Creating Valid Temporal-Spatial Digital Implementations of The Island Game - An Addendum to ‘A Framework for Creating Simulations To Study Human Goal-Driven Resource Use: The Island Game’

Authors: Jen Schellinck, Ken Webb
Version 1: Written 2025-04-23, Published 2025-12-03

Introduction

This document provides considerations to be taken into account when digitally implementing physical elements of The Island Game - time, space and related elements like energy, matter and action. These considerations are particularly important when the implementation is being used to validly model human and other agent resource use behaviours and by extension support the generation of useful hypotheses about these behaviours.

A previous technical report, ‘A Framework for Creating Simulations To Study Human Goal-Driven Resource Use: The Island Game’ (referred to subsequently as The Island Game Technical Report) described an implementation neutral framework for creating versions of The Island Game, a play-based simulation for studying goal-driven resource use in humans and other agents. In The Island Game, participants include both players (islanders) and a controller. Players interact with other players or carry out resource-use-related activities (e.g., collecting materials, building structures) on an island. The controller manages the game on behalf of the players, updating the island environment over time based on the effects of the players’ chosen actions and the presence of other game mechanics (e.g. updating plant growth over time).

The Island Game Technical Report includes an example manual implementation of The Island Game framework, presented as a tabletop game. The framework is also intended to support digital implementations, where some or all of the game is carried out on or by computers, in a simulated digital environment. In such digital implementations of The Island Game, where the controller is a digital rather than a human agent, and the players (islanders) may or may not be digital, specific consideration must be made with respect to how to validly implement time and space in the simulation, as well as other related physical elements like energy, mass and movement.

Attending to these considerations will allow for the creation of useful hypotheses about human (and other agent) resource behaviours, while at the same time taking advantage of the additional capabilities offered by computers in order to create digital implementations that are more usable and realistic-seeming from the perspective of participants playing the game.

While the focus of this document will be on The Island Game, the document will also touch on simulation best practices more generally, with respect to modelling time, space and related elements in a consistent and valid manner.

Time, Space, Energy, Mass and Movement

In the real world, the relationships between time, space, energy, mass and movement (or action) place constraints on how organisms can behave in a given situation. For example, the speed at which a person walks is related to both the amount of energy available to that person and their weight, among other factors. Walking speed will, in turn, determine how much distance the person can cover in a given amount of time. Or, if the speed at which a human walks is kept constant, this will determine how much energy is required while walking, to maintain that speed, as well as how much space can be covered and how much time is required to cover that space. Defining the value of one of these element constrains the values of the others.

Because these elements are not independent of each other, collectively modelling them in a valid and consistent way is important if the model is to be used for high quality hypothesis generation (i.e., the generation of hypotheses that have a higher probability of being more accurate or correct than alternative hypotheses). This valid modelling is particularly important when the hypotheses of interest directly involve space, time, energy or movement, as is the case when modelling resource use. In such cases, since the modelling of agent action (e.g., movement across space) is not independent from how space, time, energy and mass are modelled, the modelling of agent actions should constrain, and/or be constrained by, how time, space, energy and mass are represented in the model.

Manual vs Digital Implementations of The Island Game

Validity in modelling is accomplished, in part, by connecting model implementations (subsequently referred to more briefly as implementations) to their target phenomena through the intermediary of a conceptual model. Here, the conceptual model is an abstraction of both the implementation and the target. The conceptual model serves to map the mechanisms, structures, properties and processes of the implementation to the mechanisms, structures, properties and processes of the target.

It is possible to have multiple conceptual models involved in the modelling process. For example, in the case of The Island Game, the conceptual model of the phenomenon of interest- i.e. human resource use - is mapped onto a game, The Island Game, which is itself abstract and conceptual prior to implementation. This element of abstraction is a common feature of games. For example, a game of Monopoly may be physically implemented in different ways (e.g. as a table top game or on a computer) so long as the structure and rules of the game are maintained.

