

Domino Project	Version: <2.0>
	Date: <08/11//2014>
Approved by :	



Faculty of computers and information
Cairo university

Course name: cs 251

Program TA :mohammed Elarnaouty

Project name: Domino.

Leader name:mohammed ramadan.

Team

ID	Name	E-mail	Mobile
20120308	Mohamed Ramadan abdlmaksoud	Mido_zeko80@yahoo.com	01143514863
20120305	Mohamed khairy ibrahem	Mohammed.khairy2011@gmail.com	01000649695
20120303	Mohamed hatem minesy	Mohamed.hatem100100@gmail.com	01020828482
20120310	Mohamed salama Mahmoud	Mohamed_salama384@yahoo.com	01096244319
20120416	Mina foaud yosef	minapharos@gmail.com	01203215235

Domino Project	Version: <2.0>
	Date: <08/11//2014>
Approved by :	

Requirements document for distributed domino game

November 8, 2014

Contents

Domino Project	Version: <2.0>
	Date: <08/11//2014>
Approved by :	

1. Introduction 4

1.1 Purpose	4
1.2 Scope.....	4
1.3 Overview	4
1.4 Glossary.....	4

2. Specific Requirements 5

2.1 Functional Requirements	5
2.2 Non Functional Requirements	8
2.2.1 Portability.....	8
2.2.2 Availability.....	8
2.2.3 Recovery.....	8
2.2.4 Performance.....	8
2.2.5 Reliability.....	8
2.2.6 Efficiency.....	8
2.2.7 Throughout.....	8
2.2.8 Integrity.....	8
2.2.9 Usability.....	8
2.2.10 Implementation.....	9
2.2.11 Flexibility.....	9

3. Use cases 10

3.1 Use case diagram.....	10
3.2 Use case tables.....	11

4. Ownership report 18

Domino Project	Version: <2.0>
	Date: <08/11//2014>
Approved by :	

Chapter 1

1. Introduction:

1.1 purpose:

The purpose of this software is to make and develop distributed domino game. also this software aims to make this game easy to play.

1.2 Scope:

Our system allows multiple players to play domino game on shared server using devices enhanced with Bluetooth and Android2.3 to Android4.4.

1.3 Overview

Domino game consist of 28 stones each one has two sides and each side takes numbers from 1 to 6 or empty. if two player each one take 7 and the rest on ground for pulling put .If 3 players every one take 9 and discard white stone put .If 4 players every one take 7 and no stones for ground to pull . player who finished domino cards first win the game and take spots that remain with other players .

1.4 Glossary:

Jam (block):situation in a domino game when no player is able to play .

Stone: Individual domino piece that has two sides each side has set of dots or empty. Also called domino

Spots:circular marks on each side of stone to know its number .also called dots.

Bar:The "bar" is the line seperating the two ends of a domino. Also called the "center" or "divider".

White stone: stone its two sides not have any spots (empty). Also called blank.

Doublet:A domino with both ends having the same value.

Ace: domino that has one side contain only one spot.

deuce:domino that has one side contain tow spot.

Trey :domino that has one side contain 3 spots.

Domino Project	Version: <2.0>
	Date: <08/11//2014>
Approved by :	

Chapter 2

2. Specific Requirements

2.1 Functional requirements:

Functional requirement1:

Description:

When the user selects "Rules" the system will display rules of the game or how they will play.

Functional requirement2:

When one player enter “start game” he make server and user's can enter server at beginning of game using Bluetooth.

Functional requirement3:

Description:

Take number of player and divide stones on them randomly

Input :

Number of players

Processing:

If number of players 2 each one take 7 and the rest put on ground for pulling put. If 3 players every one take 9 and system will discard white stone and no pulling put if number of player is 4 every one take 7 and no pulling

Functional requirement4:

Description:

Player who has highest domino (total number of dots) play first.

Functional requirement5:

Every one of players take turn and play stone .

Functional requirement6:

System checks if current stone valid or not.

