**DELHI TECHNOLOGICAL UNIVERSITY**

**THEORY OF COMPUTATION PROJECT**

**FLAPPY BIRD USING AUTOMATA THEORY**

**Under supervision of: GEETANJALI GARG MAM**

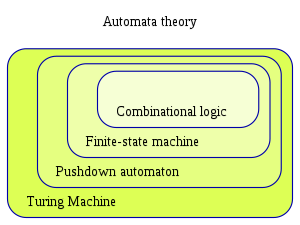


**Submitted by :**

**PRAJJWAL CHITTORI (2K18/CO/249)**

**(Dept. of Computer Science and Engineering)**

# ABSTRACT

Automata Theory is the study of abstract machines and automata, as well as computer problems that can be solved by using them. It is a theory in Theoretical Computer Science.

The Figure shows the limited state machine, which is a known type of automaton. This automaton consists of provinces (statistically represented by circles) and transitions (represented by arrows). As the automaton detects the input signal, it changes (or exceeds) to another position, depending on its conversion function, which takes the current status and the latest symbol as input.

**GAME DESIGN AND IMPLEMENTATION**

In the current scenario game design has been performed using NFA (Non-Deterministic Finite Automata) and DFA (Deterministic Finite Automata) in which NFA must be converted to DFA during development cycle and using Mealy machines helps make the design more complex.

This project focuses on the design of the game Flappy Bird (Published by dotGears).

The design approach uses Meanly Machines which will be helpful in further automation of the game.



**INTRODUCTION**

Most of the modern System Software are designed using Automate Theory. Game Design done using Automata Theory Aids the developer to program complex games. Education and Training using games have been really successful in the current times. Games provide a way to improve motor skills and decision-making abilities of the player.

Automata concept can play a critical role in bridging the distance among game designers and programmers.

The layout of the game “Flappy Bird” is offered in this paper the usage of Mealy Machines. It is very easy for programmers to apprehend the running of recreation due to Mealy machines, because these machines are self-descriptive so programmers can effortlessly understand and they can program it without any ambiguity.

Mealy machines are faster in comparison to Moore machines additionally mealy have much less quantity of states. It is easy to understand and implement as evaluate to Moore machines. In this paper we decided on mealy machines to offer sport layout because mealy machines are quicker because of instant output on every trade in input.

**METHODOLOGY**

The Game Flappy Bird is endless runner game until the bird hits any tunnel or and other type of obstruction. This recreation is made the use of limitless loops with a few situations on which it breaks the loop and exit to Game End nation.

The game will have following states-

* **Run-**The game main sprite the bird will be in the running state and all the other background sprites will be perform according to that.
* **Game over-** Game will be in over state when the bird hits the tunnel or drops onto the ground.
* **Jump-** The bird sprite’s elevation will increase during this state.

1. There will be different types of background layers which will create an illusion of parallax scrolling.
2. Different types of tunnels will be implemented to act as obstacles for the bird
3. Player can jump by pressing the space button
4. The player will collect coins which will increase the coin count.
5. Score will be the amount of distance covered by the bird.
6. The end state will be triggered when-

* Player hits an obstacle
* Player falls to the floor

**Defining Input Alphabets**

∑ Move-{Automatic Input }

∑ Action-{Space}

∑ Game-{{GS GE, CC, HH, FE,-F, C+, D+}

**Description of Input Alphabet-**

**∑ Move**- The player’s movement will be automatic and the player will keep on gliding downward until the jump action is not performed.

**∑ Action**-Defines the action of players. On Using Action the Player Sprite jumps.

**∑ Game**- This state represent different situations during game execution GS (Game starts) GE (Game Ends) CC (Coin Collected) C+ (coin added) D+ (distance add) HH (hurdle hits).

**Background (BG)**

 Overlapped Background Images will be used to create the illusion of parallax scrolling. When the first image is moving forward the second image is not in motion and when the second image is in motion the first image is static.

