December 2021

Automatically Generated Gameplay Strategy Recommendations

Abhilash Kuduvalli

Automatically Generated Gameplay Strategy Recommendations

ABSTRACT

Video game players can get stuck at certain levels of gameplay. Current techniques to make progress through a game, e.g., looking up articles or how-to videos, asking friends for tips, etc., are time-consuming and interrupt game flow. This disclosure describes techniques that, with user permission, detect when a game player is stuck at a certain level in gameplay. Based on user-permitted analysis of prior gameplay streams from other game players, strategies to advance in the game are automatically identified, e.g., video snippets that illustrate the specific actions that the player needs to perform. Such strategies are then presented to the game player, assisting the player in completing the level.

KEYWORDS

- Recommendation engine
- Gameplay
- Game strategy
- Game walkthrough
- Game playthrough
- Game hint
- Video game
- Video understanding
- Crowdsourcing
- Video analysis

BACKGROUND

Video game players can get stuck at certain levels of gameplay. For example, to complete a level, a player may need to solve a puzzle, find the next checkpoint, defeat an NPC (non-player character) opponent, etc. which the player may, in spite of multiple attempts, be unable to do. In such situations, game players utilize various tactics to make progress through the game. For example, such tactics include looking up articles and/or how-to videos, asking friends for tips, etc. These are time-consuming and interrupt game flow. Some game developers put in the effort

to include tips to complete difficult levels for specific game instances and on specific hardware. However, such efforts are not widespread, are not scalable, and may not address the particular difficulty being faced by a player.

Members of the game playing community commonly upload videos featuring their gameplays, e.g., from beginning of game to end; focusing on nuances of the game; illustrating unsuccessful attempts to proceed past a level; illustrating successful passage through a level; etc. Such uploaded videos are known as game playthroughs.

DESCRIPTION

Per techniques of this disclosure, analysis of uploaded streams of gameplays by various players is performed to determine successful playing strategies at different levels of a video game. With permission from a game player, detection techniques are employed to detect when the player is stuck at a certain level. Upon such detection, video(s) that illustrate playing strategies that illustrate how to pass the level are provided to the game player. The player can turn off automatic detection and instead, manually request assistance. Further, in addition to or instead of analysis of streams of gameplays, other methods of determining successful strategies and providing assistance, e.g., hints provided by the game developer, crowdsourced how-to videos, etc. can be used.