Domino Project	Version: <2.0>
	Date: <08/11//2014>
Approved by :	

Functional requirement7:

Description:

if 2 players are competing and one of them has no suitable stone he will pick up stones until find the suitable stone the system will notify him to play it.if he did not find one other player will take on.

Functional requirement8:

Description:

When 2 sides of game has 2 or 1 number and users played all stones that has these numbers ,The current trial will stop and sum points of stones for every player .player who has least points win the current trial.

Functional requirement9:

When a player ends his stones before others he wins the current trial and system sums points, player should collect 101 points to win the game at the end.

Functional requirement10:

If a player left the game while running the game will be ended by the system.

Functional requirement11:

Description:

The user will asked to choose if he wants to play again or exit .

input:

choice from user .

processing:

Check the choice of the user.

Output:

Welcome message and run the game again or exit.

Functional requirement12:

Description:

Make visualize for game using graphic and make update for screen for each player after every player play.

Domino Project	Version: <2.0>
	Date: <08/11//2014>
Approved by :	

For example:
When starting the game:



During the game:



When the game ends:



Domino Project	Version: <2.0>
	Date: <08/11//2014>
Approved by :	

2.2 Non Functional requirements:

2.2.1 Portability:

It's easy to install our game in platform but it's work only on Android.

2.2.2 Availability:

The game is exist Over a period of play and If one exit from the game,The game will exit.

2.2.3 Recovery:

If the game was break down because of every thing like electricity breakdown the system must go to the old version of the game.

2.2.4 Performance

The game take short time to be loaded for every player on server and changes in state take short time (few seconds) to be viewed by every player.

2.2.5 Reliability

Our game can maintain its performance along time without any reducing in performance.

2.2.6 Efficiency

The game will take small space on hard disk and very small space on memory

2.2.7 thought out:

Two or three or four can play the game at same time.

2.2.8 Integrity

In our system there exists security so no one can access Data of specified player like his /her cards.

2.2.9 Usability:

It's easy to use our system and there exist helping to learn you how to use and understand this system .

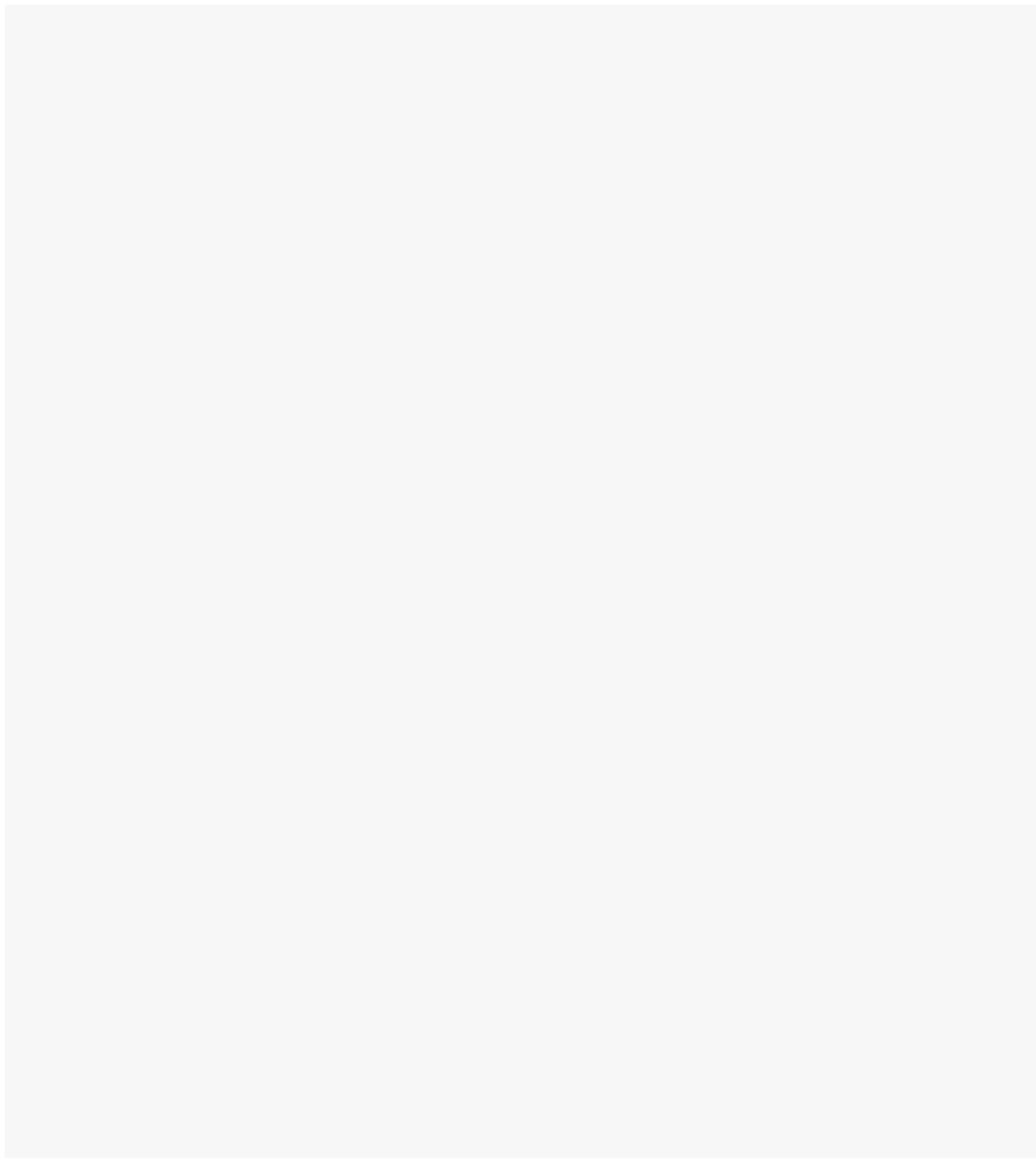
Domino Project	Version: <2.0>
	Date: <08/11//2014>
Approved by :	

