# – Game Flow Analysis

## Introduction

## Game Analysis (Methods)

## SkyNet (Bioware, Industry)

## TRUE: Tracking Real-Time User Experience

## Playtracer

Playtracer (ANDERSEN *et al.*, 2010; LIU *et al.*, 2011) is a visual tool designed to illustrate how groups of players move through the game space. Playtracer can be used for behavior analysis from games with the concept of state transitions. The transitions in the game are represented as game states by applying the Classical Multidimensional Scaling (CMDS) (COX; COX, 2010) to project the game space in two dimensions. Thus, Playtracer aids the designer by showing common pathways and alternatives that players used to succeed or fail in their tasks, identifying pitfalls and anomalies in the scene while also tracking how particular players progressed through the levels in the game.

In Playtracer, a play trace is the path that each player took in the game, scaled to two dimensions by using CMDS. The transformation places similar states close to each other while dissimilar states are placed apart. Thus, CMDS allows for easily identify similarity between states that were visited by players. The distance between states is calculated by following specific metrics that are customized by each game. Distance metrics are also used to analyze different features of the game. For example, a distance metric with a component to compare how many steps are necessary to reach a goal state will result in clustering goal states while placing states that are difficult to reach the goal far away. Thus, the designer can identify players that are not making progress in their goals and possibly investigate the issue.

An example of the Playtracer graph output is illustrated at Figure 1. The input for Playtracer is a list of all the states that the player visited during the game and a distance metric to calculate the distance between states and generates the graph. In the Playtracer’s graph, the vertices represent the game states and the directed edges are the movements the player did to move from one state to another. Furthermore, the size of the vertex, or state, is proportional to the number of players that reached that state. Thus, the size of the vertex can be used to identify which states were more visited by players.



Figure 1: Playtracer state visualization. Circles represent game states and edges represent paths taken by players. The distance metric used for this graph clustered states with similar pieces of statements. Figure taken from (LIU *et al.*, 2011).

Moreover, the graph utilizes color to distinguish displayed information. A yellow state is the game’s initial state and green state represents the goal. Blue edges represent moves made by players who won the game and red edges for those who lost. The shades between red and blue represent the probability that the player who reached the state completed the game successfully. Lastly, cycles in the graph represent failed attempts from the players, where they made a move but returned to a previous state.

The main focus of the Playertracer is to display aggregated user behavior in a graph in order to aid in understanding common strategies adopted by players and to identify points of confusion for players. To solve problems related to game with many states, Playtracer uses features to aggressively cluster states together to make a cleaner visualization. Another feature is to make equivalent states to be represented by the same state, reducing the number of states displayed in the screen. Lastly, it is possible to filter the graph (winners from losers) to visually compare their respective behaviors in order to identify similarities. A drawback is that Playtracer do not take in consideration temporal information. For example, Playtracer do not measure how long the player took to execute an action or the time between actions. Long pauses could mean that the player was thinking his next course of action. Likewise, consecutive actions could imply frustration or panicking, which is a common behavior in games.

## Play-Graph

Play-Graph (WALLNER; KRIGLSTEIN, 2012; WALLNER, 2013) is a concept to formally describe and visualize gameplay data by using different graph visualizations to describe changes between two datasets. The gameplay analysis of the play-graph illustrates the sequence of states performed by actions from the players over the course of the game. In the Play-Graph context, a game state describes a certain configuration of the game or an entity, while actions are player interactions within the game, like shooting, jumping, or using an object. These actions are responsible for changing the current game state due to influences generated in the current state or to other entities.

In this concept, a game is viewed as a finite state machine with a finite number of states and transitions between them. Thus, the state machine can be represented by a directed graph and each vertex in the graph represents a state from the game while edges represent actions. States are composed of a set of attributes that are used to define a state from the game. Meanwhile, actions are triggered by players at a specific point in the game and can be of different types or have a duration. For example, possible types of actions are running, walking, jumping, and pulling a lever. The possible types are defined by the game. Furthermore, actions (edges) linked to states (vertices) can have labels to provide additional information to differentiate from other states and actions.

The Play-Graph visualization is composed of Node-link diagrams. Nodes, or vertices, in the graph represent game states. The size of each node is directly related to the number of players that visited that state at any time during the game. Moreover, multiple edges from the same source to the same target are merged to create a meta-edge. The thickness of each meta-edge is proportional to the number of edges that composes the meta-edge. It is possible to have two meta-edges between two nodes due to the nature of the directed graph, where each meta-edge represents a different direction. Furthermore, each node and edge type in the graph is distinguished by different colors. Lastly, icons in the graph represent the players in the game. The icon color is directly related to certain attributes from the player (gender, age, character class). Figure 2 illustrates the basic representations from the graph.



Figure 2: Basic elements from the Play-Graph: (a) nodes representing states, (b) directed edges representing player’s actions, (c) meta-edges, and (d) player icon representing time-dependent location of individual players. Figure taken from (WALLNER, 2013).

## Final Considerations

## References

ANDERSEN, Erik *et al.* Gameplay analysis through state projection. *Proceedings of the Fifth International Conference on the Foundations of Digital Games*, FDG ’10. p. 1–8, 2010. Acesso em: 14 set. 2012.

COX, Trevor; COX, Michael. *Multidimensional Scaling, Second Edition*. [S.l.]: CRC Press, 2010.

LIU, Yun-En *et al.* Feature-based projections for effective playtrace analysis. *Proceedings of the 6th International Conference on Foundations of Digital Games*, FDG ’11. p. 69–76, 2011. Acesso em: 24 jun. 2013.

WALLNER, Günter. Play-Graph: A Methodology and Visualization Approach for the Analysis of Gameplay Data. *Proceedings of the Eighth International Conference on the Foundations of Digital Games*, FDG ’13. p. 253–260, 2013.

WALLNER, Günter; KRIGLSTEIN, Simone. A spatiotemporal visualization approach for the analysis of gameplay data. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’12. p. 1115–1124, 2012. Acesso em: 24 jun. 2013.