

Best Practices for Designing a TaleBlazer Game

Starting Out

Identify and incorporate your game goals from the outset.

Before you begin, use the Game Worksheet or your own preferred type of document to capture your ideas for the game.

Why are you making this game? What are the goals you have for the players? Do you want them to learn particular content? Do you want to increase their interest or awareness of a particular topic? Do you want them to visit a particular place?

Keep these goals in mind from the outset as you design your game. The key interactions and decisions for your players in the game should be centered around these goals.

Game Content

Text

Keep text content manageable for player to read and absorb.

It's much more difficult for players to read while playing an AR game than while sitting at a computer. Glare from the sun or even from a cloudy sky can make it difficult to physically see text, especially on a small screen. Besides, the outside world is inherently distracting. On the other hand, that's the whole point, isn't it? You don't want the player to spend all of their time reading text on a screen – they can stay home and do that. You want the player to look up from the screen and make connections between the game content and the real world around them. This is hard to accomplish if there is too much text.

- Edit the text to make it clear and concise. Err on the side of brevity.
- Make sure the reading level is appropriate for the target audience.
- Consider breaking large amounts of text into smaller chunks and spacing it out more along the game. For example, use multiple agents.
- For optimum accessibility, consider making videos to accompany text. See the section on videos.

Images

Include clear high contrast images.

TaleBlazer allows you to upload an image for each agent and role as well as for the game itself and for the world tab. Outdoor glare can make it surprisingly difficult for the players to see these images. High contrast images with larger shapes and details work better than low contrast images with subtle shapes and fine details.

Keep backups of your original image files. TaleBlazer compresses uploaded images. It is possible that future versions of TaleBlazer will support larger images as mobile devices evolve. In that event, you may want to re-upload your original image files to that future version of TaleBlazer to take advantage of the improvements.

Videos

Consider using videos to make your game accessible to a wider audience.

Vision, age, language, and literacy issues may make text content less accessible to certain players. Videos can mitigate this problem. Our user testing has shown that even very simple videos comprised of someone reading the text aloud accompanied by the static image of the agent and sub-titles along the bottom can be quite effective. Keep in mind that producing videos should be among the last steps of the game development as the text is likely to change frequently during the design process. Note that TaleBlazer only allows videos to be played as a ‘video action’ – the player has to press a button on a dashboard to play a video.

Reviewing Content

Allow your players to review important content.

If the game includes information or directions that the player needs to remember, careful design of the game is needed to ensure the player will be able to access critical information when it is needed. There are some intricacies of the TaleBlazer platform to keep in mind. By default, TaleBlazer only includes the ‘Game’ tab and the ‘Map’ tab. Neither of these tabs allows the player to review information effectively. To allow the player to review information, you must include one or more of the following tabs.

History Tab

The most straightforward way to allow players to review information is to include all important information and directions as text or video actions on agents (or in the agent name or description) and include the ‘History’ tab in the game.

The history tab shows a chronological list of all visited agents starting with the most recently visited agent at the top. When the player taps on an agent, they can see the agent dashboard – which shows the name, image, and description of the agent. It also shows a snapshot of the traits and actions as they were when the player bumped the agent, with one exception. Script actions are NOT visible from history. This is to maintain consistency with the history metaphor.

IMPORTANT: Avoid using the ‘say’ blocks to convey critical information. ‘Say’ blocks are executed inside scripts. Scripts cannot be executed from the history tab so the player will not be able to review information provided via the ‘say’ block from the history tab. Instead use description text or text actions.

NOTE: The introduction is included on the history tab; the player will be able to review the introduction if and only if you include the History tab in the game.

NOTE: The game designer can opt to exclude certain agents from the history tab by unchecking ‘Show agent’ in the ‘History Tab’ section of the ‘Bump Settings’ dialog. This setting cannot be changed during game play. There is no block to hide or show an agent on the history tab.

Inventory Tab

The game designer can opt to use the inventory tab to highlight a short list of important agents (and all their relevant information) to the player.