**There are following Situations in the game-**

* Player collects coin
* Player collides with enemy

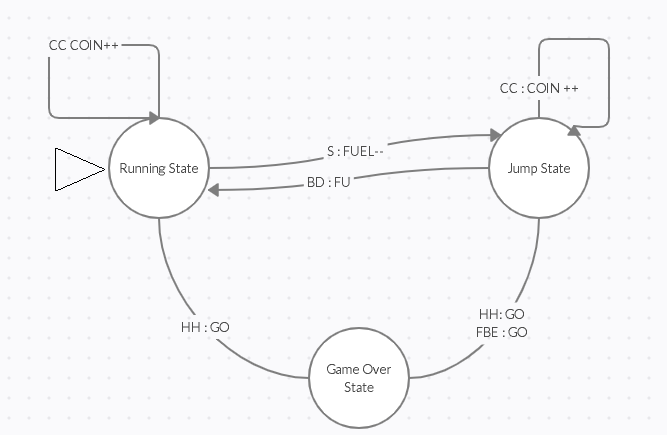
**Different Types of Automata Models**

|  |  |  |  |
| --- | --- | --- | --- |
| Model Type | Input Type | Head Direction | Memory |
| Finite State | Read Only | One way | - |
| Pushdown | Read Only | One way | Stack |
| Linear -Bounded | R/W | Two-Way Direction | (Bounded) |
| Turing Machine | R/W | Two-Way Direction | (UnBounded) |

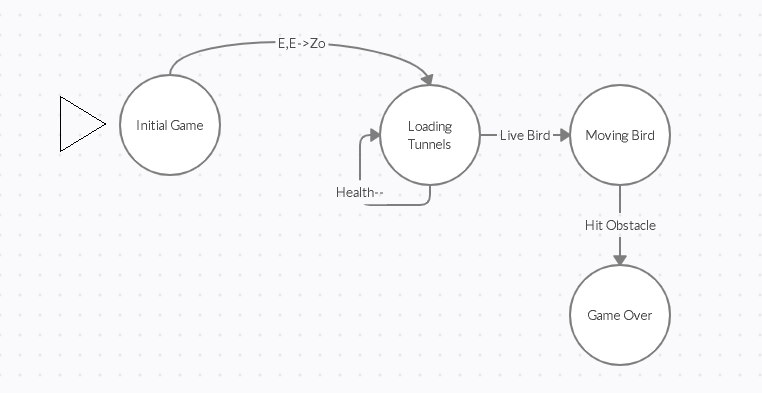
**PDA consist of six components:**

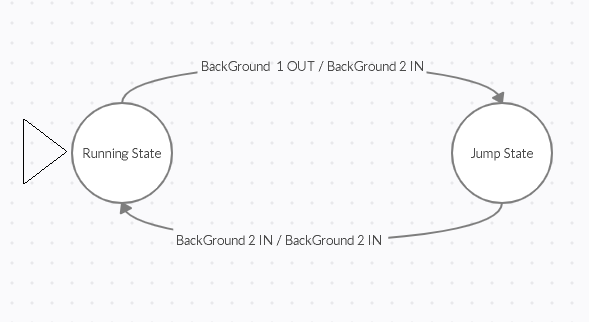
* a read-only input tape
* input alphabet
* a finite state control with two head, one read only and the other one read/write
* a finite set of final state
* an initial state
* a stack called Pushdown Store (PDS)

**Flappy Bird Game**

****Flappy Bird is a game in which the bird moves automatically and player presses the space button to make the bird jump. It is an arcade-style game in which the player controls the bird, which moves persistently to the right. The player is tasked with navigating bird through pairs of pipes that have equally sized gaps placed at random heights.