In the case of dynamic phenomena, in addition to these general requirements for validity, models and their implementations must:

(1) model cause and effect in a manner that is sufficiently realistic relative to hypotheses of interest.

(2) within this context, model simultaneous and near simultaneous interactions in a way that is sufficiently realistic relative to the phenomenon being modelled.

Because of the causal focus, a key aspect of validity for dynamic models is consistent mapping of time, space, energy, matter and action between the implementation and target, as well as across the implementation itself. In the Island Game Technical report, a demonstration manual implementation of the Island Game is provided that includes an approach to realistically modelling time, space, energy and player interactions during manual game play. This approach involves having game players take turns and, within these turns, having certain numbers of time units available. Actions take a certain numbers of time units and require a certain amount of energy. Game time is synchronized with real time through the turn taking mechanism.

When a specific version of The Island Game is constructed for research purposes, the game designer for that particular game ensures that the energy and time units required for an action in the game realistically map, in some way, onto the energy and time required for the real action, and that the chosen values are also consistent with the time and energy required for other available actions in the game. The game designer also defines any spatial parameters in the game - most importantly, the size of game tiles- in such a way that they are consistent with the choices made around actions, energy, mass and time. Then, during game play, the controller synchronizes the behaviour of the agents and their impacts on the environment to ensure temporal consistency and realistic cause and effect.

In such a manual situation, the speed of play is limited by the speed at which the players and the controller can act and carry out the game mechanics. This in turn, places constraints on the relationship between game play time (the real world time that it takes participants to play the game) and simulated time (i.e., how much simulated time one turn represents). For example, suppose that it typically takes approximately 15 minutes of game play for one turn to take place, in order to carry out all of the game mechanics. If this amount of game play is considered to realistically represent two hours of simulated time, then the actions of players during the turn should be ones that also realistically take two hours of time or less.

This choice further sets the scale of other game elements. For example, the space represented by one game tile should be set relative to the time the game states it takes players to move from tile to tile, to reflect a realistic movement rate. If a turn is set at two hours of time, and a player can move across a tile in one turn, then the size of the tile should reflect a realistic amount of distance that can be covered in that time.

The choice of how time is mapped to turns also limits how much time can practically be modelled during manual play. In the example above, if players play the game for 5 hours of game play time, this is equivalent to 40 hours of simulated time. If the phenomenon being modelled does not exhibit relevant behaviour in a simulated 40 hour time span, then the game will not provide useful simulated data during five hours of game play.

As well, this mapping choice may impact player experience, which should not be neglected. Human players are likely to refuse to play the game for any length of time if nothing of interest happens to them or because of them during the game play. Thus, participant experience is another constraint that must be taken into account when designing a particular implementation and determining how it maps onto the conceptual model, although this consideration should not come at the expense of model validity if the game play is being used for research purposes.

Digital Implementations - Specific Considerations for Validity

Physically implemented dynamic models are referred to as simulations, with the term often denoting both the conceptual model and the implementation of the model. Here, because the implementation and the target are both physical, they each have their own independent realities with respect to time, space and other elements. For example, implementation processes take a certain amount of real time to occur just as the target phenomenon processes take a certain amount of real time to occur, independent from the implementation time. The conceptual model serves to map the reality of one onto the reality of the other, abstracting away from, and smoothing over, any irrelevant, fine-grained aspects of each.

In digital implementations of The Island Game, as well as in digital implementations of models and games more generally, it is expected that the implementation will be set up within a program loop that repeats until the game play or simulation is ended. Program commands carried out during each iteration of this loop represent the actions of the simulation or game, including agent or game participant actions. All actions are carried out within the program loop at a speed determined by the nature of the program commands and the capabilities of the hardware running the program.

In a digital implementation of The Island Game, the digital controller can carry out game actions and update the environment much more quickly than a human controller. For example, suppose completing the required game actions on a turn takes the controller approximately 1/10th of a second. Ignoring, for the moment, the time required for the players to choose their actions, if 1/10 of a second is mapped on to 2 hours of simulated time, five hours of game play will represent roughly 41 years of simulated time and, assuming the simulation is properly constructed, 41 years of real, phenomenon time.