Think of it as an evidence collection bag. The inventory tab shows the agents in the player’s inventory and looks very similar to the history tab except that tapping on agents in the history is counted as a ‘bump’ – the bump script is executed and the current traits and actions are shown and fully accessible - including script actions. Perhaps the more critical difference to this use case of the inventory is that the game designer can use blocks to control what the player sees in their inventory while this is not possible for the history tab. The pick up and drop blocks add and remove agents from the inventory; the include and exclude blocks show or hide agents in the player’s inventory. For example, certain agents could represent critical facts. When the player bumps into such agents, the bump script would include a pickup block to automatically move the agent to the player’s inventory. The player reviews the important facts by looking at their inventory tab. Using the inventory tab to present a short list of important agents can prevent information overload for the player.

Keep in mind that using the inventory tab for multiple purposes may be confusing to the player. If the player is supposed to collect tools AND evidence, this may be harder for the player to grasp in the context of a single game experience.

World and Player Tabs

The World and Player tabs can be used to provide small amounts of important information to the player.

The player (role) tab and the world tab provide the player with access to the name, image, description, traits, and actions for the player or for the world respectively. These tabs are one level more accessible to the player than the agents on the history and inventory tab. Using these tabs for accessing information works particularly well for data that change over the course of the game. For example, use the player tab to show the player’s score (as a trait) and the player’s current quest as the description.

Navigation

Placing the Game

Pick a manageable area for your game to cover.

- Adjust the spacing of the game to cover a large enough area to be interesting, but small enough to be manageable.
- Consider including large visual landmarks such as nearby buildings within the Region(s) to help the player with navigation.

Placing the Agents

Make sure the agents in your game are easily and safely accessible by pedestrians.

- Make sure the agents are located in safe areas on flat level ground - away from stairs, busy roads and bodies of water.
- Whenever possible, make the agents accessible to players in wheelchairs or pushing strollers.
- Avoid placing agents near tall buildings that might interfere with GPS accuracy.
- Test your game early and often to make sure that the agents can be reached easily.
- Make sure that the scale of the game is appropriate to the target audience. Expecting the players to climb a large hill or travel long distances between agents may sabotage the success of the game.
- Don't require the players to backtrack too much. This is usually needlessly frustrating to the players and can sabotage the success of the game.
 - Avoid requiring the player to return to a centralized agent multiple times. Instead, use multiple agents at multiple locations or use the 'move' block to move the important agent to join the player at their own location.
 - Avoid requiring the player to zigzag back and forth to reach all the agents.
 - Consider sequencing the game to lead the players on the most efficient path.

Custom Map

Use a custom map to overcome limitations with the satellite view. The most common user complaint is difficulty navigating. The default Google maps satellite view is not suitable for a player navigating pedestrian pathways in an unfamiliar location. Instead, include a custom map for your game.

- Start with a satellite map by using the 'Capture Image' button on the Map tab in the editor.
- Once you have captured the map, edit it before uploading it into your game. Use Adobe Photoshop or a freeware alternative such as Paint .NET.
 - Lighten the map and increase the contrast to make the details more visible outdoors
 - Make layers for the buildings and pedestrian pathways and draw them over the map, again using high contrast colors.
 - Finally, upload the custom map to the Region.
- Test your game by walking around yourself to make sure the details on the custom map are indeed visible while playing the game

Narrative/Player Engagement

Provide the player with an explicit and specific goal at the beginning of the game to capture their interest and frame their game experience.

The player's goal should be part of the narrative of the game 'find the lost puppy' instead of being tied to 'your' goals for the player such as 'learn about careers'. A good place to provide the player with the 'overarching' goal or quest would be in the introduction. Alternatively, you can use the description on the Player tab or the 'first' agent for this purpose.

Sequential Games

Consider making parts or the whole game sequential to give you tighter control over unfolding the narrative.