Main Design Meanly Machine for Flappy Bird

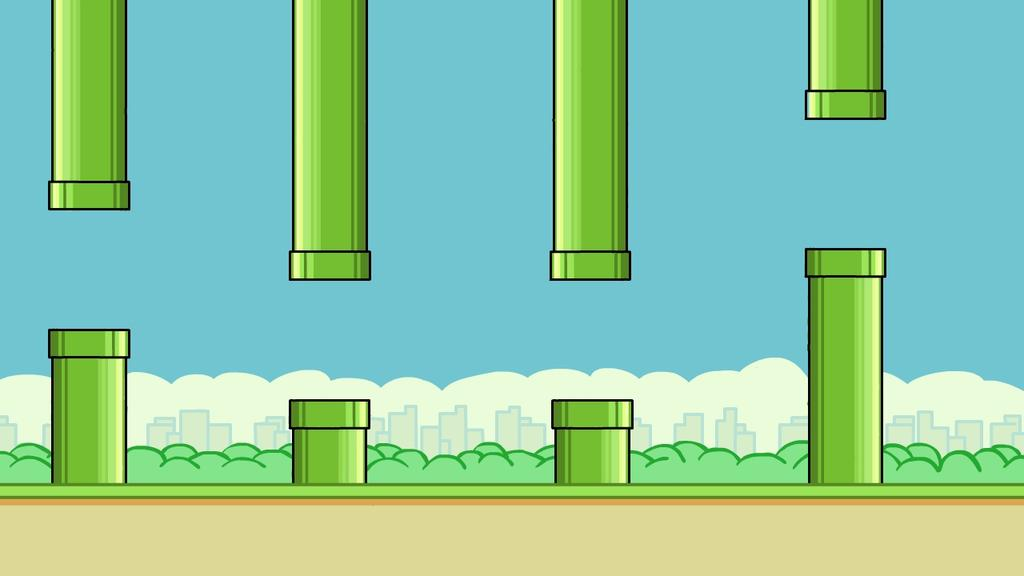
 PDA for Flappy Bird



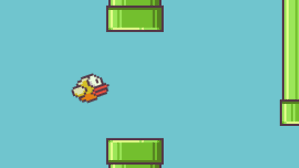
Meanly Machine for Background Movement

**Results**

The game is designed by utilizing the PDA and then developed using Unity Game Engine.



Tunnels

**** Jump Animation

Game Over State

Based on the development cycle, it is concluded that designing the PDA prior to developing simplifies the shifting between one state to another and provides simple inputs and outputs.

Furthermore, it results in a more systematic and organized code for Flappy Bird. In addition, each game class in the code represents states within the PDA design.

Afterward, Blackbox testing is conducted to check whether or not or now not inputs from user are the same as expected outcomes.

Below is the table that contains the results of Blackbox testing.

**The Result of BlackBox Testing**

|  |  |  |  |
| --- | --- | --- | --- |
| S.no | Input | Expected Outcome | Does Expected Outcome same as Actual Outcome? |
| 1. | No Input | Bird Keeps Moving Forward | Yes |
| 2. | Bird Moves Forward | Score Increases | Yes |
| 3. | Bird moves onto Coin | Coin Increases | Yes |
| 4. | Bird touches Tunnel | Game Over | Yes |
| 5. | Bird Touches Ground | Game Over | Yes |
| 6. | No Input | Bird Keeps moving Downward | Yes |

**Conclusion**

PDA has been utilized to design Flappy bird game which uses pushdown automata to emulate various states. It has been shown in the Blackbox testing that there is consistency between inputs and expected outcomes.

Furthermore, we located that designing a game using Automata is the best method via which a game designer can design a game without having any knowledge in programming with visualization. Therefore, it’s a helping tool for the game designer and visual artists to control various aspects of the game.

PDA helps in streamlining the process of programming and coding in the game and provides ease during development.

**Reference**

* <https://apiar.org.au/wp-content/uploads/2018/10/16_APCCR_Aug18_BRR748_ICT_v2_50-56.pdf>
* <https://en.wikipedia.org/wiki/Automata_theory>
* <https://www.ijcseonline.org/pub_paper/8-IJCSE-02134.pdf>
* M. K. Brunnermeier, J. Morgan, “Clock Games: Theory and

Experiments”, Games and Economic Behavior., Vol.68, Issue.2, pp.532-550, 2010

* Wattanasoontorn, V., Boada, I., García, R. and Sbert, M., 2013. Serious games for health. Entertainment Computing, vol.4, no. 4, pp.231-247.
* <http://ijcsse.org/published/volume5/issue7/p1-V5I7.pdf>