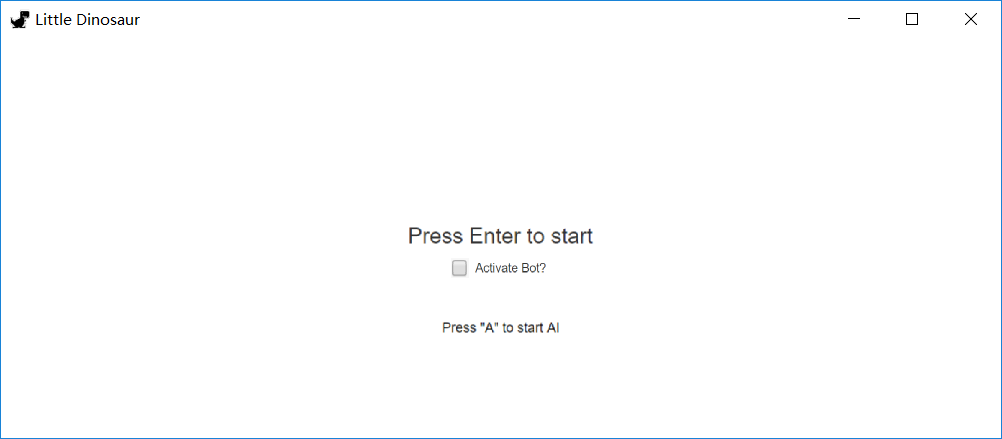
**LittleDinosaur游戏报告**

1. **游戏背景**

原作来自于chrome浏览器无网状态下T-Rex游戏，一只小恐龙在一片二维沙漠中正寻找出路，所以它需要不停地跑步。在路途中它会遇到各种仙人掌和翼龙，玩家需要通过控制小恐龙来躲避这些障碍，使其在沙漠中继续前进。

1. **游戏介绍**

游戏开始会出现如下界面：

玩家可以通过按enter键开始游戏，通过按A来启动AI。

（1）游戏操作：

游戏开始后，玩家需要通过操纵键盘来控制小恐龙的行动：

向上方向键：使小恐龙跳跃。

向下方向键：使小恐龙蹲下，或者在腾空时即刻回到地面。

向上跳跃时连按向上方向键：使小恐龙二段跳。

“A”键：切换AI模式和玩家模式。

“ENTER”键：重新开始。

“ESC”键：返回开始界面。

（2）障碍物设置：

仙人掌有以下三种形态





仙人掌在地面上随机出现，小恐龙均可通过跳跃通过。

翼龙是另一种障碍物，会在空中任何位置出现，小恐龙可以选择下蹲或跳跃两种方式通过翼龙。而一旦小恐龙遇到上述障碍物，游戏结束，画面显示GAME OVER。

玩家可以通过按Enter立即重新开始或者按ESC退回到主界面。

在屏幕右上角显示了玩家的得分，正比于小恐龙所跑的路程。

（3）AI设置

玩家可以在游戏开始时选择是否开启AI模式或者在游戏过程中进行切换。当AI模式开启之时，如果玩家按下键盘，玩家的操作成为优先指令。

**三．具体实现**

使用工具：javafx

1. 地图生成与碰撞判断

初始地图随机在画面右方生成随即数量的树。

随着游戏进行，使用随机函数得到值，根据大小判断画面右方是否存在障碍物，再根据前一个障碍物坐标（遍历所有障碍物对象得到）计算是否局面必死，必死则不生成障碍物，且添加下一次随机的补偿值。

若有障碍物生成，再判断为翼龙或为数，并随机生成翼龙高度。确保大部分翼龙位置较低，且恐龙伏下可以通过。

再随机生成云，并同时随机高度和速度。

当小恐龙与障碍物接触时，游戏判定为结束。通过getLayout函数调用小恐龙图片的坐标和尺寸判断是否和翼龙与仙人掌的图片重叠。一旦重叠就判断为碰撞。

（2）界面输出

小恐龙跳跃的抛物线考虑小恐龙的平动速度和树的高度的因素，由打表得到37帧的高度。

设置键盘监听来读取玩家的输入。

我们从chrome源码得到了游戏各成员的图片，来制作动画显示。

初始化时，生成云、翼龙、树，各自数量为5。

Java默认动画显示为60帧。换算每一帧内左移速度，实现动态效果。

在游戏过程中，不删除或添加图片对象，而是把对象隐藏在画面外，便于计算；由于对象总数较小且游戏较为简单，对于运算影响不大，由于时间紧张，只能后期再考虑优化。

当随机化函数得到障碍物出现时，把其坐标设置在屏幕最右即可。

地面由图片实现，两张图片首尾连接从而循环。

小恐龙的动作，由判断其三个状态（跑动、跳起、蹲下）实现，并以固定间隔改变图片路径从而改变动作。例如，脚动作由每隔2帧改变图片实现。

而在状态改变时，由于图片大小高度有偏差，还需要调整位置。

在跳起切换到蹲下时，则把坐标复原到初始位置，实现瞬移的特殊效果。

（3）AI设置

遍历最近且没有越过的两个障碍物。

若第一个障碍物是翼龙，则保持蹲下。

若第一个障碍物是树，则根据第二个障碍物种类，计算安全跳跃的范围，从而得到安全值。如果不在安全范围内，则保持原来状态，若在安全范围内，则判断：是否有安全值，有则随机选择是否跳跃，若本轮有安全值但未跳跃，安全值减去相应值，若无安全值，则生成安全值。特殊地，若存在安全值但小于既定范围，则立即跳跃。

其中，跳跃优先级高于蹲下，因为蹲下持续时间通常大于所需时间。

**四．AI测试**

AI死的概率很小。