Implementing The Island Game entirely digitally by making both the controller and the players digital agents, effectively turns The Island Game into a multi-agent simulation (MAS). This allows for potentially much faster game play relative to a fully manual implementation. A game turn, equivalent to a “time step” in a MAS, might still take approximately 1/10 of a second of processor time, even including all player and controller actions. This provides the possibility of simulating a much greater amount of phenomenon time. In such an implementation, in order to realistically model time, space, and other other elements, the same approach that is taken in a manual implementation can be taken in the digital implementation. The only difference in the case of the digital implementation is the speed at which the turns take place.

Human-digital hybrid play, where some or all of the game players are human, but the controllers is digital, introduces additional complications. There are at least three ways to approach such an implementation with an eye towards validity. One option would be to keep a turn based approach. In such a case, regardless of processing speed, the game would be set up so that all play would take place within the context of synchronized turns. The second option would be to set the relationship between game loop time and simulation time so that all constraints were balanced to allow for both validly modelling a particular phenomenon and for a reasonable game play experience. Taking this approach opens up the option for simultaneous rather than turn based play. The third option would be to have some aspects of the game occur in a simultaneous play context and some in a turn taking context.

Consider first the approach where the digital implementation simply uses the exact same mechanisms that are recommended for manual game implementation. In such a 'manual-digital' implementation, the game proceeds, as with the typical manual implementation, in defined turns, each of which represents a certain amount of time passing. Islander activities are synchronized within and across turns by having the controller first receive intended actions from all the islander players during each turn. The controller then implements these actions on behalf of the islanders. In the process the controller ensures that all islander actions are resolved in the proper sequence and, similarly, that the island environment is properly updated in the correct sequence. The controller also deals with any simultaneous player actions that might occur during the turn using the approach described in The Island Game Framework, in order to ensure that cause and effect are properly simulated.

There are some disadvantage and challenges associated with this approach. From a simulation perspective, it fails to take advantage of useful additional capabilities offered by digital implementations to increase the realism of the simulation (e.g. by having the environment change in a more realistic manner over time). As well, with respect to the experience of the participants, it imposes potentially unnecessary restrictions on their game play. On each turn, players must still wait until all other players have chosen and provided their actions to the controller, which may take some time. This limits the speed at which the game can be played, which will likely decrease player interest, as well as decreasing the amount of time that can easily be simulated by the game. One possible advantage is that it reduces the role that the speed of player actions can take in the simulation outcome, which could be important if some players are human and some are digital.

In hybrid situations the speed of the digital controller opens up the possibility of relaxing the turn taking approach and allowing players to act more spontaneously. In this case, players can effectively carry out actions whenever they wish. The digital controller can detect the button pushes of any of the game players nearly instantaneously, relative to the human speed of action, and equally as quickly adjust the simulation based on these actions, ensuring that the environment is correctly adjusted in enough time for any other players to perceive the correct state of the environment and act on that basis in real time as well.

In this case, the human players are constrained by the time it takes them to perceive their environment and decide how to respond to their current situation, as well as how quickly they can physically press a key or button (their physical reaction times).

Because of this, a default assumption can be that the adjustments to the environment made by the controller will occur quickly enough that human players will not act on “stale information” even in the absence of synchronizing turns. If this assumption causes concerns in a particular implementation, the synchronization can be enforced programmatically by not allowing players to choose actions unless the game environment is properly up-to-date.

In those rare cases where two human players are both within range to impact each other (either on the same tile or within perceptual distance of each other) and also do happen to act simultaneously, for example by both pushing an action button in the same program loop, the situation can be resolved by reverting briefly to the turn taking approach similar to that used in a non-hybrid (all human or all digital participants) situation. For example, suppose an agent, agent A, moves into view of another agent, agent B, at the exact same time step that that agent B decides to take an action. In this case it would be possible, if agent B were to learn about the presence of agent A before their action, that they would take a different action. Since both have acted simultaneously, the controller must make a choice about which agent’s action happens first (e.g., the controller might choose randomly), and update the environment based on that action, pushing the other agent’s action into the next time step.