To make the game sequential, drag the agents in the agent ribbon to put them in the order the players should encounter them (this is for the designer's ease of use only). Then use the 'accessibility' radio button on the agent page to make sure the first agent is included in the game and all other agents are excluded from the game. Then for each agent, add a bump script with a include block to include the next agent in the game.

Game Mechanics

Making Decisions

The player's activities and decisions should focus around your goals for the player.

Whatever your goals are for the player, you'll want to spend some time planning ways to elicit the learning in your game. Asking the player to make a choice/decision is a great way to prompt the player to think more deeply. The decision points in the game should be aimed at your goals for the player. There are many ways a player can make a decision in a TaleBlazer game.

Using an Agent to Ask a Question

Ask the player to pick one action among multiple actions on an agent dashboard.

It's easy for the game designer to add multiple actions to the dashboard. It is more difficult to ensure that the player picks one and only one action. This will be important for games in which decisions are meaningful choices that affect the outcome of the game. Once the player has made a choice, it often makes more sense to let the player experience the outcome of this choice than to allow the player to change their mind. To force the player to make a decision and to prevent them from later changing their mind, do the following:

- Make the game sequential (or at least each 'question' node sequential) so that all agents are excluded at the outset and are only included in the bump script of the previous agent.
- On the agent that will ask the question, set the description of the agent to the question.
- Modify the bump settings for that agent and turn ON 'Allow re-bump'. This allows the player to revisit the agent if they close the agent dashboard without making the choice.
- Add a script action for each choice the player can make (yes/no, true/false, etc).
- Cut the include block for the next agent from the bump script and paste it to the script action for each choice. These may be different agents based on the outcome of the choice your player has made.
- Consider giving the player feedback for each choice by including a 'say' block in each action script. Remember not to give them critical information in the 'say' block as they will not be able to review it.
- Prevent the player from making the other choice(s) by either
 - Using the hide action blocks in each action script to hide all choices OR
 - RECOMMENDED
 - Close the agent dashboard by using the switch to tab block

- Prevent the player from returning to the question agent by excluding it from the game (otherwise they will be able to revisit it)

Reviewing Decisions

Allow your players to review important choices they made.

You can do this in many ways.

- *Display choices as agent names in the history tab.* Add a block to the choice action script to display the choice the player made such as: ‘set name of <choice agent> to “You planted a tree at Spot Pond”’ – then when the player looks at their history tab, they will see a list of the choices they made.
- *Display choices as agent names in the inventory tab.* Same idea as above, only the game designer can more easily display JUST the choices instead of all the visited agents.
- *Display choices as traits on the player dashboard.* Add the question as a hidden trait for all roles. Once the player makes the choice, set the value of the trait to the choice they made using the ‘set trait’ block and show the trait on the dashboard using the ‘show trait’ block.
- *Display choices as actions or descriptions on agents, world or player.* The game designer has tight control over the text on the agent, player, and world dashboards. Depending on the other game mechanics you provide, one of these areas may make more sense for your game.

Inventory as a Choice

Allowing the player to pick up and drop items using the pickup and drop actions on the agent dashboards means that the player will have the choice on whether or not to have or keep an agent in their inventory. What happens if the player decides not to pick up the agent? Consider using a 'pick up' block to force the object into the player's inventory if this is not really a meaningful choice for the player.

Making your Game Easy to Play

The goal is for the player to think deeply about the game content instead of being frustrated over not knowing how to play the game. Scaffold the player’s experience by repeating a pattern of game mechanics and starting the player off with a brief tutorial on how to play your game.

Repeating a Pattern

Consider using a repeating pattern to help the player master the game mechanics quickly.

For example, consider chunking the game into 'nodes' of several agents each. For example, the first agent in each node might give the player the challenge, the second agent provides information the player can use to solve the challenge, and the third agent asks the player to make a choice to solve the challenge. Finally the fourth agent can give the player feedback on how they did with the challenge. The first node should include meta instructions as a tutorial (see the section on Tutorials), then include 3 - 5 more nodes for the rest of the game in the main Region.

Creating a Tutorial

Consider writing a tutorial to scaffold the player’s experience.