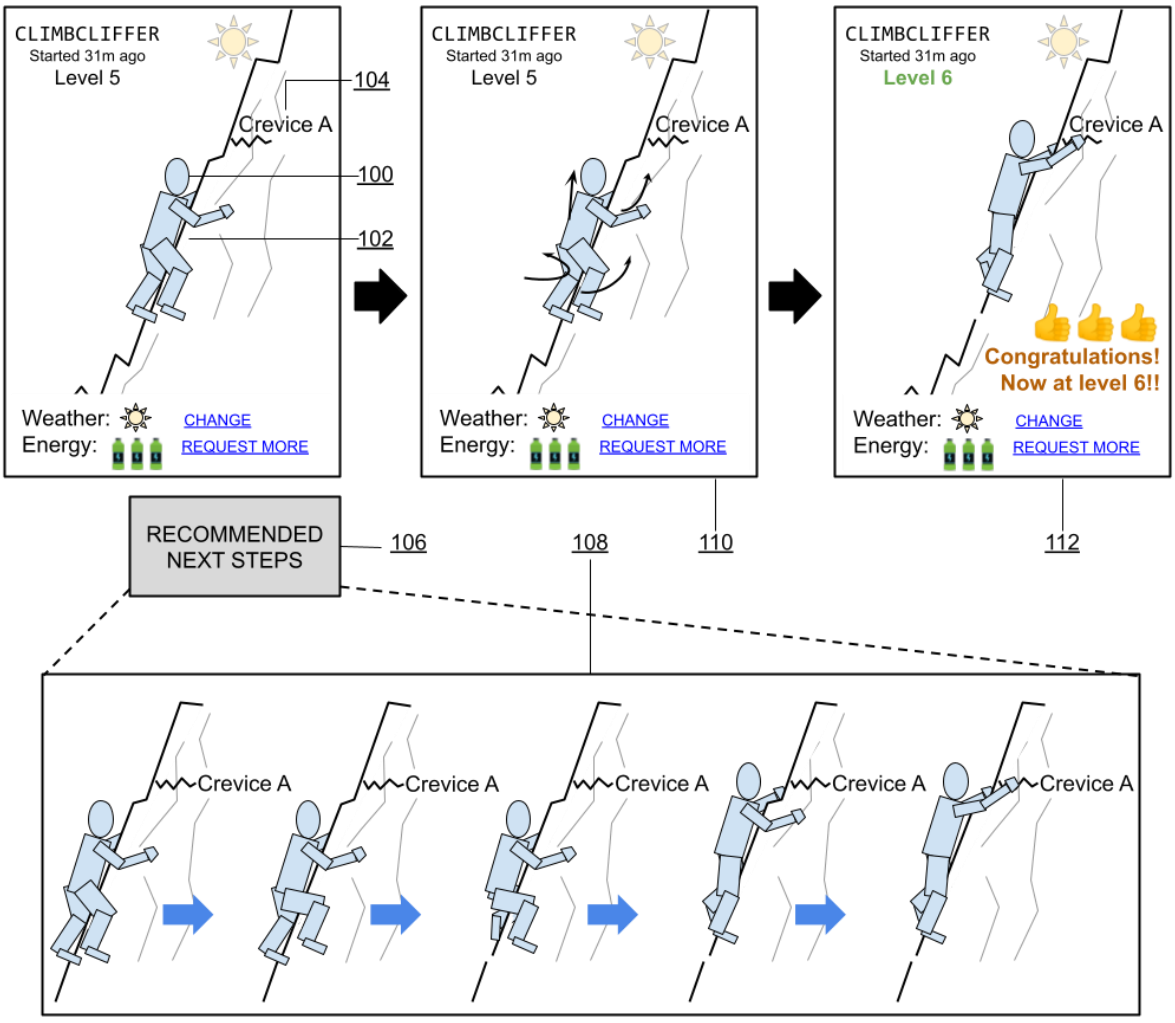

Fig. 1: Example of video recommendation to traverse a game level

Fig. 1 illustrates an example of a video recommendation to traverse a game level. The objective of this game is to get an avatar (100) to climb a cliff (102). With user permission, it is detected that the player is stuck at a certain level (level 5). This may occur due to an inability of the player to operate the game controls to get the avatar to grip on to a crevice A (104).

Upon detecting that the player is stuck, a recommendation window (106) is popped up based on the current level of gameplay. When the player interacts with the recommendation window, e.g., by clicking on or hovering over it, a video (108) of a successful climb at level 5 is shown. This video illustrates the avatar gripping the crevice that enables traversal to a higher

level of gameplay. As illustrated, in the example successful attempt, the avatar moves its legs up, crouches in position, lifts its body, and stretches its arm to grip the crevice. The example video that illustrates successful gameplay can be generated automatically (e.g., by the game engine) or with appropriate permissions, be mined from gameplays uploaded by other players in the game playing community. The video is clipped to just the section that pertains to traversing the level that the player is stuck at or seeking next steps for, allowing the player to enjoy the next challenge in the game.

As illustrated in Fig. 1, after viewing the example of successful level traversal, the player executes the corresponding arm/leg/body movements of the avatar (110) to reach the crevice and to pass successfully to the next level - from level 5 to level 6 (112).

