privacy issues related to the experiments.

Once successfully logged in, a participant will reach the landing page, as shown in Fig. 2. The left top of the screen displays the user's nickname and accumulated food resource, which is used to calculate rewards. The left panel shows the top 20 participants who won a match,

Fig. 4 The main part of the experiment.

The left panel shows:
Here is the request from [nickname] with time resource to cost: 1440
Would you accept this request?
Deny — Accept
The right panel shows: Day No. — Time resource used — My move — Partner's move

Fig. 5 The modal dialog box showing the request sent from a partner.

ordered by their average payoffs per round in the match. The right panel is the main panel, where the information from top to bottom is: “There is no pending match now. Please wait”, “Casual mode”, “Story”, “Help page”, “In order to have the best gaming experience, modern browsers including Chrome, Firefox, and Safari are suggested.” If there exists a pending match, the match schedule will be shown in the first line, followed by an enroll button.

In the experiment, the network topology is generated by a network model. After the generation of the social network, as shown in Fig. 4, the main process of the experiment begins. The status of the current match is shown on the top of the page, which is “Day 1, 33s left in the daytime”. On the left-hand side of the page, the upper panel shows the participant’s personal information, including the nickname, food resource, and the remaining time source. These properties are only visible to the participant himself; no one else can see them. The bottom-left panel is the operation panel. The participant can send gaming requests to their friends. The nickname is randomly generated, and the participant’s actual ID is hidden to clear the memory generated in the previous matches. The request can be canceled, accepted, denied, or ignored. The first line shows that a social request has been sent to a friend. The label on the button is “cancel”. The button can be used for withdrawing the request. The second line shows that a request has been accepted. As shown in the figure, the participant’s move is cooperation, and the assigned time resource is 720. The third line shows that the participant just received a request from a friend, where the label on the button is “request received”. The button can be used to trigger a modal dialog box for further operation, shown in Fig. 5. The fourth line shows that there is no interaction yet. Therefore the label on the button is “take action”. The button can be used to trigger a modal dialog box, as shown in Fig. 6. The buttons on its right-hand side show “check gaming history”, which can trigger a modal dialog box to review the gaming records.

Fig. 5 shows the modal dialog box triggered by request. The left panel is the operation panel and the right panel shows the gaming history. For integrity, a complete translation of the modal dialog box is provided.

Fig. 6 shows the modal dialog box of the drafting request. The left panel is the operation panel, and the right panel shows the gaming history. For integrity, a complete translation of the modal dialog box is provided.

Fig. 7 shows the modal dialog box when the participant accepts a request. Then, the participant should choose his move as a response. Note that the opponent’s move is not shown to the others.

For researchers, the WoS provides a user-friendly interface to manage experiments. The management interface provides the services of checking, creating, and editing experimental configurations, schedules, and exporting data. Fig. 8 shows the page for creating an experimental configuration. The bar on the top of the page includes two drop-down lists, which are “configuration” and “matches”. The drop-down items are related operations, such as checking, creating, and editing. The WoS uses JSON [18] to store the configuration. The figure shows a sample of the temporal social dilemma experiment. In the configuration, researchers can specify the payoff matrix, the duration of a round, the resource consumption, etc. The properties can be modified and created if the corresponding implementation is developed to conduct a customized experiment.

The left panel shows:
Choose a strategy to play with [nickname]
The move to take (Cooperate or Compete)
Assign the time resource. The more time resource you use, the more food you will gain in the same condition.
Cancel — OK
The right panel is the same as shown in Fig. 5.

Fig. 6 The modal dialog box of request composing.

Fig. 7 The modal dialog box shows that the participant accepts the request.

