

---

Department of Electrical and Computer Engineering  
Technical University of Crete  
Autonomous Agents  
Instructor: MGLagoudakis  
Christos Ziskas  
Register Number: 2014030191  
February 23, 2020-"Project Report"

---

KLONDLIKE SOLITAIRE

# 1 INTRODUCTION

The object of the study is the construction of the agent that implements the solitaire game. Agent constructs the graphical environment of the game. In addition there are features for performance of a complete game.

- Graphic illustration of cards, board
- Original Menu and other scenes
- Show score
- Show number of moves
- Game duration
- Win/Loss Background
- Audio accompaniment
- Card Movement

It provides the player with possibilities to solve the game, but also to recognize his defeat, through buttons.

1. Hint - Find available traffic

2. Quit - Acknowledgment of defeat

## 1.1 GAME PRESENTATION

The game of solitaire is played with usually 1-2 decks of cards. The particular version is the basic game played with a deck of cards.

### 1.1.1 Purpose Of The Game

The ultimate goal is to turn the entire deck into stacks, and if this is achieved, the game Solitaire is won. This is usually done by releasing the cards and making room for the stacks to be placed.

### 1.1.2 Paper Hierarchy

The order of the cards in the game is: K (high), Q, J, 10, 9, 8, 7, 6, 5, 4, 3, 2, A (low).

### 1.1.3 By Sharing

There are four different types of stacks:

1. Tableau
2. Foundations
3. Inventory
4. Talon

Tableau: Seven stacks that make up the main tableau.

Foundations: Four piles in which to create a full breed or order. In most games Solitaire, the four aces are the first card or the base of the foundations. The 4 foundation piles are hearts, diamonds, spades and clubs.

Stock (or "hand"): If the entire deck is not laid out in Tableau at the start of a game, the remaining cards make up the stock pile from which additional cards are played according to the rules.

Talon (or "Trash"): Cards from the stockpile that do not they have a place on the table or on the foundations they are placed in front of the waste pile.

To create the Tableau, seven piles must be created. Start moving from left to right, the first card is placed face up to make the first pile, placing a card face down for the next six piles. Starting again from left to right, a face-up card is placed in the second pile and a face-down card is placed in piles three through seven. Again starting from left to right, one card is placed face up in the third pile and one card is placed face down in the piles 4 to 7. The pattern continues until the seventh pile has one card face up up into a pile of six cards face down.

The remaining cards form the stock pile (or "hand") and are placed on the table. When the game starts, the foundations and waste pile have no cards.

#### 1.1.4 Game

The original board can be changed by “building” – moving cards between the face-up cards in Tableau. Some cards of the board can be played immediately, while others may not be played until certain cards that are in the way are removed. For example, of the seven cards face up on the table, if one is a nine and the other is a ten, the transfer is made immediately to the pile with the smallest being transferred to the largest to begin building the pile in turn. Since a card is moved from one of the seven piles, it now unblocks a face-down card. This card can be flipped and is put into play.

As cards are moved to the table and ranks begin to be formed, if an ace is revealed, the ace must be placed on one of the foundation piles. The foundation is built with suit and in order from ace to king. The transfer of cards to the piles continues at Tableau.

If no face-up cards can be moved, the stock pile is used by turning over the first card. This card can be played in the foundation or on Tableau. If you can't play the card at Tableau or in the foundation piles, the card is moved to pile of garbage and turns another.

Vacancy at Tableau is created by removing the papers elsewhere and is called space. Space is of great importance in its management Tableau. If a space is created, it can only be filled with one king. Filling a space with a king potentially unlocks one of the face-down cards in another pile on the table. Transfer of cards to the table continues and cards are inserted from the stock pile until all cards are built in suit sequences in the foundation piles. If this scenario does not occur, the game is lost.

## 2 IMPLEMENTATION

The implementation took place in stages:

- Menu
- Other Scenes
- Game environment

At each stage, work was carried out to shape the final implementation. These stages are essentially the states the environment transitions into during a game

## 2.1 Menu construction

At this stage the messages that should appear in the original were taken into accountscreen and are essentially the options of the game.

This screen Menu has 3 options.

1. Game Option #1 - Draw per 1 card from the stock pile
2. Game Option #2 - Draw every 3 cards from the stock pile
3. Exit



Figure 1: Home Menu

The menu, has fonts to form the menu items and display are displayed on the screen. The title is formed in its center screen and with particular arrangement, the are lined up items on the rest of the screen. Selecting items as well as crossing them is done by pressing its buttons keyboard. By choosing any of the 2 game options, the environment transitions to the state play. With the exit, the game terminates - state exit (none). The rest of the screen is formed by a white background that is overlapped by the desired image.