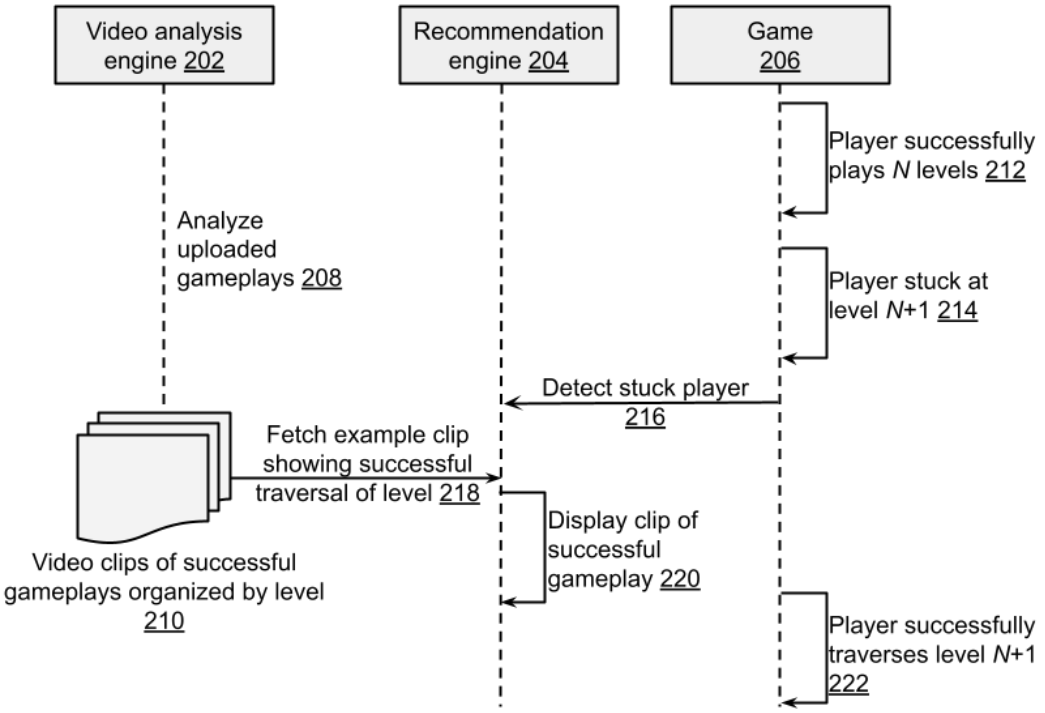

Fig. 2: Recommending next steps in gameplay

Fig. 2 illustrates an example process to detect when a player is stuck and recommend next steps in gameplay. A player plays a video game (206). A video analysis engine (202) analyzes

gameplay content provided by other players (208) that have provided permission for such analysis and stores video clips of successful gameplays, organized by level (210). The video analysis can be performed offline, separate from ongoing gameplay instances. The example clips of successful gameplays can be short, e.g., 5-60 seconds in duration, and are captured to include just the portion of gameplay where a level was successfully traversed. Example techniques of video analysis are explained in greater detail below.

The player successfully traverses the first N levels (212) but is stuck at level N+1 (214). With permission, it is detected (216) that the player is stuck, or the player may manually request assistance. Upon detection of a stuck player or receipt of a request for assistance, a recommendation engine (204) fetches from the stored clips of successful gameplays an example clip that shows successful traversal of the level (218) that the player is stuck at. Example techniques of detecting a stuck player are explained in greater detail below. The clip of successful gameplay is displayed (220) to the player, e.g., in a picture-in-picture format within the main game or in a side window. The player can view the example clip of successful level traversal and implement the strategy to successfully traverse level N+1 (222).

Analyzing uploaded streams to detect next steps for a given level

It is common for online video game services to receive and store gameplay streams uploaded by the game playing community in large numbers, with individual streams being several hours long. Such streams can include gameplay events for several levels. Permissions are associated with such videos where individual players can provide consent for the game service to perform automated analysis of the stream to detect successful level traversals.