2.2.10 Implementation:

Our system will be implemented using java language for android.

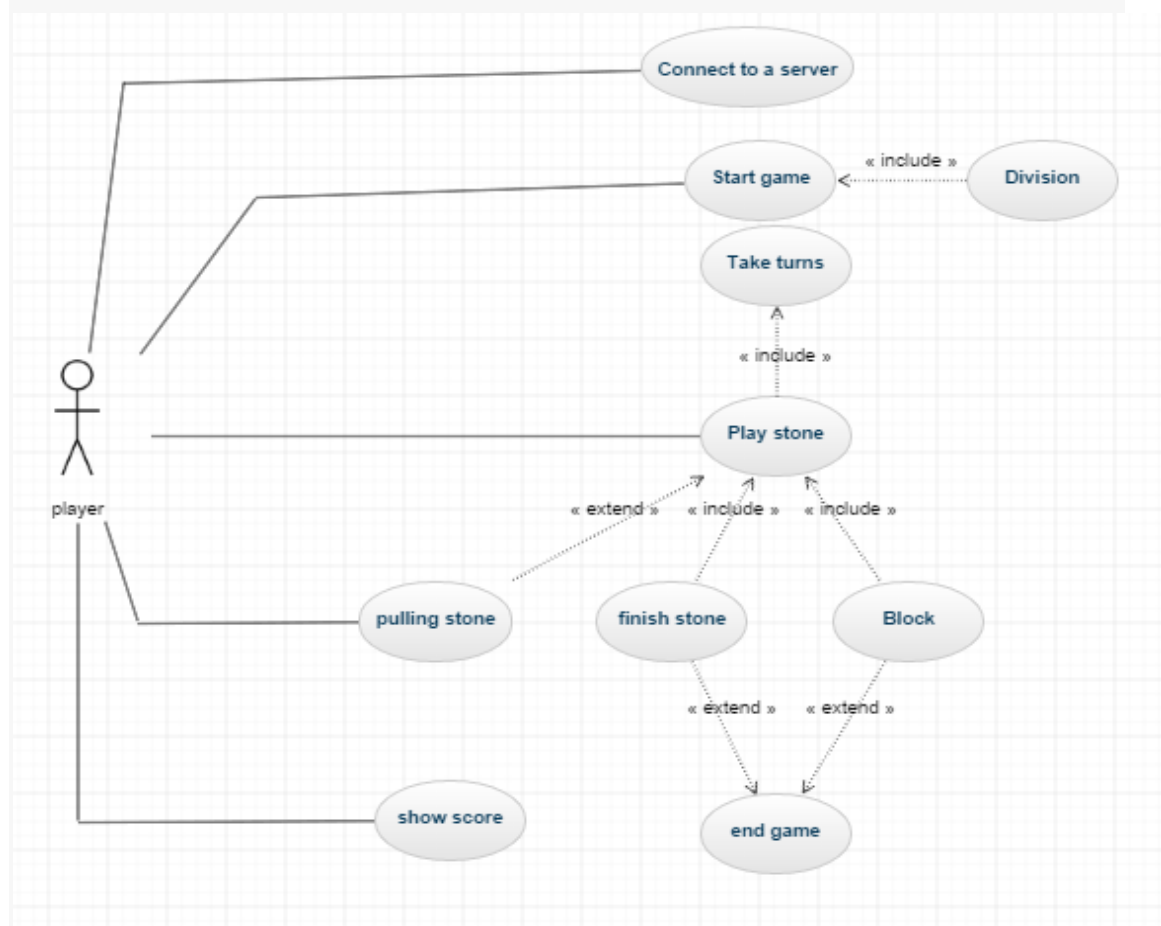
2.2.11 Flexibility

Our system is flexible so it's easy to make changes in software after developing it and company that take this software Can extend its function.



Chapter 3

3.1 Use case diagram



Domino Project	Version: <2.0>
	Date: <08/11//2014>
Approved by :	

3.2 Use case tables

Use Case ID:	110	
Use Case Name:	Connect to the server	
Actor:	Player	
Pre-conditions:	1-User open the game 2-User connected to Bluetooth	
Post-conditions:	1-Player will join to game server 2-Player will see all other players in server	
Flow of events:	User Action	System Action
	1- select a specific server	
		2- system will enter player to server
Exceptions:	If there is no server system allow first user to make server	
Includes:		
Notes and Issues:		

Use Case ID:	111
Use Case Name:	Start game
Actor:	Player
Pre-conditions:	1-The number of players on server is complete
Post-conditions:	1-Every player take stones

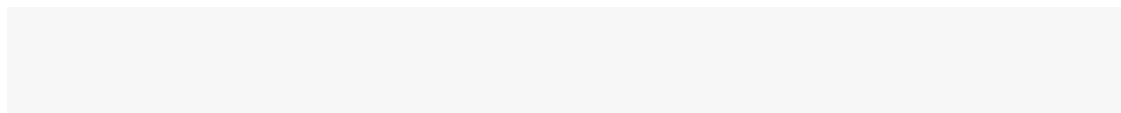
Domino Project	Version: <2.0>
	Date: <08/11//2014>
Approved by :	

	2-first player can play	
Flow of events:	User Action	System Action
	1- player choose start game	
		2-System will take every player his stones
Exceptions:	If server not complete , System prevent me to Start	
Includes:		
Notes and Issues:	First Player connect to server who can start game	

Use Case ID:	112	
Use Case Name:	Division stones	
Actor:	No actor	
Pre-conditions:	1-Player start game 2-Player choose mode of game	
Post-conditions:	1-Every player must take his stones	
Flow of events:	User Action	System Action
		1-System will divide stones randomly according to game mode (mode1,mode2,mode3)
Exceptions:		
Includes:	Start game	
Notes and Issues:	For two players everyone must take 7 stones and the rest for pulling. For three players everyone must take 9 stones and the White block will discard. For four players everyone must take 7 stones.	