In such implementations it is further recommended that all actions can be interrupted in a spontaneous fashion. This functionality is analogous to the ability of players to interrupt their actions during a turn in manual, turn based implementations.

Because in spontaneous play there are no turns, a time mapping cannot be established by stating how much time a turn represents. The question becomes instead how to map modelled time onto the program loop time length. Importantly in this case, the program will most likely loop substantially faster than human players will be entering game commands. Because of this, if each program loop is presumed to represent the passage of a particular amount of model time, and by extension phenomenon time (e.g. 15 minutes), substantive amounts of model time may pass while human players are doing nothing. For example, suppose the program loop typically takes 1/10th of second to complete. If a program loop is defined as representing 15 minutes, and a human player does nothing for ten seconds of real time, then approximately 25 hours of simulation time will have passed in the game while the player does nothing.

In general, how the relationship between program loop time and model time is defined can have substantial implications with respect to both the game experience of the players and the validity of the model, as well as what can realistically be modelled. Because of this, it is important to carefully consider the relationships of all relevant elements during construction of the simulation and set up the mappings and values of these elements in such a way that they are properly balanced with respect to all of the considerations mentioned above.

As an example of a balanced mapping, consider a situation where a primary constraint is that human players are comfortable taking actions roughly every five seconds, although they may under some circumstances choose to act more slowly or more quickly. Further suppose that typical program loop time length is 1/10th of a second. This means that approximately 50 program loops will typically go by between actions taken by a human player.

Now suppose that, from a simulation perspective, relative to the phenomenon of interest and the desired granularity of actions being modelled, five seconds of simulated time could feasibly represent 15 minutes of actual time (with 1 second representing 3 minutes of actual time), relative to the desired granularity of actions in the simulation. This means that each program loop, and by extension time unit of the game, should be defined to represent 18 seconds of model time passing.

In this situation, players may still take more or less time to act, and they can still act spontaneously. Actions that they start should take a defined amount of play time, and be assigned a certain number of time units, that is consistent with the implementation-model relationships that have now been set up. For example, if a player eats a meal, and the modeller wants eating a meal to take the equivalent of 20 minutes in real time, this should take 66 time units, which would be expected to take roughly 6.6 seconds of implementation/game time, given that typical program loop time length is 1/10th of a second. Players must then wait for time to pass in-game while eating a meal. To maintain player experience, this aspect of the game may require the game designer to define actions with enough granularity to keep players actively playing the game rather than waiting for long periods while actions complete.

Spatial elements of the game, such as the size of game tiles in the implementation and then, by extension, the size of objects on the tiles, should also be set accordingly. If agents in the implementation are defined as being able to move across tiles in a certain number of time steps/time units, this will effectively define the amount of space being represented by a tile in the game. For example, if it takes the agent 10 time units to move across a tile, then this means that it takes 1 second of game time representing 180 seconds of model time to move across the tile. If the desired movement rate of players is 3km/hour, based on the phenomenon being modelled, this means a tile is constrained to be 150 meters in length. As well, how the environment is modelled and changes to the environment must be consistent with these established relationships. For example, the time it takes items to grow or change should be consistent with the amount of time passing, and the density of objects on a tile should be consistent with the defined size of the tile and the density of the objects in the phenomenon of interest.

In circumstances where there are major differences between player speed of play - for example, when some players are digital and some are human, or when some players have substantial experience with the mechanics of The Island Game and some do not - it might be necessary to implement a combination of spontaneous and turn based play. For example, if digital agents have an interface to the game that lets them play much more quickly than human players, it could be convenient to impose a semi-turn based approach on all players where they can only perform actions at a rate that is pre-defined. The rate of action selection might also be partially coordinated across all players, such that faster players may have to pause until slower players have carried out a certain number of actions or a certain amount of game time has passed.