As explained earlier, portions of successful level traversals are few and short, e.g., 5-60 seconds in duration, their content focused on sections of video that indicate specific level

traversals. Of the large numbers of uploaded videos, the most pertinent ones can be selected by analysis of metrics, e.g., the number of likes, views, claps, etc., and/or analysis of comments posted by viewers ("I just loved the level jump at 38:40"). Within a selected video, sections that pertain to level traversals can be isolated by techniques such as:

- *Video understanding* can be used to home into sections of the video that pertain to level traversals.
- *Auto-generated audio commentary* ("Woo-hoo! You've made it to Level 8!!") accompanying gameplay can be analyzed to determine sections of the video that display level traversals.
- *Title cards* within the video stream announcing milestones in gameplay ("Entering Level 3!").
- *Comments posted by viewers* ("Level 8 jump at 38:40 was brilliant!") can be used to determine sections of the video that display level traversals.
- Some uploaded gameplay videos include a *game-state evolution*, e.g., the state of the game corresponding to the frame of the video. Levels being a type of game state, level traversals in gameplay can be determined from the game-state evolution, if available.
- *Comments, descriptions, or audio commentary posted by the uploader* ("folks, I want to show you how I cracked Level 8 in climbcliffer") can be used to determine sections of the video that display level traversals.

Detecting when a game player is stuck at a certain level

A player can herself request recommendations for strategies at any level in the game. Even without a request from the player for recommended strategies, certain player actions can be indicative of the player being unable to move beyond a certain level. Examples of such actions

include pausing a game; spending inordinate time at a level compared to other levels or compared to game designer's expectation of the time needed to cross the level; zooming out of the game window to see other windows including the recommendations window; using game help pages or a search engine to look up game hints or game walkthroughs; video analysis that indicates inconclusive movement within the level; etc. With user permission, one or more such actions can be analyzed to detect that a player is stuck. The player can provide or decline consent to access any of these factors.

While the foregoing description refers to game levels, strategy suggestions can be provided at any point in the game where the player is stuck. Further, for video games with gameplay elements that are different for different players (e.g., due to random seeding of game elements, different orders of gameplay sections, etc.), the suggestions are specific to the particular player's instance/ version of the game.

In this manner, the techniques of this disclosure enable a game player to cross gameplay sections where they may be stuck. Automated analysis of gameplay streams from the community of game players is performed to determine those that correspond to success at the particular gameplay section. If an ongoing game of a player is detected as being stuck at a certain level, an overlap of the player's position with a successful gameplay is found and strategies - the best next step(s) - are suggested based on prior successful gameplay. Based on where the game player is in the game, the player is frictionlessly provided successful examples that assist them in making progress, effectively automating the finding of game playthroughs.

Further to the descriptions above, a user may be provided with controls allowing the user to make an election as to both if and when systems, programs, or features described herein may enable the collection of user information (e.g., information about a user's games, gameplay

actions, events within a game that a user is playing, a user's game streams, a user's gameplay videos, or a user's preferences), and if the user is sent content or communications from a server. In addition, certain data may be treated in one or more ways before it is stored or used so that personally identifiable information is removed. For example, a user's identity may be treated so that no personally identifiable information can be determined for the user. Thus, the user may have control over what information is collected about the user, how that information is used, and what information is provided to the user.

CONCLUSION

This disclosure describes techniques that, with user permission, detect when a game player is stuck at a certain level in gameplay. Based on user-permitted analysis of prior gameplay streams from other game players, strategies to advance in the game are automatically identified, e.g., video snippets that illustrate the specific actions that the player needs to perform. Such strategies are then presented to the game player, assisting the player in completing the level.

REFERENCES

[1] "PlayStation 5 Activity Cards Are Brilliant For Families, These 15 Games Do Them Right" available online at https://www.forbes.com/sites/andyrobertson/2021/09/16/playstation-5-activity-cards-are-brilliant-for-familiesthese-15-games-do-them-right/?sh=4d195ee4d194 accessed Dec 10, 2021.

[2] "A closer look at PS5's Game Help feature" available online at https://blog.playstation.com/2021/06/11/a-closer-look-at-ps5s-game-help-feature/ accessed Dec 10, 2021.