## 2.2 Other Scenes

There are two versions of scenes:

1. Preparation Stage
2. Final Result Stage (Win/Loss)

At this point, in whatever sub-stage the system finds itself, similar results are presented depending on the outcome of the result. Each scene lasts a certain amount of time and then transitions to the next state. Specifically, the preparation stage lasts 2 seconds and then there is a transition to game mode. The final ones stages have an unlimited duration. As the duration time is unlimited there is no transition to any state. In these, an exit of the program is required.



Figure 2: Preparation Stage

The information at this point is according to the stage (Preparation/ Result Message) accompanied by the score, moves and completion time. In the initialization stage, the data has the initial values. Messages are displayed using fonts and their layout is central, one below the other.



Figure 3: Result/Victory Stage

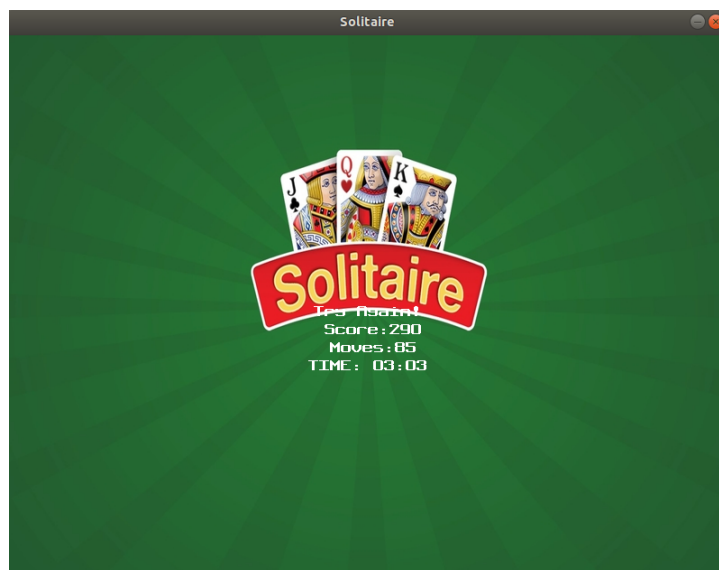


Figure 4: Stage of Result/Loss

## 2.3 Game Environment

The game can also be divided into sections



construction:

1. Initialization
2. Design
3. Update

## 2.4 Initialization

Initializing the game includes:

1. Create cards
2. Creating the positions of the cards on the screen
3. Create background sound

In addition, a variable is initialized that is used to identify the available moves. Its value is defined as the number of cards in the stock pile. Its usefulness will be analyzed below.

### 2.4.1 Creating Cards

This process is carried out by reading the images of cards. They are stored in a format pixels. Then a string defined as the physical existence of the card (e.g. 8 spades) is stored in a dictionary with a key. The keys of the dictionary create a list which is shuffled. This action shuffles the deck and to each hand, there are different cards on the board.

### 2.4.2 Creating the positions of the cards on the screen

The construction of the board is done by creating classes that capture the stacks now as entities. The pile has the characteristic that provides the set of cards and the area covered by each card is recorded. There are its entities/heaps Tableau, inventory, and the foundations with their additional features. Therefore one inventory entity, four foundation entities and seven entities are created Tableau which they do inheritance of the heap entity. First, the inventory pile is initialized, followed by the tableau and foundations. then cards are inserted into the piles - otherwise inserted into the class lists.

The pileTableau has a structure that stores the hidden cards as well and the height at which the pile is so that the cards are placed one under the other. In addition, there are also the open cards in the pile in the basic structure. This is how the correct layout of the game is done as the distance that the piles must cover is taken into account.

The Stock pile has a structure for hidden cards. Initially this structure is full. Upon opening the pile the hidden cards are reduced and placed on the pileTalon. It is modeled by a list structure inside in class. In addition to this list, they are also kept in another list that will keep all the cards with the opening of the inventory so that the process is repeated and kept in the pile. The input layout is taken into account to adhere to the rules of the game.

The foundation pile consists of the structure that holds the cards that are placed on it. Initially the structure is empty. Each time a data is entered it enters the end of the list and the game surface shows the item at the end of the list.

The cards are first inserted into theTableau. The cards that are stored inTableau are deleted from the card structure so that they do not a second choice of card is made in another pile. The remaining cards are stored in the stock pile.