The tutorial is played at the outset of the game and provides specific step by step instructions on how begin to play the game. Consider your audience. If there's a chance that the player might be playing their first TaleBlazer game, then at a minimum, the tutorial should scaffold the player on how to:

- Read the map
- Bump an agent
- Switch between tabs

Your tutorial may include additional scaffolding to learn the game mechanics specific to your game. It should be long enough to cover the essentials but otherwise brief. If your game is comprised of several sets of nodes which repeat a pattern, then design the first node to be the easiest and include meta-directions within the agents in this node on how to play your game. Alternatively, you can provide "just in time" tutorial content which scaffolds the player regarding additional game mechanics later in the game.

- Think of the tutorial as the first part of your game. It should be fun and engaging and hook the player from the get-go.
 - Integrate the tutorial within the game narrative or context.
 - Think of the tutorial as a part of the game with extra help that will no longer be necessary once the player masters the fundamental game play mechanics.
- Don't fall into the trap of too much text. Have your player DO the game mechanic rather than simply telling them how.
- Consider introducing the player to the narrative and initial (or overarching) quest in a *brief* introduction.
- Create a 'tutorial' Region.
 - Consider locating the tutorial in a place where staff will be available to help the player.
 - Make this Region the default Region (where the player begins the game).
 - This Region should have a custom map (see above).
 - Before uploading the custom map image, add two layers:
 - A text layer with a text box instructing the player to walk to the first agent.
 - A layer containing an arrow from the text box to the first agent's location.
 - Separating these layers from the map and from each other allows for ease of modification should the need arise (for example if you choose to move the location of the first agent).
 - Either make the map tab first or use the 'Switch to map tab' block in a 'When game starts' script to make sure that the map shows first.
- Include 'meta' instructions on the agents in the tutorial Region to tell the player what to do to play the game. For example, in the description of the first agent, include the text: 'Tap the Back button to return to the map' (author's note – the text on this specific button may change to 'OK' in an upcoming build of the mobile)
- Introduce the game mechanics one at a time. Consider hiding all the tabs except the Map and the Game tabs when the game starts and then showing additional tabs one at a time to introduce them.

Ending the Game

Coming soon!

Testing the Game

Before Testing with Others

Check for obvious errors before asking others to test your game.

Glaring errors can be distracting to your testers. Before you ask others to help, put your best foot forward.

- Grammar
- Placement of agents – make sure all agents are accessible to pedestrians or your testers will not be able to complete the game.
- Game Debugging – does the game do what you expect it to do?

What to test

Playability Issues

- Did the players figure out what they were supposed to do in an acceptable amount of time?
- reading level/quantity - is the amount of reading and the reading level appropriate for the target audience?
- A/V - were the players able to hear the audio (if any)? Was the player able to make out the images in the game?
- glitches - were there problems with the game? with the software? with the location? with the hardware (GPS/battery/etc)

Logistical Issues

- length (time/distance-wise) - is the game length too long/too short/just right? Is the amount of walking too much/too little/just right?
- density - are the agents too spread out/too concentrated/just right?
- real world - are there opportunities/encouragement for the players to observe or interact with the real world? do the players interact with the real world?
- conclusion - does the player know when the game ends?

Engagement

- appeal - did the player enjoy the experience?
- narrative - engaging? too babyish? too sophisticated?
- agents - engaging? too babyish? too sophisticated?
- roles (if any) - did the player enjoy picking a role? Did they enjoy playing that role?
- game complexity - is the complexity of the game play appropriate for the target audience? (too hard/too easy/just right)?
- images/videos - did the player like the images and/or videos in the game?

- surprises - did anything in the game surprise the players (good/bad)
- conclusion - did the player feel satisfied with the conclusion of the game?

Game Goals

This is very important, but it's also game specific. What is the game trying to get the player to think about or learn about?

Did the game succeed in its lofty goals?

Research Questions

As your project matures, a list of research questions may develop that should be included on the evaluation tools.