Domino Project	Version: <2.0>
	Date: <08/11//2014>
Approved by :	

Use Case ID:	113	
Use Case Name:	Take turn	
Actor:	No actor	
Pre-conditions:	1- The game is started 2-previous player played	
Post-conditions:	1-Player who must play will announce	
Flow of events:	User Action	System Action
		1-Determine who must play and change the status of the game.
Exceptions:	if one finished stone game end and no take turn	
Includes:		
Notes and Issues:	First time, player who has highest stone will play	



Use Case ID:	114	
Use Case Name:	Pulling stone	
Actor:	Player	
Pre-conditions:	1-The current player does not find a suitable stone with him to play	
Post-conditions:	1-The player will take stone from ground and his stones increase by one	
Flow of events:	User Action	System Action
	1- the player pull card	

Domino Project	Version: <2.0>
	Date: <08/11//2014>
Approved by :	

	2-System give stone from ground to player
Exceptions:	
Includes:	
Notes and Issues:	If player didn't find suitable stone to play he must still pull until find it

Use Case ID:	115	
Use Case Name:	Play stone	
Actor:	Player	
Pre-conditions:	1-The player take turn 2-Player have suitable stone	
Post-conditions:	1-put this stone on ground to appear for every player 2-allow another player to play	
Flow of events:	User Action	System Action
		1- the system allow player to play only the stones suitable to stones on the ground.
	2- The player must play one stone suitable to stones on the ground.	
Exceptions:	1-IF player play unsuitable stone , system will prevent him 2-IF player does not has stone to play , system allow him to pull from ground	
Includes:	Take turn	
Notes and Issues:		

Domino Project	Version: <2.0>
	Date: <08/11//2014>
Approved by :	

Use Case ID:	116	
Use Case Name:	block	
Actor:	No actor	
Pre-conditions:	1-There is no suitable stone to play with any player or in pulling ground	
Post-conditions:	1-The game will end, and the player who has the least sum of spots will win the trial.	
Flow of events:	User Action	System Action
		1-System will end the game
Exceptions:		
Includes:	Play stone	
Notes and Issues:		

Use Case ID:	117	
Use Case Name:	Show score	
Actor:	Player	
Pre-conditions:	1-The user started the game	
Post-conditions:	1-Score of player will appear on console	
Flow of events:	User Action	System Action
		1-System review score of every trial.
Exceptions:	IF game not started system prevent player to show score	
Includes:		
Notes and Issues:		

Domino Project	Version: <2.0>
	Date: <08/11//2014>
Approved by :	

Use Case ID:	118	
Use Case Name:	End game	
Actor:	No actor	
Pre-conditions:	1-If game is blocked or one player finished his stone	
Post-conditions:	One player win the game and take spots	
Flow of events:	User Action	System Action
	1-if Player play last stone with him.	1-if the game blocked the system will sum spots for every player and determine the winner and end game.
		2-The game end and this player win
	3-else if player play last stone from specified number that make game block	
		4-The game end and player that has least spots will win trial.
Exceptions:	If one player exit from game , the game will end	
Includes:		
Notes and Issues:		

Use Case ID:	119
Use Case Name:	Finish stone
Actor:	No actor

Domino Project	Version: <2.0>
	Date: <08/11//2014>
Approved by :	

Pre-conditions:	Player play stone	
Post-conditions:	end game and the player win the trial	
Flow of events:	User Action	System Action
		1-system check player stone if he play last stone system end the game
		2-System end the trial and player who finish stone win
Exceptions:		
Includes:	Play stone	
Notes and Issues:		

Domino Project	Version: <2.0>
	Date: <08/11//2014>
Approved by :	

Ownership Report

Item	owner
Introduction	Mohamed Hatem <i>Mina fouad</i>
Functional Requirments	<i>Mohamed khairy</i> Mohamed Salama <i>Mohamed Ramadan</i>
Non-Functional Requirments	Mohamed Salama <i>Mina fouad</i>
Use Cases	Mohamed Ramadan Mohamed khairy Mohamed Hatem