Figure 5: Starting a game

Additionally, it is important to note that there is an additional class in the implementation that deals with the cards that the player moves. It does inheritance the base stack.

#### 2.4.3 Creating background sound

It creates a channel that plays audio in the his background game on repeat stream.

### 2.5 Design

In this stage, the cards are modeled and drawn on the playing surface. Each class has methods for drawing the cards whenever, depending on the entity it represents, the corresponding drawing is done. The dictionary containing as keys the string of cards and prices pixels of the images, are captured in the recording and depending on the logic that each pile has, the corresponding appearance is made. The heights of the stacks and cards as well as the dimensions of the cards are taken into account. They are recorded on the surface fonts they present the number of score, the moves and the duration of the game. Additional, the box is drawn hint which contains functionality to find possible movement. If the agent perceives the sure solution of the game, it is formed button that leads to the result/win state. If the agent realizes that there are no moves available, then it is formed button leading to result/loss state. Then there is a draw and for card moved from pile.

### 2.6 Update

This event includes:

1. Move cards
2. Hint Button functionality
3. Check for Hint Availability

### 2.7 Moving Cards

The movement of the cards consists of moves drag n drop. Drag done by each heap class. Drop is done only by the moving object. Drag refers to a method that involves grabbing a card from a pile, while

thedrop refers to an object method that is moved and left on a heap. OR card being madedrag is inserted into the mover's information if- object All modifications to heap sizes are taken into account, clearing data from departure heaps, Whendrop it movable element is released. If successful then it is moved to the list of open cards in the new pile. Otherwise it goes back to the pile it was on. Moving cards from the foundation pile is not allowed.

## 2.8 Hint Button Functionality

The button acts as a means for the player to recognize possible moves in the tableau facing. The aid has a possible origin, either from his pile tableau, or from the stock pile. The method it is the responsibility of the agent and provides a service to the player. O agent follows a policy to highlight potential traffic:

- Move cards into his pile tableau with control from left to the right
- Move cards from tableau on the foundation
- Move cards from inventory to tableau
- Move cards from stock to foundations

Its implementation begins by checking the open cards that are in his pile tableau. The check for finding possible movement is done checking the stacks from left to right. If there is any movement available, then the two cards are colored around for as long as the button is held down. A different policy is followed for the Ace and K cards lying in piles at tableau. The ace cards in tableau, are driven to the foundations. The leaves of K at tableau, are driven to empty positions in the tableau. To find inter- set move, the two open cards at the bottom of the stacks are checked. If they have different suit and have a unit difference, then n traffic is available. This is done iteratively for each combination of piles in tableau. Then a check is made to see if there is movement available from the tableau to the foundations. Traffic availability in this case is controlled by tai if in the foundations there is the card with the suit and with number smaller by a unit from the sheet under consideration. If no traffic is found from the tableau then it starts searching for movement from the inventory.



Figure 6: Presentation of Hint

Implementation continues with stock. It is checked if the stock cards are open. Depending on the draw from the stock, the control is also done (at the draw3 cards first the card at the bottom of them is checked open cards of the inventory). The process is similar to searching for movement when the card is on tableau.

## 2.9 Check for Hint Availability

When initializing the game, a variable is initialized that helps with this process. The variable is initialized to the number of cards in the stock pile. Therefore it has value24. When the stock pile is completely opened then the number of cards opened is checked. This check is done to see if leaves have been removed from the stockpile.

If they did not leave, then in order for there to be no traffic availability, two conditions must apply:

1. Unavailability of card movement from tableau
2. Non-availability of Movement cards from stock

The way in which the movement of cards from a pile is controlled tableau has been analyzed previously. Stock motion control is true as long as a single motion is possible - depending on the type of pull. So the inventory is divided into slices according to the number of cards where the pull is done. From this slice is checked the paper at the bottom with some from the tableau. It is enough to complete the process for each slice and not is there any sheet available.



Figure 7: Unavailability Hint

When the stockpile has completed the cycle, no moves are available, and the button is pressed, then the appropriate no moves message is displayed. Then a new button appears in the game which takes the game to a final defeat state.

If the game reaches a point where the agent realizes that a solution is possible, then a button is created that takes the program to a final win state.



Figure 8: Game resolution

### 3 Future Features

Plans noted that could be implemented:

1. Undo action
2. Indication of different hints by repeatedly pressing the button
3. Different score depending on the movement and time penalty
4. Optimizing the algorithm for finding possible motion

The task was implemented in the toolPyCharm