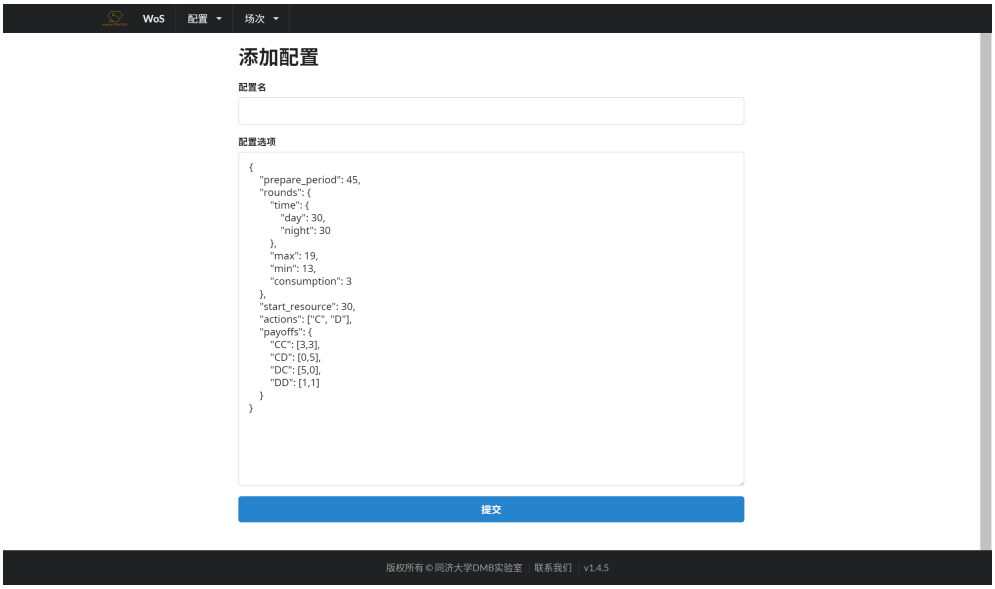

Fig. 8 The new configuration page of the management interface.

Privacy Policy

The privacy policy of the platform is shown at (http://strategywar.net/privacy), the details of which are listed in the following (in italics).

The data management and bioinformatics (DMB) laboratory of Tongji University is responsible for running and maintaining the http://strategy war.net website (the "Service"). This page informs you of our policies regarding data collection, usage, privacy protection of personal data, and corresponding options.

We use your data to provide and improve the Service. By using the Service, you agree to the collection and use of information in accordance with this policy. Unless otherwise specified in this Privacy Policy, the terms used in this Privacy Policy have the same meanings as those in our Terms and Conditions shown on http://strategywar.net.

Information Collection and Use

Several different types of information are collected, including personal data, usage data, and tracking & cookies data (the details are listed below) to provide and improve the Service to the user.

Types of Data Collected

Personal Data

When using the Service, the user may be asked to provide us with personally identifiable information that can be used to contact or identify the user ("Personal Data"). The personally identifiable information may include, but is not limited to:

- *Email address*
- *Cookies*
- *Usage data*

Usage Data

The information concerning how the Service is accessed and used ("Usage Data") will be collected. The Usage Data may include the user's computer Internet Protocol address (e.g., IP address), browser type, browser version, the pages of the Service that the user accessed, the time and date of the user's visit, the time spent on those pages, unique device identifiers, and other diagnostic data.

Tracking & Cookies Data

Cookies and similar tracking technologies are used to track the activity on our Service and hold certain information.

A cookie is a small file containing a string of characters that is sent to your computer when a user visits a website. When the user visits the site again, the cookie allows it to recognize the user's browser. Tracking technologies like beacons, tags, and scripts to collect and track information are applied to improve and analyze the Service.

The user can configure his browser to refuse all cookies or to set when a cookie is being sent. However, if the user does not accept cookies, he may not be able to use some functions of the Service.

Examples of Cookies used:

- ***Session Cookies.*** *Using Session Cookies to provide Service.*
- ***Preference Cookies.*** *Using Preference Cookies to record user preferences and settings.*
- ***Security Cookies.*** *Using Security Cookies for security purposes.*

Use of Data

War of Strategies uses the collected data for the following purposes:

- *To provide and maintain the Service*
- *To notify the user of changes to the Service*
- *To allow the user to participate in the interactive features of the Service when the user chooses to do so*
- *To provide customers with technical support*
- *To analyze the performance of the system for improving the Service*
- *To monitor the usage of the Service*
- *To detect, prevent, and address technical issues*

Transfer of Data

The user's information, including personal data, may be transferred to and maintained on computers outside his state, province, country, or other governmental jurisdiction where the data protection laws may differ from those of his jurisdiction.

If the user is located outside China and chooses to provide information for the Service, it should be noted that the data, including Personal Data, will be transferred to China and processed there.

The user's consent to this Privacy Policy, followed by his submission of such information, represents his agreement to that transfer.