Conclusion

Manual, fully digital and hybrid human-digital simulations all require deliberate approaches for modelling time, space and other related elements (energy, mass, action) in order to ensure the validity of the model. When The Island Game is being used as a research tool, models and implementation of the Island Game must take these relationships into account . It is likely that taking the approach of mapping the relationship between game loop time and model time in such a way that that all constraints are successfully balanced, while still allowing spontaneous play will result in both the best player experience and the most realistic game play, but the feasibility of this approach will also depend on what is being modelled. It is important to take all relevant elements into account when determining digital implementation details.

Appendix - Defining Terms to Support Valid Implementation-Target Mapping

When mapping and discussing the relationships between the target phenomenon, the model and the implementation of that model, it can be useful to have some clearly defined terminology, as follows:

Implementation time: Refers to time or the passage of time associated with the events, behaviours, processes and mechanisms required to run a model implementation. For example, in a digital implementation, one iteration of a program loop might take 1/8th of a second of implementation time.

Phenomenon time: Refers to time or the passage of time associated with events, behaviours and processes of the target phenomenon. For example, if the phenomenon of interest encompasses human movement, you could say that a human walking 5km might take 20 minutes, in phenomenon time. Will often be used hypothetically, as in: If this model is correct, that would take 10 years, in phenomenon time.

Real time: Refers to the actual passage of time. Depending on the context this term can be equivalent to either phenomenon time or implementation time. For example, one iteration of a program loop might take 1/12th of a second in real time (and implementation time), and a human walking 5 km might take 20 minutes in real time (and phenomenon time).

Game time: Refers to the passage of time when discussing implementations of The Island Game. For example, in a manual implementation of the Island Game, completing a turn could take 15 minutes of game time. Typically used when talking about player experience.

Model time: Refers to time as represented in the conceptual model. For example, a conceptual model might define eating a meal as always taking 30 minutes, in model time.

Simulation time: Here defined as equivalent to model time. However, as the term simulation is ambiguous, in that it is sometimes used to refer to a model and sometimes used to refer to an implementation, it is recommended to avoid using this term.

Program loop: as discussed above, this is the iterative loop that the digital implementation of the island game runs through to process the actions of the agents and update the environment. In the case of a hybrid situation with human players and a digital controller, the program loop also detects the interactions (e.g. button presses) of the human players.

Program loop time length: the amount of implementation time the digital implementation takes to run through one iteration of a program loop. This time may vary from loop to loop, depending on the processor load, the state of the environment and player actions. For example, one iteration of the program loop might take 1/10th of a second (in implementation time).

Typical program loop time length: an amount of time representing the typical amount of time required for an iteration of the program loop. This could be an approximate or rounded value.

Time step: Terminology used in MAS to refer to the program loop in the context of a MAS, as well as what the program loop represents in the MAS conceptual model. Regardless of the program loop time length for a given iteration of the program loop (sometimes an iteration might take, for example 1/10 of a second and other times 1/12 of a second in implementation time) each time step is viewed as representing a certain constant amount of time (in model time).

Time step time length: The amount of time represented by a time step. For example, one time step, may represent 15 minutes or 20 years (in model time). If the model is well constructed, this will also represent the same amount of time in phenomenon time

Time unit: An Island Game element that represents time in The Island Game. Actions take certain numbers of time units.

As an example of how some of these terms are used in context, suppose that a digital implementation of a model can process program loop iterations at a rate of approximately 10 program loop iterations per each real/implementation time second. Here, the exact number of program loop iterations per second will vary depending on the amount of processing required for a given program loop iteration, but suppose that in this case, the typical program loop time length is 1/10th of a second, implementation time. Viewing the digital implementation as a MAS, each program loop iteration could represent a time step that is defined to always be 6 months in duration, model time. Running the implementation for 5 hours of real time results in approximately 180 000 time steps, which would be 90,000 years of model time. Assuming the model is valid, this also represents 90,000 years of phenomenon time.