War of Strategies will take all steps reasonably necessary to ensure that personal data is treated securely and in accordance with this Privacy Policy. No transfer of Personal Data will take place to an organization or a country unless there are adequate controls in place, including the security of the user's data and other personal information.

Disclosure of Data

Legal Requirements

War of Strategies may disclose the user's data only in the case that such action is necessary to:

- *Comply with a legal obligation*
- *Protect and defend the rights or property of War of Strategies*
- *Prevent or investigate possible wrongdoing in connection with the Service*

Security of Data

The security of the user's data is important, but remember that no method of transmission over the Internet or method of electronic storage is 100% secure. Although the means of data protection adopted in the Service is commercially acceptable, there is no guarantee of its absolute security.

Service Providers

A third-party companies and individuals may be employed to facilitate the Service ("Service Providers"), to perform Service-related services, or to assist in the runtime analysis of the system.

For the users' data, the third parties are only allowed to accomplish the tasks specified by the host and are obligated not to disclose or use it for any other purpose.

Links to Other Sites

This Service may contain links to other sites that are not operated by the host. If the user clicks on a third-party link, he will be directed to its site. The user is strongly advised to review the Privacy Policy of every site that he visits.

The content and privacy policies of the third parties are not under control. Accordingly, the host is not responsible for them.

Changes to This Privacy Policy

This Privacy Policy may update aperiodically. The user will be notified of any changes by seeing the new Privacy Policy on this page. The user is advised to review this Privacy Policy periodically for any changes. Changes to this Privacy Policy are effective when they are posted on this page.

Questionnaire for Volunteers

Volunteers were required to read the privacy policy and answer the questions. They would only participate in the experiment if they accepted the privacy policy. In the experiments, 40 valid answers were received, where 37 volunteers (92.5%) accepted the policy and became participants.

Table 1 The answer of acceptance of the privacy policy.

Answer	Count	Ratio
Yes, I agree	37	92.5%
No, I disagree	3	7.5%

Among the participants, 25 of them are male, and the male-to-female ratio is $2.08 : 1$.

The average age of the participants was 24.41 ($standardvariation = 3.77$). The detail of the age distribution is shown in Table 3.

The institutions of participants are listed in Table 4.

Table 2 The distribution of the sex.

Sex	Count	Ratio
Male	25	67.57%
Female	12	32.43%

Table 3 The ages of participants.

Age range	Count	Ratio
¡ 20	1	2.70%
20 - 24	23	62.16%
25 - 29	10	27.03%
$\geqslant$ 30	3	8.11%

Table 4 The institutions of participants.

Institution	Count	Ratio
Southeast University	21	56.76%
Beijing Institute of Technology	6	16.22%
Tongji University	5	13.51%
Others	5	13.51%

Experiment Procedures

In a round of the match, a player's operation flow is listed below:

1. Refill the time resource
2. Choose a friend to interact with
   (a) Choose your move and assign proper time resources to the game
   (b) Send request
   (c) Wait for response
   (d) Cancel the request if necessary
3. Handle the requests from friends
   (a) Deny the request if the remaining time resource is not sufficient
   (b) Accept the request and choose your move
   (c) Reject the request
   (d) Ignore the request
4. Repeat the above two steps until the decision-making time is up
5. Review the strategy and prepare for the next round.
6. Go to step (a)

Top Voted Strategies

The top three winning strategies in the temporal games are listed below.

Strategy 1

Round 1:

1. He/She sent request 'C' to 20047 with time resource 1440.
2. He/She played 'C,' and player 20047 played 'C' with time resource 1440.

3. He/She denied the request from player 20039.
Round 2:
1. He/She sent request ‘C’ to 20047 with time resource 1440.
2. He/She played ‘C,’ and player 20047 played ‘C’ with time resource 1440.
3. He/She denied the request from player 20039.
4. He/She denied the request from player 20024.
Round 3:
1. He/She sent request ‘C’ to 20047 with time resource 1440.
2. He/She played ‘C,’ and player 20047 played ‘C’ with time resource 1440.
Round 4:
1. He/She sent request ‘C’ to 20047 with time resource 1440.
2. He/She played ‘C,’ and player 20047 played ‘C’ with time resource 1440.
3. He/She denied the request from player 20039.
4. He/She denied the request from player 19982.
Round 5:
1. He/She sent request ‘C’ to 20047 with time resource 1440.
2. He/She played ‘C,’ and player 20047 played ‘C’ with time resource 1440.
Round 6:
1. He/She sent request ‘C’ to 20047 with time resource 1440.
2. He/She played ‘C,’ and player 20047 played ‘C’ with time resource 1440.
3. He/She denied the request from player 20024.
4. He/She denied the request from player 19971.
Round 7:
1. He/She sent request ‘C’ to 20047 with time resource 1440.
2. He/She played ‘C,’ and player 20047 played ‘C’ with time resource 1440.
Round 8:
1. He/She played ‘C,’ and player 20047 played ‘C’ with time resource 1440.
2. He/She denied the request from player 20039.
3. He/She denied the request from player 19982.
Round 9:
1. He/She sent request ‘C’ to 20047 with time resource 1440.
2. He/She played ‘C,’ and player 20047 played ‘C’ with time resource 1440.
3. He/She denied the request from player 19971.
Round 10:
1. He/She sent request ‘C’ to 20047 with time resource 1440.
2. He/She played ‘C,’ and player 20047 played ‘C’ with time resource 1440.
3. He/She denied the request from player 19982.
4. He/She denied the request from player 19971.
Round 11:
1. He/She sent request ‘C’ to 20047 with time resource 1440.
2. He/She played ‘C,’ and player 20047 played ‘C’ with time resource 1440.
3. He/She denied the request from player 20039.
4. He/She denied the request from player 20039.
Round 12:
1. He/She sent request ‘C’ to 20047 with time resource 1440.

2. He/She played ‘C,’ and player 20047 played ‘C’ with time resource 1440.
3. He/She denied the request from player 19982.
Round 13:
1. He/She sent request ‘C’ to 20047 with time resource 1440.
2. He/She played ‘C,’ and player 20047 played ‘C’ with time resource 1440.
3. He/She denied the request from player 20024.
4. He/She denied the request from player 19971.
Round 14:
1. He/She sent request ‘C’ to 20047 with time resource 1440.
2. He/She played ‘C,’ and player 20047 played ‘C’ with time resource 1440.
Round 15:
1. He/She sent request ‘D’ to 20047 with time resource 1440.
2. He/She played ‘D,’ and player 20047 played ‘C’ with time resource 1440.
3. He/She denied the request from player 20039.
4. He/She denied the request from player 20024.
5. He/She denied the request from player 20039.
Round 16:
1. He/She sent request ‘D’ to 20047 with time resource 1440.
2. He/She played ‘D,’ and player 20047 played ‘C’ with time resource 1440.
3. He/She denied the request from player 20039.
Strategy 2
Round 1:
1. He/She sent request ‘C’ to 19844 with time resource 720.
2. He/She played ‘C,’ and player 19902 played ‘C’ with time resource 120.
3. He/She sent request ‘C’ to 19890 with time resource 600.
4. He/She played ‘C,’ and player 19890 played ‘C’ with time resource 600.
5. He/She played ‘C,’ and player 19863 played ‘C’ with time resource 360.
6. He/She sent request ‘C’ to 19868 with time resource 360.
7. He/She sent request ‘C’ to 19873 with time resource 360.
8. He/She sent request ‘C’ to 19873 with time resource 360.
9. He/She played ‘C,’ and player 19868 played ‘C’ with time resource 340.
10. He/She sent request ‘C’ to 19896 with time resource 20.
Round 2:
1. He/She played ‘C,’ and player 19902 played ‘C’ with time resource 720.
2. He/She sent request ‘C’ to 19890 with time resource 720.
3. He/She played ‘C,’ and player 19863 played ‘C’ with time resource 360.
4. He/She sent request ‘C’ to 19868 with time resource 360.
5. He/She played ‘C,’ and player 19868 played ‘C’ with time resource 360.
Round 3:
1. He/She played ‘C,’ and player 19902 played ‘C’ with time resource 720.
2. He/She denied the request from player 19863.
3. He/She sent request ‘C’ to 19863 with time resource 720.
4. He/She played ‘C,’ and player 19863 played ‘C’ with time resource 720.
Round 4:
1. He/She played ‘C,’ and player 19902 played ‘C’ with time resource 720.

2. He/She sent request ‘D’ to 19890 with time resource 720.
3. He/She denied the request from player 19863.
4. He/She sent request ‘C’ to 19863 with time resource 720.
5. He/She played ‘C,’ and player 19863 played ‘C’ with time resource 720.
6. He/She had to deny playing with player 19873 due to a lack of time.
Round 5:
1. He/She played ‘C,’ and player 19902 played ‘C’ with time resource 720.
2. He/She sent request ‘D’ to 19890 with time resource 720.
3. He/She sent request ‘D’ to 19873 with time resource 720.
4. He/She denied the request from player 19863.
5. He/She sent request ‘D’ to 19863 with time resource 720.
6. He/She played ‘D,’ and player 19873 played ‘C’ with time resource 720.
7. He/She denied the request from player 19863.
Round 6:
1. He/She sent request ‘C’ to 19863 with time resource 720.
2. He/She played ‘C,’ and player 19863 played ‘C’ with time resource 720.
3. He/She played ‘C,’ and player 19902 played ‘C’ with time resource 720.
4. He/She denied the request from player 19890.
Round 7:
1. He/She sent request ‘D’ to 19890 with time resource 360.
2. He/She played ‘C,’ and player 19902 played ‘C’ with time resource 720.
3. He/She sent request ‘C’ to 19863 with time resource 360.
4. He/She played ‘C,’ and player 19863 played ‘C’ with time resource 360.
5. He/She sent request ‘D’ to 19896 with time resource 360.
6. He/She sent request ‘D’ to 19844 with time resource 360.
7. He/She sent request ‘D’ to 19868 with time resource 360.
Round 8:
1. He/She played ‘C,’ and player 19902 played ‘C’ with time resource 720.
2. He/She played ‘C,’ and player 19863 played ‘C’ with time resource 720.
3. He/She denied the request from player 19890.
Round 9:
1. He/She sent request ‘D’ to 19890 with time resource 360.
2. He/She played ‘C,’ and player 19902 played ‘C’ with time resource 720.
3. He/She denied the request from player 19863.
4. He/She sent request ‘C’ to 19863 with time resource 360.
5. He/She played ‘D,’ and player 19890 played ‘C’ with time resource 360.
6. He/She denied the request from player 19896.
7. He/She played ‘C,’ and player 19863 played ‘C’ with time resource 360.
Round 10:
1. He/She sent request ‘D’ to 19896 with time resource 720.
2. He/She played ‘C,’ and player 19902 played ‘C’ with time resource 720.
3. He/She played ‘D,’ and player 19896 played ‘D’ with time resource 720.
4. He/She denied the request from player 19863.
Round 11:
1. He/She played ‘C,’ and player 19902 played ‘C’ with time resource 720.

2. He/She sent request ‘C’ to 19863 with time resource 720.
3. He/She played ‘C,’ and player 19863 played ‘C’ with time resource 720.
Round 12:
1. He/She played ‘C,’ and player 19863 played ‘C’ with time resource 720.
2. He/She played ‘C,’ and player 19902 played ‘C’ with time resource 720.
Round 13:
1. He/She sent request ‘D’ to 19844 with time resource 720.
2. He/She played ‘C,’ and player 19863 played ‘C’ with time resource 720.
3. He/She denied the request from player 19902.
4. He/She sent request ‘C’ to 19902 with time resource 720.
5. He/She played ‘C,’ and player 19902 played ‘C’ with time resource 720.
Round 14:
1. He/She sent request ‘D’ to 19863 with time resource 1080.
2. He/She sent request ‘C’ to 19902 with time resource 360.
3. He/She played ‘D,’ and player 19863 played ‘C’ with time resource 1080.
4. He/She played ‘C,’ and player 19902 played ‘C’ with time resource 360.
Strategy 3
Round 1:
1. He/She sent request ‘C’ to 19840 with time resource 300.
2. He/She played ‘C,’ and player 19866 played ‘C’ with time resource 360.
3. He/She sent request ‘C’ to 19862 with time resource 360.
4. He/She played ‘C,’ and player 19904 played ‘C’ with time resource 360.
5. He/She played ‘C,’ and player 19897 played ‘C’ with time resource 360.
6. He/She played ‘C,’ and player 19862 played ‘C’ with time resource 360.
7. He/She denied the request from player 19840.
Round 2:
1. He/She sent request ‘C’ to 19840 with time resource 700.
2. He/She played ‘C,’ and player 19897 played ‘C’ with time resource 360.
3. He/She played ‘C,’ and player 19904 played ‘C’ with time resource 360.
4. He/She had to deny playing with player 19866 due to a lack of time.
5. He/She played ‘C,’ and player 19840 played ‘C’ with time resource 720.
6. He/She denied the request from player 19862.
7. He/She denied the request from player 19862.
Round 3:
1. He/She sent request ‘C’ to 19897 with time resource 360.
2. He/She sent request ‘C’ to 19904 with time resource 360.
3. He/She played ‘C,’ and player 19840 played ‘C’ with time resource 720.
4. He/She played ‘C,’ and player 19904 played ‘C’ with time resource 360.
5. He/She played ‘C,’ and player 19897 played ‘C’ with time resource 360.
6. He/She denied the request from player 19866.
7. He/She denied the request from player 19884.
8. He/She denied the request from player 19862.
Round 4:
1. He/She sent request ‘C’ to 19904 with time resource 360.
2. He/She played ‘C,’ and player 19904 played ‘C’ with time resource 360.

3. He/She sent request ‘C’ to 19897 with time resource 360.
4. He/She played ‘C,’ and player 19840 played ‘C’ with time resource 720.
5. He/She played ‘C,’ and player 19897 played ‘C’ with time resource 126.
6. He/She sent request ‘C’ to 19884 with time resource 234.
7. He/She denied the request from player 19862.
8. He/She sent request ‘C’ to 19862 with time resource 234.
9. He/She played ‘C,’ and player 19862 played ‘C’ with time resource 234.
Round 5:
1. He/She sent request ‘C’ to 19840 with time resource 1440.
2. He/She had to deny playing with player 19862 due to a lack of time.
3. He/She sent request ‘C’ to 19862 with time resource 700.
4. He/She sent request ‘C’ to 19868 with time resource 740.
5. He/She played ‘C,’ and player 19862 played ‘C’ with time resource 700.
6. He/She had to deny playing with player 19877 due to a lack of time.
7. He/She sent request ‘C’ to 19884 with time resource 740.
8. He/She played ‘C,’ and player 19840 played ‘C’ with time resource 720.
Round 6:
1. He/She sent request ‘C’ to 19890 with time resource 20.
2. He/She played ‘C,’ and player 19866 played ‘C’ with time resource 720.
3. He/She sent request ‘C’ to 19840 with time resource 700.
4. He/She denied the request from player 19897.
5. He/She sent request ‘C’ to 19897 with time resource 20.
6. He/She played ‘C,’ and player 19897 played ‘C’ with time resource 20.
7. He/She played ‘C,’ and player 19877 played ‘C’ with time resource 40.
8. He/She had to deny playing with player 19862 due to a lack of time.
9. He/She sent request ‘C’ to 19862 with time resource 660.
10. He/She played ‘C,’ and player 19862 played ‘C’ with time resource 660.
Round 7:
1. He/She sent request ‘C’ to 19904 with time resource 360.
2. He/She played ‘C,’ and player 19904 played ‘C’ with time resource 360.
3. He/She played ‘C,’ and player 19897 played ‘C’ with time resource 720.
4. He/She denied the request from player 19866.
5. He/She sent request ‘C’ to 19866 with time resource 360.
6. He/She played ‘C,’ and player 19866 played ‘C’ with time resource 360.
7. He/She denied the request from player 19840.
8. He/She denied the request from player 19862.
Round 8:
1. He/She sent request ‘C’ to 19840 with time resource 720.
2. He/She played ‘C,’ and player 19904 played ‘C’ with time resource 360.
3. He/She played ‘C,’ and player 19840 played ‘C’ with time resource 720.
4. He/She denied the request from player 19866.
5. He/She sent request ‘C’ to 19866 with time resource 360.
6. He/She played ‘C,’ and player 19866 played ‘C’ with time resource 360.
7. He/She denied the request from player 19862.
8. He/She denied the request from player 19877.

9. He/She denied the request from player 19897.
10. He/She denied the request from player 19877.
11. He/She denied the request from player 19897.
Round 9:
1. He/She played ‘C,’ and player 19897 played ‘C’ with time resource 1000.
2. He/She sent request ‘C’ to 19840 with time resource 440.
3. He/She played ‘C,’ and player 19862 played ‘C’ with time resource 360.
4. He/She denied the request from player 19877.
5. He/She denied the request from player 19890.
6. He/She sent request ‘C’ to 19890 with time resource 80.
7. He/She played ‘C,’ and player 19890 played ‘D’ with time resource 80.
Round 10:
1. He/She played ‘C,’ and player 19866 played ‘C’ with time resource 360.
2. He/She played ‘C,’ and player 19897 played ‘C’ with time resource 1000.
3. He/She denied the request from player 19862.
4. He/She sent request ‘C’ to 19862 with time resource 80.
5. He/She played ‘C,’ and player 19862 played ‘C’ with time resource 80.
6. He/She denied the request from player 19877.
Round 11:
1. He/She played ‘C,’ and player 19862 played ‘C’ with time resource 360.
2. He/She played ‘C,’ and player 19897 played ‘C’ with time resource 1000.
3. He/She denied the request from player 19884.
4. He/She sent request ‘C’ to 19884 with time resource 80.
5. He/She denied the request from player 19840.
6. He/She sent request ‘C’ to 19840 with time resource 80.
7. He/She sent request ‘C’ to 19904 with time resource 80.
8. He/She sent request ‘C’ to 19890 with time resource 80.
9. He/She denied the request from player 19866.
Round 12:
1. He/She played ‘C,’ and player 19897 played ‘C’ with time resource 1440.
2. He/She denied the request from player 19884.
3. He/She denied the request from player 19862.
4. He/She denied the request from player 19877.
5. He/She denied the request from player 19866.
6. He/She denied the request from player 19840.
Round 13:
1. He/She played ‘D,’ and player 19897 played ‘C’ with time resource 1440.
2. He/She denied the request from player 19866.
3. He/She denied the request from player 19862.
4. He/She denied the request from player 19840.
5. He/She denied the request from player 19877.
6. He/She denied the request from player 19866.
Round 14:
1. He/She sent request ‘D’ to 19840 with time resource 700.
2. He/She sent request ‘D’ to 19866 with time resource 500.

3. He/She had to deny playing with player 19904 due to a lack of time.
4. He/She sent request 'D' to 19904 with time resource 240.
5. He/She played 'D,' and player 19866 played 'D' with time resource 500.
6. He/She played 'D,' and player 19904 played 'C' with time resource 240.
7. He/She had to deny playing with player 19884 due to a lack of time.

References

[1] U Fischbacher, z-tree: Zurich toolbox for ready-made economic experiments. *Experimental Economics* **10**, 171–178 (2007).

[2] S Balietti, nodegame: Real-time, synchronous, online experiments in the browser. *Behavior Research Methods* **49**, 1696–1715 (2017).

[3] RX Hawkins, Conducting real-time multiplayer experiments on the web. *Behavior Research Methods* **47**, 966–976 (2015).

[4] J Pettit, D Friedman, C Kephart, R Oprea, Software for continuous game experiments. *Experimental Economics* **17**, 631–648 (2014).

[5] UD Reips, C Neuhaus, Wextor: A web-based tool for generating and visualizing experimental designs and procedures. *Behavior Research Methods, Instruments, & Computers* **34**, 234–240 (2002).

[6] MH Birnbaum, Surveywiz and factorwiz: Javascript web pages that make html forms for research on the internet. *Behavior Research Methods, Instruments, & Computers* **32**, 339–346 (2000).

[7] F Cole, Understanding knowledge as a commons: From theory to practice. *The Electronic Library* **25**, 630–631 (2007).

[8] K Lakkaraju, et al., The controlled, large online social experimentation platform (close) in *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*. (Springer), pp. 339–344 (2015).

[9] DL Chen, M Schonger, C Wickens, oTree–An open-source platform for laboratory, online, and field experiments. *Journal of Behavioral and Experimental Finance* **9**, 88–97 (2016).

[10] Amazon mechanical turk (https://www.mturk.com/).

[11] Postgresql: The world's most advanced open source database (https://www.postgresql.org/).

[12] Redis (https://redis.io/).

[13] Sinatra (http://sinatrarb.com/).

[14] Ruby programming language (http://www.ruby-lang.org/en/).

[15] Thin - yet another web server (http://code.macournoyer.com/thin/).

[16] I Fette, A Melnikov, The websocket protocol, (Google Inc. and Isode Ltd.), RFC 6455 (2011).

[17] Github - faye/faye-websocket-ruby: Standards-compliant websocket client and server (https://github.com/faye/faye-websocket-ruby).

[18] Json (http://www.json.org/).